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

Rafał Latkowski and Michał Mikołajczyk

[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 $\mathrm{D}^3\mathrm{RJ}$ method it is possible to obtain smaller set of rules and next better classification accuracy than classic decision rule induction methods. We provide an empirical evaluation of the $\mathrm{D}^3\mathrm{RJ}$ 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 classic 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., [41]) 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 $\mathrm{D}^3\mathrm{RJ}$.

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 D^{3}RJ 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 classic 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 D^{3}RJ method makes is possible to obtain smaller set of rules and to achieve similar or even better classification accuracy than standard decision rule induction methods known from literature.

In the following section the related work on missing values handling and decision rule joining is presented. In Section 3 we introduce some necessary formal concepts. In Section 4 overview of the D^{3}RJ method is provided. Section 5 describes the data decomposition phase. Next, the description of rule induction is provided. Section 7 describes the decision rule joining. In Section 8 contain empirical evaluation of the D^{3}RJ method. The final section presents some conclusions and remarks. This paper is an extended version of [30] where several issues related to decision rule joining were improved.

2 Related work

2.1 Missing Attribute Values

The problem of reasoning with missing attribute values is known in machine learning and a lot of work has been already done for interpretation of the issues related to this problem as well as methods for reasoning with missing attribute values. However, there is no one satisfactory solution to the problems related to reasoning over incomplete data in the considered sense. In relational databases the nature of missing values was established and for more than a decade also the industrial standards fulfill the proposed logical framework and semantical meaning of the null values. Such an approach is not yet available in data mining at all and particularly, in the rough set theory and practice. Furthermore, it seems to be almost infeasible to discover one theoretical framework for dealing with missing attribute values and their role in induction learning that will fit in all aspects of Machine Learning. The findings in area of missing attribute values are rather loosely connected or even exclusive and do not form any coherent guidelines that would be applicable to a wide range of data mining problems.

The problem of missing values in inductive learning received its attention very early. In late '70 and early '80 there were proposed some findings of Friedman in [9], Kononenko et al. in [24] and Breiman et al. in [7] in this area. The proposed methods are addressed to induction of decision trees. The main idea is based on partitioning and replicating data objects and test nodes. In 1989 Quinlan published experimental evaluation of some proposed approaches (see [45]). The experimental evaluation proved, that the Kononenko's method that partition objects with missing values across all nodes is in most cases the best choice. His work influenced a lot the researchers and many later implementations of decision tree induction follow Kononenko's method used also in *C4.5*. This approach became widely applied due to its high performance and simple interpretation. Recent research made on the complexity of this method showed the great complexity breakdown that occurs when data contain many missing values (cf. [28]).

The methods presented above, for decision trees induced by recursive partitioning, build rather an isolated case that is thoroughly investigated. It is a consequence of popularity of decision trees in research and industry, as well as relative simpleness of decision tree induction algorithms. The other approaches for inducing classifiers directly from data with missing attribute values are usually loosely related to each other, but they perform quite well and are based on interesting ideas for dealing with missing values. Two recent examples of such a methods are *LRI* and *LazyDT*. Weiss and Indurkhya in [57] presented the Lightweight Rule Induction method that is able to induce decision rules over data with missing values. This method is trying to induce decision rules by ignoring cases with missing values on estimated test (descriptor). The proper functioning is obtained by redundancy of descriptors in decision rules as well as by normalization of the test evaluation. Friedman's Lazy Decision Tree method (see [10]) presents a completely different approach to classification process, called lazy learning. The decision tree is constructed on the basis of an object that is currently a subject to classification. Missing values are omitted in classified case and ignored in heuristical evaluation of tests.

Besides the methods that can work directly on data with missing attribute values, also the methods for missing values imputation or replacement were proposed. The simplest method — replacing the missing values with an unused domain value — is known from the beginning of the machine learning. However, this yields in significant decrease of classification accuracy. The applied imputation methods can be roughly categorized into simple ones, that do not build any special model of data or such a model is relatively simple, and more complex ones, that impute the missing values with respect to a determined model for a particular data. The most commonly used simple imputation methods are: imputation with mean or median value, imputation with most common value or imputing with mean, median or most common value, where the mean, median or most common value is calculated only over the objects from the same decision class (see, e.g., [18, 19]). There were proposed also some modifications of these methods, such as using the most correlated attribute instead of the decision class (e.g., [11]). The model based imputation methods are usually used with statistical learning methods and are not widely used in other machine learning algorithms like, e.g., decision rule induction. One of the best methods is the *EM* imputation (see, e.g., [13, 58]), where the Expectation-Maximization model is builded for the data and missing values are replaced by randomizing values with probability taken from the model. The EM imputation can be used together with the *Multiply Imputation* (see [46]), that is applied to improve the accuracy of calculating aggregates and other statistical methods. The imputation methods are inevitable in some applications, e.g., in data warehousing. However, in machine learning the imputation methods not always are competitive and their application is not justified or cannot be properly interpreted.

The problem of missing attribute values was investigated also within the rough set framework. We can mainly distinguish two kinds of approaches to this problem with respect to the modifications of the rough set theory they introduce. In the first group of approaches it is assumed that the missing value handling should be an immanent but special part of rough set theory. As the consequence approaches from this group consist in modification of the indiscernibility relation. In the second group of approaches we include all others that do not assume or do not require such a modification.

The practice of modifying of the indiscernibility relation is rather old and originates not directly from the rough set theory but rather from other mathematics areas like, e.g., universal algebra. The adaptation of concept "partiality" from universal algebra leaded to the *tolerance* or *symmetrical similarity* relation as a replacement for the indiscernibility relation. The successful application of symmetrical similarity relations were investigated among others by Skowron, Słowiński, Stefanowski, Polkowski, Grzymała-Busse and Kryszkiewicz (see, e.g., [19, 25, 44, 52]).

To overcome some difficulties in provided semantics of missing values (see, e.g., [52]) also the other types of the indiscernibility relation replacements were proposed. The one of them is the *nonsymmetric similarity* relation which was investigated in [14, 16, 49, 51–53]. To achieve yet more flexibility also the parametric relations were proposed, sometimes also with the fuzzy extension to the rough set concepts (see, e.g., [15, 51, 53]). All of this modifications enforce a certain semantic of the missing values. Such a sematic applies to all data sets and their attributes (i.e., properties of objects) identically and produce a bias in form of model assumptions. One should state, however, that this approach can be very successful in some applications and definitely produces superior results over the standard indiscernibility relation.

There are some other methods proposed within rough set framework that do not assume modification of the indiscernibility relation. The approach proposed by Grzymała-Busse in *LEM2* algorithm for decision rule induction is to modify the induction process itself (see [19, 20]). The special version of LEM2 algorithm omits the examples with unknown attribute values when building the block for that attribute. Than, a set of rules is induced by using the original LEM2 method.

The completely different approach is proposed in the *Decomposition Method*, where neither the induction process nor the indiscernibility relation is modified (see [27, 29]). In the decomposition method data with missing attribute values is decomposed into subsets without missing values. Then, methods for classifier induction are applied to these sets. Finally, a conflict resolving method is used to obtain final classification from partial classifiers. This method can be applied to any algorithm of classifier induction, also these ones, that cannot directly induce classifiers over data with missing values.

The decomposition method performs very good on data, but introduce some difficulties in interpreting last step of reasoning related to conflict resolving. In this chapter, among the decision rule joining, the idea of data decomposition is investigated. The most important improvement of the data decomposition in comparison to the previous research is avoiding the necessity of combining several different classifiers. The decision rules from resulting classifiers are subject to joining similarly as it is described in [33].

2.2 Decision Rule Induction

The decision rule induction problem has been extensively investigated not only within the rough set framework, but also in other fields. In machine learning several efficient algorithms have been proposed, like, e.g., Michalski's AQ algorithms or *CN2* algorithms from Clark and Niblett. Rough sets can be used on different stages of rule induction and data processing. The most commonly used approaches are induction of certain and approximate decision rules by generating exhaustive, minimal or satisfactory set of decision rules (see [23, 50]). Such algorithms for decision rule induction were extensively investigated and are implemented in many software systems (see, e.g., [6, 17]).

There were proposed also methods for decision rule induction related to the local properties of data objects (see, e.g., [4, 5]). This approach combines advantages of lazy learning, e.g., reduced computational complexity in the learning phase with advantages taken from induction of rough set based decision rules.

In recent years also a similar problem to the decision rule induction has been investigated — the searching for association rules (see, e.g., [2, 21, 35, 36]). It is possible to represent a set of all the possible descriptors as a set of items. Then the problem of calculating the decision rules corresponds to searching for the sets of items. Each item set corresponds with one decision rule.

Decision rules express the synthesized knowledge extracted from data set. In our research we use the decision rule induction using the indiscernibility matrix and boolean reasoning techniques described in, e.g., [23, 47, 48]. Such decision rules represent some level of redundancy that from one point of view can increase the classification accuracy, while from the second one can result in too many decision rules. There were proposed some approaches for redundancy elimination as well as for classification accuracy improvement in the case of noisy or inexact data. These approaches are mainly based on shortening of decision rules (see, e.g., [4, 34]). The shortening techniques are very useful and with careful parameter assignment can improve classification accuracy and decrease the number of decision rules.

2.3 Decision Rule Joining

Decision rule joining is one of the methods reducing number of decision rules. Although there is not much done in area of decision rule joining or clustering, the reducing number of rules has already been investigated. The most common approach to reduction of the number of decision rules is filtering. The filtering methods assume some heuristical measure on decision rule evaluation and drop unpromising rules with respect to this heuristical evaluation.

One can reduce the number of rules by selecting a subset of all rules using for example quality-based filtering (see, e.g., [1, 40]). In this way we get fewer rules at the cost of reduced classification quality mainly (see, e.g., [54]). With such a reduced set of decision rules some new objects cannot be recognized because they are not matched by rules. Hence, with fewer rules it is more probable that an object will not be recognized. The essential problem is how to rate the quality of decision rules and how to calculate the weights for voting (see, e.g., [12]).

The quality-based filtering methods give low classification quality, but make fewer mistakes than many other decision systems. This is a consequence of smaller set of rules taking part in the voting, which results in lack of classification for weakly recognized objects. This shows that dropping decision rules decrease important information about the explored data.

Recently some methods for decision rule joining and clustering were proposed. The System of Representatives, described in [33], is the method that offers a rule joining. This method achieves very good classification accuracy and model complexity reduction, but it is very time consuming. Therefore we utilize here a simplified method for rule joining that is less time consuming, called Linear Rule Joining (LRJ). This method also achieves good results and has been designed to cooperate with data decomposition method.

3 Preliminaries

3.1 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., [23, 42]). A *decision system* (*decision table*) is an information system with a distinguished attribute called decision (see, e.g., [23, 42]).

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

This definition assumes that all objects have complete description. However, in a real world data frequently not all attribute values are known. Such attribute values that are not available are called *missing attribute values*. The above definition of decision table does not allow an object to have an incomplete description.

As a consequence of this fact the missing attribute values are rarely considered and they are not present in theoretical foundations of proposed methods. To be able to deal with missing attribute values we have to extend the definition of a decision table. There are two main concepts on how to give consideration to missing values. The first and more popular is to extend the attribute domains with a special element that denote absence of a regular attribute value. The other approach, taken from universal algebra, is to assume that attributes are *partial* functions in contrast to attributes without missing values assumed to be *total* functions. Both approaches are equivalent, but the first one is easier to implement in computer programs.

Definition 2. *A decision table with missing attribute values* $\mathbb{A} = (U, A, \{d\})$ *is a triple, 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. Such an approach is frequently used in all domains of computer science. For example in the relational databases a similar notion — "NULL" is used for representing missing attribute values in database record.

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.

3.2 Total Templates

To discover knowledge hidden in data we should search for patterns of regularities in decision tables. We would like to focus here on searching for regularities that are based on the presence of missing attribute values. A standard tool for describing data regularities are *templates* (see, e.g., [37, 38]). The concept of template requires some modifications to be applicable in the incomplete decision table decomposition.

Definition 3. *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 4. *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.

Once we have a total template t, we can identify it with the sub-table $\mathbb{B}_t$ of original decision table. Such a sub-table consists of the decision attribute, all attributes that are elements of total template and it contains all objects that satisfy template t. Obviously, the decision table $\mathbb{B}_t$ does not contain missing attribute values. We will use this fact later to present the data decomposition process in a formal and easy to implement way.

3.3 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., [23, 42]).

Definition 5. *Let $\mathbb{A} = (U, A, \{d\})$ be a decision table. The decision rule is a function $\mathbb{R}\colon U \to V_d \cup \{?\}$, where $? \notin V_d$. The decision rule consist of condition φ and value of decision $d^{\mathbb{R}} \in V_d$ and can be also denoted in form of logical formula $\varphi \Rightarrow d^{\mathbb{R}}$. If the condition φ 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 $? \notin V_d$ $(\mathbb{R}(x) = ?)$.*

In above definition one decision rule describes a part of exactly one decision class (in mereological sense [41]). 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 support or quality).

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

$$(a_{k_1}(x) = r^{k_1} \wedge \ldots \wedge a_{k_n}(x) = r^{k_n}) \Rightarrow d^{\mathbb{R}},$$

where $x \in U$, $a_{k_1}, \ldots, a_{k_n} \in A$, $r^{k_i} \in V_{k_i}$. For example:

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

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. [37, 56]). The generalized rules have the form:

$$(a_{k_1}(x) \in R^{k_1} \wedge \ldots \wedge a_{k_n}(x) \in R^{k_n}) \Rightarrow d^{\mathbb{R}},$$

where $R^{k_i} \subseteq V_{k_i}$. It is easy to notice, that any classic decision rule is also a generalized decision rule, where $R^i = \{r^i\}$. From now on we will assume that all decision rules are generalized.

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

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

The empty set denotes absence of condition for that attribute.

4 D^{3}RJ

The D^{3}RJ method is developed in the frameworks of Granular Computing and Rough-Mereology [41]. 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 describes the whole investigated phenomenon is decomposed — partitioned into a number of subsets that describe, in a sense, parts of investigated phenomenon. Such a procedure creates an overlapped, but non-exhaustive covering that consist 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 phase 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. Each decision rule can be perceived as an information granule that correspond to knowledge induced from the set of objects that satisfy the conditional part of decision rule. The set of decision rules can be perceived as a higher level granule that represents knowledge extracted from the data subset. 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 decision rules are induced from data without missing values to take an advantage of 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 disturbances.

In the fourth phase the set of classic decision rules is converted to the smaller and simplified set of 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 — lower level granules. The main objectives of the decision rule joining are reduction of classifier complexity and simplification of knowledge representation.

The D^{3}RJ method returns a classifier that can be applied to a data with missing attribute values in both, learning and classifying.

5 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. Ideally, the complete sub-tables that are result of the decomposition should correspond to natural subproblems of the whole problem domain.

The result of data decomposition is a family of subsets of original data. Subsets of original decision table must meet some requirements in order to achieve good quality of inductive reasoning as well as to be applicable in case of methods that cannot deal with missing attribute values. We expect the decision sub-tables to exhaustively cover the input table, at least in the terms of objects, to minimize the possibility of loosing useful information. They should contain no missing values. It is also obvious that the quality of inductive reasoning depends on a particular partition and some partitions are better then others.

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 maximal decision sub-table $\mathbb{B}_t \in \mathcal{G}_t$ is uniquely determined by template t. With such an assignment we can consider the data decomposition as a problem of covering data table with templates. The finite set of templates $S = \{t_1, t_2, \ldots, t_n\}$ determines uniquely a finite decomposition $D = \{B_{t_1}, B_{t_2}, \ldots, B_{t_n}\}$ 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. With such a unique assignment the decomposition process can be formally described in terms of total templates. The preference over particular decompositions can be translated to preference of particular set of templates.

We illustrate the data decomposition with an example. Let consider the following decision table:

	a	b	c	d
x_1	1	0	*	1
x_2	0	1	1	0
x_3	*	0	1	1
x_4	*	1	0	1

In above decision table 92 out of 127 nonempty combinations of seven possible total templates create proper data decompositions, i.e. that exhaustively cover all objects. For example, the total template $(a \neq *) \wedge (b \neq *)$, which covers objects x_1 and x_2, with the total template $(b \neq *) \wedge (c \neq *)$, which covers objects x_2, x_3 and x_4, create a proper data decomposition.

The problem of covering decision table with templates is frequently investigated (see, e.g., [37, 38]) and we can make an advantage of broad experience in this area. In our case the templates cover the original decision table in following sense. We say that an object x is *covered* by a template t if object x satisfies a template t. The decision table is covered (almost) completely, when (almost) all objects from U are covered by at least one template from a set of templates. The preferences or constraints on the set of templates are translated partially to preferences or constraints on templates and partially to constraints on an algorithm that generates the set.

The standard approach to the problem of covering decision table with templates is to apply a greedy algorithm. Such an approach is justified, because it is known that greedy algorithm is close to best approximate polynomial algorithms for this problem (see [8, 22, 32, 39]). 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 a defined stop criterion is satisfied. The most popular stop criterion is to have all objects covered by generated set of templates. Such a criterion is also suitable for our purpose. One should notice that a template generated in further iteration can cover objects already removed from decision table. This property allows, if it is necessary, to include a particular object in two or more data sub-tables related to a specific pattern of data.

5.1 Decomposition criteria

Following the guidelines on covering decision table with templates we have to choose a preference measure that define the concept of best template. The template evaluation criterion should prefer decision sub-tables relevant to the data decomposition and to the approximated concept. This nontrivial problem was investigated in [26, 27, 29]. It is very difficult to define the proper criterion for an individual template for generating decompositions of high quality. This problem could be possibly solved with help of ontology knowledge base for investigated phenomenon, but for now such ontologies are not commonly available. We have to relay only on some morphological and data-related properties of decision table $\mathbb{B}_t$ in order to evaluate template t.

The frequently applied template evaluation function measures the amount of covered data with help of *template height* and *template width*. The template height, usually denoted as $h(t)$ is the number of covered objects, while the template width, denoted as $w(t)$ is the number of descriptors. To obtain a template evaluation function, also called template quality function, we have to combine these two factors to get one value. The usual formula is to multiply these two factors and get the number of covered attribute-value pairs.

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

The importance of with and height in q_1 can be easily controlled by manipulating the importance factor.

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

For such preference measures finding the best template is NP-hard (see, e.g., [37]), so usually also here approximated algorithms instead of exact one are used. It was also proved that there exist measures for which problem of searching for maximal template is PTIME. For example it is enough to replace the multiplication of width and height with addition and the resulting problem can be solved by a polynomial algorithm. Unfortunately, for the decomposition and generally for knowledge discovery problems measures that lead to polynomial complexity are inaccurate and unattractive.

The relation of template evaluation functions q_1 and q_2 with expected properties of decision tables relevant for inductive learning can be easily justified. From one point of view the quality of learning depends on the number of examples. It is proven that inductive construction of concept hypothesis is only feasible, when we can provide enough number of concept examples. A strict approach to this problem can be found in [55] where Vapnik-Chervonenkis dimension is presented as a tool for evaluating required number of examples. From the second point of view using inductive learning we try to discover relationships between decision attribute and conditional attributes. A precise description of concepts in terms of conditional attributes values is required to achieve good quality of classification. Without an attribute that values are important to concept description it is impossible accurately approximate a concept.

Methods that determine the best template with respect to the quality functions q_1 and q_2 are frequently investigated and well documented (see, e.g., [37, 38]). Unfortunately, in our case such quality functions do not sufficiently prefer the templates that are useful for data decomposition over the others. The experimental evaluation in [26] showed that a lot of templates with similar size (i.e. width and height) have very different properties for classifier induction and data decomposition.

There were proposed some other template evaluation functions (cf. [27, 29]) that perform much better than simple q_1 and q_2 functions presented above. These function have some similar properties to the feature selection criteria because the data decomposition itself depends on proper feature selection. The most important issue in selecting such measures is to solve the trade-off between computational complexity of function evaluation and the quality of resulting decomposition.

In rough sets some useful concepts to measure the information-related properties of data set are known, e.g., size of positive region or conflict measure. Based on these and similar concepts a number of template evaluation function were proposed and examined. One of the most promising heuristical template evaluation counts the average purity in each indiscernibility class:

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

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 above formula is calculated for the maximal decision sub-table

$\mathbb{B}_t$ related to the template t, in particular the indiscernibility relation IND_t is based on the attributes from the template t and the indiscernibility classes are constructed only from objects that do not have missing values on these attributes.

To ensure the expected properties of decomposition the heuristical template evaluation can be combined with size properties. Similarly to the q_2 function, one can incorporate an exponent to control the importance of each component of the formula.

$$q_3(t) = w(t)^{\beta} \cdot h(t) \cdot G(t)^{\gamma} \tag{4}$$

There is a number of possible heuristical evaluations functions that can be apply here. One can also use an approach known in feature selection as *wrapper* method, where the classifier induction algorithm is used to evaluate properties of investigated feature subset (see, e.g., [29]). The q_3 template evaluation function combining the heuristical function G with size properties showed in experiments to be reasonable good with respect to quality at the minimal computational cost, while, e.g., the functions based on classifier trials improve quality not so much at the enormous computational cost.

6 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 $\mathrm{D}^3\mathrm{RJ}$ 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., [23, 47, 48]). The indiscernibility matrix, related to the indiscernibility relation, indicates which attributes differentiates each two objects from different decision classes. Using this matrix and boolean reasoning we can calculate a set of reducts.

A reduct is a minimal (in inclusion sense) subset of attributes that is sufficient to separate every two objects with different decision. For each reduct decision rule induction algorithm can generate many decision rules. Different reducts usually yields to different rule sets. These sets of rules are subject to joining, clustering and reduction in the next, decision rule joining phase.

The treatment of decision rule sets in $\mathrm{D}^3\mathrm{RJ}$ differs from usual role of these sets. The obtained sets of decision rules, each one from one decision sub-table, are merged into one set of decision rules. It gives highly redundant set of decision rules, where each object is covered by at least one decision rule using its non-missing attribute values. The simplest reduction of obtained set of rules is that duplicate rules are eliminated. The more advanced classifier complexity reduction employed in $\mathrm{D}^3\mathrm{RJ}$ is decision rule clustering and joining.

6.1 Rule Shortening

The decision rule shortening is a frequently utilized approach for achieving shorter and more noise-redundant decision rules (see, e.g., [2, 34, 60]). In shortening process unnecessary or weak descriptors in the conditional part of a decision rule are eliminated. The method for decision rule shortening drops some descriptors from conjunction φ in the left part of the rule.

The shortened decision rules can possibly misclassify objects. To control this phenomenon the parameter α of decision rule shortening is utilized, which steers the minimal possible accuracy of decision rule. In other words decision rule after shortening cannot misclassify more than $1-\alpha$ objects. The side effect of the decision rule shortening is possibility of multiplication of decision rules, i.e., the result of shortening of one decision rule can be several decision rules. This effect is balanced from the other side by that one shortened decision rule can be a result of shortening of several decision rules. For example, the decision rule $\mathbb{R}$ can be shortened to decision rules $\mathbb{R}_1$, $\mathbb{R}_2$ and $\mathbb{R}_3$:

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

$$\begin{array}{ll} \mathbb{R}_1\colon & (a_1(x)=1 \wedge a_3(x)=4) \Rightarrow d, \\ \mathbb{R}_2\colon & (a_1(x)=1 \wedge a_7(x)=2) \Rightarrow d, \\ \mathbb{R}_3\colon & (a_3(x)=4 \wedge a_7(x)=2) \Rightarrow d. \end{array}$$

Continuing the example, the decision rule $\mathbb{R}_1$ can be result of shortening of decision rules $\mathbb{R}$ and $\mathbb{S}$:

$$\mathbb{S}: \quad (a_1(x)=1 \wedge a_3(x)=4 \wedge a_5(x)=3) \Rightarrow d.$$

In practice we never observe increase of decision rule set after shortening. The decision rule shortening always decrease number of rules almost linearly with respect to the factor α.

7 Decision Rule Joining

The decision rule joining is employed at the end of D^{3}RJ method to reduce complexity of classifier and improve the classification quality. 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 would make this method more complicated. By joining rules with different decisions we calculate rules dedicated not for one decision but for a subset of possible decisions. These rules could be used to build hierarchical rule systems.

The main idea of decision rule joining is clustering that depends on distance computed from comparison of logical structures of rules. Similar rules are easy to join and by joining them we get rules that have similar properties.

Definition 6. *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 $\mathbb{A}$. We define the distance function:*

$$\mathrm{dist}(\mathbb{R}_1,\mathbb{R}_2)=\begin{cases} \mathrm{card}(A) & \textit{when } d^{\mathbb{R}_1} \neq d^{\mathbb{R}_2} \\ \sum_{a_i \in A} d_i(R_1^i, R_2^i) & \textit{otherwise} \end{cases}$$

where:

$$d_i(X,Y) = \frac{\mathrm{card}((X-Y)\cup(Y-X))}{\mathrm{card}(V_i)}.$$

The above distance function is used for comparison of decision rule logical structures and for estimation of their similarity. This function differs from the one presented in [30]. It gives better results and is easier to interpret. Let us consider an example of simple rule joining:

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

If we suppose that each attribute a_i has a domain V_i with ten values $card(V_i) = 10$, then distance between these two rules is $dist(\mathbb{R}_1,\mathbb{R}_2) = 0.6$. After joining decision rules $\mathbb{R}_1$ and $\mathbb{R}_2$ we obtain a generalized decision rule:

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

To illustrate on example the further classification abilities of created generalized decision rule lets consider following objects:

$$\begin{array}{lccccccclccccccc} x_1: & 1 & 3 & 3 & 1 & 2 & 3 & 2 & \qquad x_3: & 6 & 3 & 7 & 1 & 2 & 3 & 2\\ x_2: & 2 & 3 & 1 & 2 & 2 & 5 & 2 & \qquad x_4: & 1 & 3 & 4 & 1 & 5 & 3 & 3\end{array}$$

The objects x_1 and x_2 are classified by the generalized rule $\mathbb{R}$ to the decision class d, while the objects x_3 and x_4 are not recognized and the rule $\mathbb{R}$ returns the answer "?".

Moreover, we can join the generalized rules exactly in the same way as the classic ones. Formally speaking a new rule obtained from $\mathbb{R}_m$ and $\mathbb{R}_n$ have a form $\left\{R^i_{\mathbb{R}_m+\mathbb{R}_n}\right\}_{a_i\in A} \Rightarrow d$, where $R^i_{\mathbb{R}_m+\mathbb{R}_n} := R^i_m \cup R^i_n$. 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 classic 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
$$\mathrm{dist}(\mathbb{R}_m,\mathbb{R}_n) = \min_{i,j}\{\mathrm{dist}(\mathbb{R}_i,\mathbb{R}_j) : \mathbb{R}_i,\mathbb{R}_j \in X^{\mathbb{R}} \wedge d^{\mathbb{R}_i} = d^{\mathbb{R}_j}\}.$$
3. If there exist $\mathbb{R}_m$ and $\mathbb{R}_n$ in $X^{\mathbb{R}}$ such that $\mathrm{dist}(\mathbb{R}_m,\mathbb{R}_n) < \varepsilon$ 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.

Table 1. Classification accuracy of the classic exhaustive decision rule induction and the D^{3}RJ method using various decomposition criteria and decision rule shortening.

α	No decomposition	$w \cdot h$	$w{\cdot}h{\cdot}G$	$w{\cdot}h{\cdot}G^8$
1.0	70.15	70.86	71.57	70.65
0.9	71.64	71.02	71.80	71.18
0.8	73.30	72.41	73.11	72.69
0.7	71.87	71.71	72.11	72.21
0.6	69.72	69.37	70.06	69.80
0.5	67.93	70.40	71.13	71.86
0.4	66.81	70.98	71.06	71.11
0.3	68.28	71.23	71.41	71.33
0.2	66.47	71.60	71.54	71.55
0.1	66.14	71.73	71.61	71.60

8 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. All data sets used in evaluation of the D^{3}RJ method were taken from *Recursive-Partitioning.com* [31]. The selection of these data sets was based on amount of missing attribute values and their documented natural origin. We selected following 11 data tables:

- att — AT&T telemarketing data, 2 classes, 5 numerical attributes, 4 categorical attributes, 1000 observations, 24.4% incomplete cases, 4.1% missing values.
- ech — Echocardiogram data, 2 classes, 5 numerical attributes, 1 categorical attribute, 131 observations, 17.6% incomplete cases, 4.7% missing values.
- edu — Educational data, 4 classes, 9 numerical attributes, 3 categorical attributes, 1000 observations, 100.0% incomplete cases, 22.6% missing values.
- hco — Horse colic database, 2 classes, 5 numerical attributes, 14 categorical attributes, 368 observations, 89.4% incomplete cases, 19.9% missing values.
- hep — Hepatitis data, 2 classes, 6 numerical attributes, 13 categorical attributes, 155 observations, 48.4% incomplete cases, 5.7% missing values.
- hin — Head injury data, 3 classes, 6 categorical attributes, 1000 observations, 40.5% incomplete cases, 9.8% missing values.
- hur2 — Hurricanes data, 2 classes, 6 numerical attributes, 209 observations, 10.5% incomplete cases, 1.8% missing values.
- hyp — Hypothyroid data, 2 classes, 6 numerical attributes, 9 categorical attributes, 3163 observations, 36.8% incomplete cases, 5.1% missing values.
- inf2 — Infant congenital heart disease, 6 classes, 2 numerical attributes, 16 categorical attributes, 238 observations, 10.5% incomplete cases, 0.6% missing values.
- pid2 — Pima Indians diabetes , 2 classes, 8 numerical attributes, 768 observations, 48.8% incomplete cases, 10.4% missing values.

Table 2. Number of decision rules using the classic exhaustive decision rule induction and the D^{3}RJ method using various decomposition criteria and decision rule shortening.

α	No decomposition	$w \cdot h$	$w{\cdot}h{\cdot}G$	$w{\cdot}h{\cdot}G^8$
1.0	9970.54	1149.67	1031.33	872.90
0.9	8835.55	1050.29	941.30	807.19
0.8	6672.00	862.11	783.09	677.45
0.7	4945.65	685.23	626.16	545.29
0.6	3114.22	384.32	349.19	308.29
0.5	1682.63	203.57	193.37	176.61
0.4	1158.45	164.12	159.44	150.85
0.3	661.78	74.09	75.77	72.65
0.2	366.80	43.77	44.95	42.85
0.1	227.59	35.49	36.25	34.00

- smo2 — Attitudes towards workplace smoking restrictions, 3 classes, 4 numerical attributes, 4 categorical attributes, 2855 observations, 18.7% incomplete cases, 2.5% missing values.

In presented results the exhaustive rule induction method was used to induce classifiers from the decision subtables. This method is implemented in the *RSES-Lib* software (see [6]). The data decomposition was done with the help of a genetic algorithm for best template generation (see [29]).

Table 1 presents a general comparison of the classification accuracy using the classic exhaustive decision rule induction with the D^{3}RJ method using various decomposition criteria and shortening factor values α in range from 0.1 to 1.0. Table contains the classification accuracy averaged over eleven tested data sets and ten folds of cross-validation (CV10). 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 similar or slightly worse than classic decision rules at the top of the table, but slightly better at the bottom of it, where the shortening factor is lower. It suggest that if the decision rules are more general and shorter then they are easier to join and the D^{3}RJ method performs better. Table 2 that present number of decision rules, shows that the D^{3}RJ method requires averagely 8 times less decision rules than the classic exhaustive decision rules, called also optimal decision rules. Thus, the reduction of the classification abilities is not as high as the reduction of the model size.

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 $\alpha = 0.8$. The results are presented for the decomposition method without decision rule joining as well as with the decision rule joining. The decomposition method without 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

Table 3. The detailed empirical evaluation of the D^3RJ method using the shortening factor $\alpha = 0.8$, and template evaluation function $q = w{\cdot}h{\cdot}G$.

Table	Before joining		After joining		Profits	
	Accuracy	# Rules	Accuracy	# Rules	Compression	Improvement
att	60.50 ±4.39	2459.3 ±586.24	59.48 ±4.80	673.0 ±169.03	3.65	-1.02
ech	69.19 ±8.65	201.0 ±39.41	68.00 ±7.90	93.9 ±12.57	2.14	-1.19
edu	50.91 ±3.20	3580.9 ±61.20	54.20 ±3.97	397.2 ±22.84	9.02	3.29
hco	82.58 ±7.97	1440.1 ±527.19	83.94 ±6.88	391.2 ±142.73	3.68	1.36
hep	79.42 ±1.52	1454.5 ±104.70	79.42 ±1.52	1253.5 ±85.62	1.16	0.00
hin	72.51 ±4.46	436.2 ±17.88	72.01 ±4.72	285.5 ±15.38	1.53	-0.50
hur2	82.93 ±7.49	197.9 ±38.73	82.84 ±7.58	94.6 ±21.87	2.09	-0.09
hyp	95.23 ±0.09	420.2 ±39.26	95.29 ±0.16	150.0 ±8.16	2.80	0.06
inf2	70.22 ±9.67	4298.1 ±206.92	69.43 ±9.48	3866.3 ±192.67	1.11	-0.79
pid2	72.52 ±4.10	2606.3 ±121.75	70.84 ±3.92	222.9 ±16.86	11.69	-1.68
smo2	64.90 ±2.69	6108.8 ±64.77	68.72 ±0.85	1185.9 ±29.38	5.15	3.82
avg	72.81 ±4.93	2109.39 ±164.37	73.11 ±4.71	783.1 ±65.19	2.69	0.30

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. However, the improvement of the classification accuracy is not significant (Wilcoxon signed rank test p-value is 0.17) as well as the worsening in comparison to classic decision rule induction is not significant (Wilcoxon signed rank test p-value is 0.47). Reducing shortening factor gives the D^3RJ method advantage over both other approaches.

Table 4 presents detailed experimental results of D^3RJ method with use of template evaluation function $q = w{\cdot}h{\cdot}G^8$ and shortening factor $\alpha = 0.9$. Similarly to the previous table the results are presented for the decomposition method without decision rule joining as well as with the decision rule joining. The compression and improvement factors are also provided. The D^3RJ method using the $w{\cdot}h{\cdot}G^8$ criterion in the decomposition phase achieves similar results to the D^3RJ method using the $w{\cdot}h{\cdot}G$. As we can see, the decision rule joining significantly reduces model complexity and improves its predictive abilities. In this case the classification accuracy improvement is significant (Wilcoxon signed rank test p-value is 0.001), but the worsening in comparison to classic decision rule induction is not significant (Wilcoxon signed rank test p-value is 0.39). The D^3RJ method performs quite well requiring almost three times less decision rules then the decomposition method without rule joining and almost eleven times less then classic decision rule induction.

Table 4. The detailed empirical evaluation of the D^{3}RJ method using the shortening factor $\alpha = 0.9$, and template evaluation function $q = w{\cdot}h{\cdot}G^8$.

Table	Before joining		After joining		Profits	
	Accuracy	# Rules	Accuracy	# Rules	Compression	Improvement
att	55.20 ±2.21	2275.8 ±32.88	57.48 ±5.34	612.2 ±8.61	3.72	2.28
ech	67.34 ±9.94	157.3 ±33.67	65.58 ±7.09	58.1 ±9.51	2.71	-1.76
edu	47.72 ±5.09	4080.2 ±56.81	53.10 ±2.76	427.0 ±23.32	9.56	5.38
hco	81.79 ±6.26	2126.9 ±120.15	82.60 ±6.05	593.4 ±71.54	3.58	0.81
hep	79.46 ±4.93	758.4 ±177.42	81.41 ±5.99	611.1 ±162.98	1.24	1.95
hin	68.30 ±3.35	589.9 ±19.25	67.90 ±3.76	358.8 ±13.98	1.64	-0.40
hur2	76.10 ±7.86	77.3 ±12.02	74.69 ±11.72	31.7 ±7.58	2.44	-1.41
hyp	95.23 ±0.09	562.5 ±46.01	95.26 ±0.13	169.3 ±11.19	3.32	0.03
inf2	64.75 ±8.20	4854.0 ±488.92	66.08 ±8.96	4412.6 ±389.21	1.10	1.33
pid2	72.53 ±5.27	1953.7 ±136.74	71.09 ±5.16	149.9 ±13.81	13.03	-1.44
smo2	55.97 ±2.38	7897.4 ±57.71	67.81 ±0.99	1455.0 ±21.06	5.43	11.84
avg	69.49 ±5.05	2303.0 ±107.42	71.18 ±5.27	807.2 ±66.62	2.85	1.69

9 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 obtain a simple strategy for building decision systems for data tables with missing attribute values. Although the obtained decision rules are generated only for complete data, they are able to classify data with missing attribute values. It is done without using the missing values explicitly in the decision rule formula. For bigger decision tables the proposed approach works faster than one-pass classic decision rule induction. 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 further advantage from the decision rule set reduction is reduction of time necessary for classification of test objects. The obtained results showed that the presented method is very promising for classification problems with missing attribute values in data sets.

Acknowledgments

The authors would like to thank professor Andrzej Skowron for his support while writing this paper. The research has been supported by the grant 3T11C00226 from Ministry of Scientific Research and Information Technology of the Republic of Poland.

References

1. Ågotnes, T.: Filtering large propositional rule sets while retaining classifier performance. Master's thesis, Department of Computer and Information Science, Norwegian University of Science (1999)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In Bocca, J.B., Jarke, M., Zaniolo, C., eds.: VLDB'94, Morgan Kaufmann (1994) 487–499
3. 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)
4. Bazan, J.G.: A comparison of dynamic and non-dynamic rough set methods for extracting laws from decision table. [43] 321–365
5. Bazan, J.G.: Discovery of decision rules by matching new objects against data tables. In Polkowski, L., Skowron, A., eds.: Rough Sets and Current Trends in Computing, RSCTC'98. LNCS 1424, Springer (1998) 521–528
6. Bazan, J.G., Szczuka, M.S., Wróblewski, J.: A new version of rough set exploration system. [3] 397–404
7. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, P.J.: Classification and Regression Trees. Wadsworth International Group (1984)
8. Feige, U.: A threshold of $\ln n$ for approximating set cover (preliminary version). In: Proceedings of the 28th ACM Symposium on the Theory of Computing, ACM (1996) 314–318
9. Friedman, J.H.: A recursive partitioning decision rule for non-parametric classification. IEEE Trasactions on Computer Science **26** (1977) 404–408
10. Friedman, J.H., Kohavi, R., Yun, Y.: Lazy decision trees. In Shrobe, H., Senator, T., eds.: Proceedings of the AAAI96 and IAAI96. Volume 1., AAAI Press / The MIT Press (1996) 717–724
11. Fujikawa, Y., Ho, T.B.: Cluster-based algorithms for filling missing values. In: Proceedings of PAKDD-2002. LNCS 2336, Springer (2002) 549–554
12. Gago, P., Bento, C.: A metric for selection of the most promising rules. [62] 19–27
13. Ghahramani, Z., Jordan, M.I.: Supervised learning from incomplete data via an EM approach. In Cowan, J.D., Tesauro, G., Alspector, J., eds.: Advances in Neural Information Processing Systems. Volume 6., Morgan Kaufmann (1994) 120–127
14. Greco, S., Matarazzo, B., Słowiński, R.: Handling missing values in rough set analysis of multi-attribute and multi-criteria decision problems. [59] 146–157
15. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets processing of vague information using fuzzy similarity relations. In Caldue, C.S., Paun, G., eds.: Finite vs. infinite: contribution to an eternal dilemma, Berlin, Springer (2000) 149–173
16. Greco, S., Matarazzo, B., Słowiński, R., Zanakis, S.: Rough set analysis of information tables with missing values. In: Proceedings of 5th International Conference Decision Sciences Institute. Volume 2. (1999) 1359–1362
17. Grzymała-Busse, J.W.: Lers–a system for learning from examples based on rough sets. In Słowinski, R., ed.: Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, Kluwer (1992) 3–18
18. Grzymała-Busse, J.W., Grzymała-Busse, W.J., Goodwin, L.K.: A closest fit approach to missing attribute values in preterm birth data. [59] 405–413
19. Grzymała-Busse, J.W., Hu, M.: A comparison of several approaches to missing attribute values in data mining. [61] 378–385

20. Grzymała-Busse, J.W., Wang, A.Y.: Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In: Proceedings of RSSC'97 at the 3rd Joint Conference on Information Sciences. (1997) 69–72
21. Hipp, J., Myka, A., Wirth, R., Güntzer, U.: A new algorithm for faster mining of generalized association rules. [62] 74–82
22. Johnson, D.S.: Approximation algorithms for combinatorial problems. Journal of Computer and System Sciences **9** (1974) 256–278
23. 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
24. Kononenko, I., Bratko, I., Roškar, E.: Experiments in automatic learning of medical diagnostic rules. Technical report, Jozef Stefan Institute, Ljubljana (1984)
25. Kryszkiewicz, M.: Properties of incomplete information systems in the framework of rough sets. [43] 422–450
26. Latkowski, R.: Application of data decomposition to incomplete information systems. In Kłopotek, M.A., Wierzchoń, S.T., eds.: Proceedings of the International Symposium "Intelligent Information Systems XI", Physica-Verlag (2002)
27. Latkowski, R.: Incomplete data decomposition for classification. [3] 413–420
28. Latkowski, R.: High computational complexity of the decision tree induction with many missing attribute values. In Czaja, L., ed.: Proceedings of CS&P'2003, Czarna, September 25-27, Volume 2., Zakłady Graficzne UW (2003) 318–325
29. Latkowski, R.: On decomposition for incomplete data. Fundamenta Informaticae **54** (2003) 1–16
30. Latkowski, R., Mikołajczyk, M.: Data Decomposition and Decision Rule Joining for Classification of Data with Missing Values In Tsumoto, S., Komorowski, J., Grzymała-Busse, J.W., Słowiński, R., eds.: Rough Sets and Current Trends in Computing, RSCTC'2004, Springer (2004)
31. Lim, T.: Missing covariate values and classification trees. http://www.recursive-partitioning.com/mv.shtml, Recursive-Partitioning.com (2000)
32. Lovasz, L.: On the ratio of optimal integral and fractional covers. Discrete Mathematics **13** (1975) 383–390
33. Mikołajczyk, M.: Reducing number of decision rules by joining. [3] 425–432
34. 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
35. Nguyen, H.S., Nguyen, S.H.: Rough sets and association rule generation. Fundamenta Informaticae **40** (1999) 383–405
36. Nguyen, H.S., Ślęzak, D.: Approximate reducts and association rules — correspondence and complexity results. [59] 137–145
37. Nguyen, S.H.: Regularity Analysis and its Application in Data Mining. PhD thesis, Warsaw University, Institute of Computer Science (1999)
38. 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
39. Nigmatullin, R.G.: Method of steepest descent in problems on cover. In: Memoirs of Symposium Problems of Precision and Efficiency of Computer Algorithms. Volume 5., Kiev (1969) 116–126
40. Øhrn, A., Ohno-Machado, L., Rowland, T.: Building manageable rough set classifiers. In Chute, C.G., ed.: Proceedings of the 1998 AMIA Annual Symposium. (1998) 543–547

41. Pal, S.K., Polkowski, L., Skowron, A., eds.: Rough-Neural Computing: Techniques for Computing with Words. Springer (2004)
42. Pawlak, Z.: Rough sets: Theoretical aspects of reasoning about data. Kluwer, Dordrecht (1991)
43. Polkowski, L., Skowron, A., eds.: Rough Sets in Knowledge Discovery 1: Methodology and Applications. Physica-Verlag (1998)
44. Polkowski, L., Skowron, A., Żytkow, J.M.: Tolerance based rough sets. In Lin, T.Y., Wildberger, A.M., eds.: Soft Computing, San Diego Simulation Councils Inc. (1995) 55–58
45. Quinlan, J.R.: Unknown attribute values in induction. In Segre, A.M., ed.: Proceedings of the Sixth International Machine Learning Workshop, Morgan Kaufmann (1989) 31–37
46. Rubin, D.B.: Multiple Imputation for Nonresponse in Surveys. John Wiley & Sons, New York (1987)
47. Skowron, A.: Boolean reasoning for decision rules generation. In Komorowski, H.J., Raś, Z.W., eds.: ISMIS 1993. LNCS 689, Springer (1993) 295–305
48. 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
49. Słowiński, R., Vanderpooten, D.: A generalized definition of rough approximations based on similarity. IEEE Transactions on Data and Knowledge Engineering **12** (2000) 331–336
50. Stefanowski, J.: On rough set based approaches to induction of decision rules. [43] 500–529
51. Stefanowski, J., Tsoukiàs, A.: On the extension of rough sets under incomplete information. [59] 73–81
52. Stefanowski, J., Tsoukiàs, A.: Incomplete information tables and rough classification. International Journal of Computational Intelligence **17** (2001) 545–566
53. Stefanowski, J., Tsoukiàs, A.: Valued tolerance and decision rules. [61] 212–219
54. Swets, J.A.: Measuring the accuracy of diagnostic systems. Science **240** (1988) 1285–1293
55. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
56. Wang, H., Düntsh, I., Gediga, G., Skowron, A.: Hyperrelations in version space. Journal of Approximate Reasoning **36** (2004)
57. Weiss, S.M., Indurkhya, N.: Lightweight rule induction. In: Proceedings of the International Conference on Machine Learning ICML'2000. (2000)
58. Witten, I.H., Frank, E.: Data Mining: Practical Mashine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann (2000)
59. Zhong, N., Skowron, A., Ohsuga, S., eds.: New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, RSFDGrC '99. LNCS 1711, Springer (1999)
60. Ziarko, W.: Variable precision rough sets model. Journal of Computer and System Sciences **46** (1993) 39–59
61. Ziarko, W., Yao, Y.Y., eds.: Rough Sets and Current Trends in Computing, Second International Conference, RSCTC 2000 Banff, Canada, October 16-19, 2000, Revised Papers. LNCS 2000, Springer (2001)
62. Żytkow, J.M., Quafafou, M., eds.: Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '98, Nantes, France, September 23-26, 1998, Proceedings. LNCS 1510, Springer (1998)