Fundamenta Informaticae 54 (2003) 1–16
IOS Press

On Decomposition for Incomplete Data

Rafał Latkowski
Institute of Computer Science, Warsaw University
ul. Banacha 2, 02–097 Warsaw, Poland
rlatkows@mimuw.edu.pl

Abstract. In this paper we present a method of data decomposition to avoid the necessity of reasoning on data with missing attribute values. This method can be applied to any algorithm of classifier induction. The original incomplete data is decomposed into data subsets without missing values. Next, methods for classifier induction are applied to these sets. Finally, a conflict resolving method is used to obtain final classification from partial classifiers. We provide an empirical evaluation of the decomposition method accuracy and model size with use of various decomposition criteria on data with natural missing values. We present also experiments on data with synthetic missing values to examine the properties of proposed method with variable ratio of incompleteness.

Keywords: Data Mining, Rough Sets, Missing Attribute Values.

1. Introduction

Classification is an important problem in the field of Data Mining. Data acquisition and warehousing capabilities of computer systems are sufficient for wide application of computer aided Knowledge Discovery. Inductive learning is employed in various domains such as medical data analysis or customer activity monitoring. Due to various factors that data suffer from impreciseness and incompleteness. The hard task of dealing with data imperfection in inductive learning methods was addressed in the area of data impreciseness by Pawlak in early 80's [17]. He proposed a *Rough Set* approach that made possible to precisely express facts about imprecise data in a formal way. The main concept of Rough Sets, the *indiscernibility relation*, proved to be very useful for analysis of decision problems concerning objects described in a data table by a set of conditional attributes and a decision attribute [18, 19]. In recent years a great research effort has been made in the area of data incompleteness to develop methods inducing classifiers for data with missing attribute values. Some approaches making possible to handle missing

Address for corespondance: Institute of Computer Science, Warsaw University ul. Banacha 2, 02–097 Warsaw, Poland

attribute values have been developed within the Rough Sets framework [9, 24]. In those approaches a modification of indiscernibility relation is considered to handle missing attribute values. The other approach presented in *LEM1* and *LEM2* methods [6, 7] is to modify an algorithm that search for a set of decision rules. In this paper we present a method of data decomposition to avoid the necessity of reasoning on data with missing attribute values without modification of the inductive learning algorithm itself.

The decomposition method was developed to meet certain assumptions. There exist many well-known classifier induction methods that are initially not capable to handle missing attribute values. The primary aim was to find a possibility to adapt these methods to the case of incomplete data. In other words, we search for a solution, which makes possible to analyze incomplete information systems by already known and implemented classification methods. Such a solution will reduce an effort necessary to construct a new software and framework for such a data analysis. The secondary aim was to cope with the problem of incomplete information systems without making an additional assumption of independent random distribution of missing values and without using data imputation methods [5, 6]. Many real world applications have showed that appearance of missing values is governed by very complicated dependencies. Missing attribute values are frequently not uniformly distributed, but their distribution is determined by the hidden nature of investigated phenomenon, just like in the case of regular values. The application of an arbitrary method for data imputation can reduce accuracy of a classifier.

The decomposition method tries to avoid the necessity of reasoning on data with missing attribute values. The original incomplete data is decomposed into data subsets without missing values. Next, methods for classifier induction are applied to these subsets. Finally, a conflict resolving method is used to obtain final classification from partial classifiers. The rest of this paper is organized as follows. In the second section we describe basic notions used to formalize the decomposition method. Then, in the following section the decomposition method is described. The fourth section describes used decomposition criteria. In the fifth section we provide an empirical evaluation of the decomposition method in comparison to the standard Rough-Set rule induction method (see e.g. [21, 23]) and the decision tree method (see e.g. [13, 14]). The experiments are divided in two groups. The first group consists of the general evaluation of the decomposition method on data with natural missing values. Secondly, the experiments on data with synthetic missing values are presented in order to evaluate the decomposition method with variable ratio of incompleteness. The experiments were carried out with help of RSES-Lib software (see [1]). The sixth section presents final conclusions.

2. Preliminaries

For the classification and the concept approximation problems we are considering a special type of information systems — decision tables $\mathbb{A} = (U, A \cup \{d\})$, where $a_i \in A$, $a_i : U \to V_i$ are conditional attributes and $d : U \to V_d$ is a special attribute called decision. In a presence of missing data we may consider the attributes $a_i \in A$ as functions $a_i : U \to V_i^*$, where $V_i^* = V_i \cup \{*\}$ and $* \notin V_i$. The special symbol "$*$" denotes absence of regular attribute value and if $a_i(x) = *$ we say that a_i is not defined on x. In the relational databases exists a similar notion — "NULL" that represents missing attribute value in a database record. The other area, where the missing values are known, is *universal algebra*. In terms of universal algebra we can interpret an attribute with missing values $a_i : U \to V_i^*$ as a *partial* function in contrast to an attribute without missing values $a_i : U \to V_i$ interpreted as a *total* function. We will use the word „total" to denote the completeness of object description.

To discover the 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 a data regularities are *templates* (cf. [15, 16]). The concept of template requires some modifications to be applicable to the problem of incomplete decision table decomposition.

Definition 2.1. Let $\mathbb{A} = (U, A \cup \{d\})$ 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 $a_i \in A$ for this object u is not missing in $\mathbb{A}$, otherwise the object u does not satisfy total descriptor.

Definition 2.2. Let $\mathbb{A} = (U, A \cup \{d\})$ 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 values of attributes $a_{k_1}, \ldots, a_{k_n} \in A$ for the object u are not missing in $\mathbb{A}$.

Total templates are used to discover regular areas in data that contain no missing values. Once we have a total template, we can identify it with a subtable of original decision table. Such a subtable consists of the decision attribute, all attributes that are elements of total template and it contains all objects that satisfy this template. Obviously, the decision table that corresponds to the total template contains no missing attribute values.

3. Method description

The decomposition method consists of two phases. In the first step the data decomposition is performed. In the second step classifiers are induced and merged with help of a conflict resolving method.

3.1. Decomposition

In the data decomposition phase the original decision table with missing attribute values is partitioned to a number of decision subtables without missing values. This 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 subtables that are result of the decomposition should correspond to natural subproblems of the whole problem domain.

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 subtables to exhaustively cover the input table (at least in the terms of objects). 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 than 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. In the previous section was described the construction of complete subtable that is determined by a total template. With such an assignment we can consider the data decomposition as a problem of covering data table with templates. Then, the preference of particular decomposition can be formally expressed in the terms of total templates. We should construct the template evaluation criterion for templates preferring decision subtables relevant to the approximated concept.

The task of the first phase is now formulated as the problem of covering data with total templates. A similar problem of data decomposition with use of templates is described in [15, 16]. A standard approach to covering data table with templates is to iteratively generate the best template for objects that remains uncovered. This greedy strategy of proceeding is chosen because of a high computational complexity of the original problem and the fact that the results of greedy heuristic are usually satisfactory. The algorithm starts from the full set of objects. Then, the best template is generated according to a certain criterion. All objects that satisfy the generated template are removed and the process is continued until the set of uncovered objects becomes empty. The set of templates generated by this algorithm covers all objects from original decision table. The above algorithm can be slightly modified. One of the most popular modification is to do not remove objects, but instead of that just decrease importance weights for covered objects (see e.g. [4, 25]). We found that such a modification does not have a significant impact on the final classification accuracy. The results were slightly better when the importance weights were drastically reduced for already covered objects, or symmetrically, the weights of uncovered objects were drastically increased (e.g. $w := (1+w)^6$). In the final experiments only the strategy that removes objects was used.

The greedy algorithm of covering data table with templates uses, as its inner step, a procedure that computes the best template. As it will be discussed in detail in the following section, the best template is selected according to a certain criterion. It is assumed that this criterion is expressed as a function that value represents a template "goodness" for the decomposition. The problem of searching for the best template is known to be NP-hard for all interesting goodness criteria[1] (cf. [15]). Therefore also this inner step of the above algorithm cannot be solved directly. We have to use a heuristical approach. In [15, 16] there were proposed very efficient algorithms *MAX I* and *MAX II* for this problem. Nevertheless, for our purposes those algorithms make too strong assumption about the function that represents a goodness criterion[2]. In our experiments we used a dedicated *genetic algorithm* to generate a sub-optimal template (see e.g. [12]). The implemented genetic algorithm uses uniform mutation, uniform crossing, union and intersection of total templates as the genetic variability operators. It uses also the tournament selection and variable population size . The coefficients were experimentally chosen to achieve similar results to the exact (exponential) algorithm (cf. [10]). The algorithm uses a special initialization. All total templates existing in the data (called *total schemes* of objects) are incorporated in the initial population. We believe that for this particular application the proposed algorithm performs not worst than MAX I and MAX II algorithms, but it can not be experimentally checked.

3.2. Merging

Once we have data decomposed into complete decision subtables we can apply any method of inductive learning on them. After such a proceeding we get a number of local classifiers. Each one is designated to predict the class-membership of objects that satisfy a total template related with this classifier. It is possible that some objects satisfy more than one total template. The subtables indeed usually have non-empty intersections, since in the decomposition phase we do not require the total templates to be

[1] There are some trivial criteria that make this problem P-Time, but they find little application.

[2] In the MAX I and MAX II algorithms the function of template evaluation is based only of the template width and the template height.

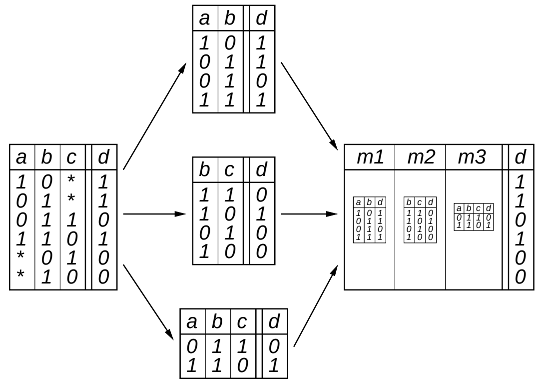

Figure 1. The decomposition method. Original incomplete data is decomposed into complete subtables. Then, a conflict resolving method is applied.

disjointed or to create disjoint partitions of data. This leads to the problem of multiply answers for one object. The second phase of the decomposition method tries to combine the partial answers to obtain the global one.

The problem of joining multiply answers into one is an example of conflict resolving problem. The very standard proceeding to such kind of problems is to apply one of the voting strategies. The simplest voting strategy is called simple or majority voting. Each classifier selects (votes) a decision as its answer. All classifiers have an equal impact to the final decision and the decision that receives the most of votes becomes the selected answer. A more sophisticated strategy of voting is the weighted voting. Each classifier selects one or more decisions as its answer. With each decision is related a confidence factor that reflects the certainty of taken decision. Usually, the confidence factor is selected from the interval $[0, 1]$. For each decision the confidence weights are summed and wins the decision with highest sum of confidences.

There are some disadvantages related with this standard approach to the conflict resolving. Both presented methods are universal, which means that they are immediately applicable to any instance of conflict resolving problem. It is commonly observed that the more specific or customized methods perform better than the general ones (see e.g. [12]). The second disadvantage is specific to the decomposition method. The decision subtables contain usually less attributes than the original data. This often alters the *indiscernibility* relation and finally reduces the *positive region* of decision subtable [8, 17]. The positive region reduction leads to systematic errors made by a classifier due to inconsistent training data. The voting methods perform the decision-making in straightforward, linear-like way. It is easy to imagine the situations, in which such a conflict resolving is not enough.

The conflict resolving problem itself resembles the concept approximation. Thus, we can apply inductive learning methods to merge partial answers into global one. As a training data should be taken a decision table made from partial answers. The objects in this decision table correspond to the objects in original data. The conditional attributes correspond to answers of induced local classifiers. If the object satisfies a total template that corresponds to a classifier, the value of an attribute related with the classifier is the decision of this classifier. Otherwise the value is “no decision”. The decision attribute d is taken without any modifications from original decision table. It represents the desirable decision of a conflict resolving method. A classifier induced over a decision table constructed in this way can be used as the conflict resolver.

The presented above special construction of the table used for conflict resolving introduce a special value called “no decision”. This value could be viewed as the missing value and indeed represents the absence of a regular answer. From the other hand it is our strong believe that the discovered regularities in the distribution of missing attribute values are relevant to the inductive learning. Thus, “no decision” is not a lack of information, but contains information related with properties of this object as well as the group of similar objects. This point of view assumes that the “no decision” cannot be interpreted as the missing attribute value and should be treated as a usual property of investigated objects. In presented implementation the “no decision” value was encoded as a next, regular domain value.

Briefly we can summarize the decomposition method as follows:

- Create a temporary set $\mathbb{T}$ of objects from the original decision table and repeat two following steps until the temporary set $\mathbb{T}$ becomes empty:
  - Generate the best total template according to a chosen criterion;
  - Remove from the temporary set $\mathbb{T}$ objects that are covered by generated template;
- Create complete decision subtables that correspond to generated set of templates;
- Induce classifiers over complete decision subtables;
- Apply a conflict resolving method (or learn a conflict resolving strategy) to get the final answer.

4. Decomposition criteria

A common approach to measure adequacy of a template for a particular task is to define a function $q(t)$ which describes an overall quality of template t with respect to the considered task. In our case the quality measures adequacy of a template for decomposition of a particular data set. Then, the best template is understood as a template with the best value of such a quality function (cf. [15]). To achieve good results we should select quality function $q(t)$ very carefully and in accordance to nature of the optimized problem.

A standard approach to measure template quality is to define a quality function using *width* and *height* of a template [15, 16]. The *template height* is the number of objects that satisfy a template and the *template width* is the number of attributes that are elements of a template. To obtain a quality function $q(t)$ of a template we have to combine width and height to get one value. A usual formula that combines these two factors is

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

We can also add a simple mechanism to control the importance of each factor

$$q_1(t) = w(t)^{\alpha} \cdot h(t), \tag{2}$$

where $\alpha > 0$. If we apply $\alpha > 1$ the importance of the width, thus importance of the size of available object description, increases and the number of necessary templates to cover original decision table is

Table 1. Comparison of the number of subtables (templates) used in the decomposition method. The numbers are averages over ten Cross-Validation folds with their standard deviations.

Table	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot H$	$w \cdot h \cdot H^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$
att	3.9 ±0.03	4.9 ±0.28	11.9 ±0.03	2.0 ±0.00	2.0 ±0.00	4.0 ±0.00	4.0 ±0.63
ech	3.6 ±0.19	3.9 ±0.03	4.2 ±0.25	1.6 ±0.19	1.0 ±0.00	2.9 ±0.03	3.3 ±0.09
edu	3.0 ±0.00	4.0 ±0.00	7.9 ±0.28	1.0 ±0.00	2.8 ±1.01	3.1 ±0.03	6.9 ±0.28
hco	6.0 ±0.00	8.4 ±0.13	11.5 ±0.47	3.2 ±0.57	2.8 ±0.06	6.4 ±0.19	10.2 ±0.25
hep	3.9 ±0.28	3.9 ±0.28	3.9 ±0.28	3.2 ±0.70	2.6 ±0.19	4.2 ±0.38	5.6 ±0.82
hin	3.9 ±0.28	19.8 ±0.57	27.0 ±0.32	3.2 ±0.06	5.9 ±0.03	4.7 ±0.22	7.8 ±0.06
hur2	2.0 ±0.00	2.0 ±0.00	1.0 ±0.00	2.0 ±0.00	1.9 ±0.35	2.0 ±0.00	2.4 ±0.13
hyp	2.0 ±0.00	6.9 ±0.03	8.7 ±0.22	1.0 ±0.00	1.0 ±0.00	2.0 ±0.00	2.1 ±0.03
inf2	2.0 ±0.00	2.0 ±0.00	2.1 ±0.03	1.2 ±0.25	1.1 ±0.28	2.0 ±0.00	2.8 ±0.06
pid2	3.0 ±0.00	3.0 ±0.00	2.0 ±0.00	3.0 ±0.00	2.8 ±0.06	2.7 ±0.09	3.3 ±0.09
rcd128	9.7 ±0.54	7.1 ±0.28	6.1 ±0.03	30.2 ±0.70	54.2 ±1.83	8.9 ±0.35	9.5 ±0.79
smo2	2.0 ±0.00	3.0 ±0.00	3.0 ±0.00	1.0 ±0.00	1.0 ±0.00	2.1 ±0.03	1.5 ±0.16

higher. The empirical results showed, however, that α does not have as significant an impact on overall classification quality as the further introduced coefficient β.

The quality function based only on width and height is not always enough to classify objects by the decomposition method better than by the method with native missing attribute values handling. The empirical evaluation demonstrated that in data exist many templates with similar width and height, but with different potential for the data decomposition.

We can estimate the template quality by evaluating homogeneity of the indiscernibility classes induced by this template. Such a measure should correspond to the quality of the classification by *prime implicants* (see e.g. [8]). We used two measures for estimating the template quality:

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

$$H(t) = \frac{\sum_{i=1}^{K} max_{c \in V_d} card(\{y \in [x^i]_{IND} : d(y) = c\})}{K \cdot L}, \tag{4}$$

where K is the number of indiscernibility classes $[x^1]_{IND}, \ldots, [x^K]_{IND}$ and L is the size of subtable. We can easily incorporate these factors into the quality function

$$q_2(t) = w(t)^\alpha \cdot h(t) \cdot G(t)^\beta, \;\; q_3(t) = w(t)^\alpha \cdot h(t) \cdot H(t)^\beta, \tag{5}$$

where β controls the influence of factors to the whole quality value.

The next measure is similar to the *wrapper* approach in the feature selection (see e.g. [3]). Instead of estimating the template quality we can use the predictive accuracy of the data subset. The classifier itself is executed on decision subtable determined by the total template and the number of correct answers is counted.

$$P(t) = \frac{\textit{number of correct answers}}{\textit{number of objects}}. \tag{6}$$

Table 2. Comparison of the decomposition method that uses various template evaluation criteria with the rule induction method. The numbers are averages over ten Cross-Validation folds with their standard deviations and the best classification accuracy for each table is marked with bold face.

Table	-	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot H$	$w \cdot h \cdot H^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$
att	56.10 ±4.37	60.10 ±4.05	60.00 ±5.17	58.40 ±5.18	**79.50** ±18.17	78.90 ±18.01	60.30 ±4.05	61.00 ±3.89
ech	58.02 ±9.85	62.60 ±11.16	64.12 ±9.69	67.18 ±9.59	64.89 ±7.52	67.18 ±2.16	64.12 ±11.82	**72.52** ±11.86
edu	41.60 ±4.64	47.40 ±5.53	49.10 ±4.16	49.70 ±4.29	**53.90** ±0.40	53.60 ±0.97	46.40 ±5.54	51.10 ±4.44
hco	**81.25** ±8.04	79.08 ±4.96	79.35 ±8.59	80.71 ±6.03	65.49 ±5.38	64.67 ±4.62	79.89 ±5.52	80.98 ±5.70
hep	82.58 ±7.38	78.71 ±9.57	76.13 ±8.49	75.48 ±9.15	81.29 ±7.19	**83.23** ±5.47	79.35 ±9.45	81.94 ±5.37
hin	53.70 ±3.09	65.00 ±4.87	66.90 ±5.17	66.80 ±4.12	56.90 ±4.56	57.90 ±5.78	**67.70** ±3.69	67.10 ±3.21
hur2	77.99 ±7.89	77.99 ±9.04	77.99 ±9.04	77.99 ±9.04	77.99 ±9.04	71.77 ±9.72	78.47 ±8.14	**81.82** ±9.37
hyp	98.10 ±0.78	97.47 ±1.09	97.60 ±0.85	97.63 ±0.86	95.23 ±0.09	95.23 ±0.09	98.74 ±0.65	**98.77** ±0.64
inf2	64.71 ±7.89	59.24 ±6.91	60.50 ±8.12	59.24 ±6.71	42.44 ±4.92	39.50 ±3.94	64.29 ±11.88	**65.97** ±10.28
pid2	68.75 ±6.84	**71.22** ±5.93	70.31 ±4.63	63.80 ±6.66	69.53 ±5.06	69.40 ±3.08	70.31 ±6.42	69.79 ±5.08
rcd128		76.69 ±1.73	74.76 ±5.99	75.74 ±2.78	76.73 ±18.79	**82.77** ±5.31	77.50 ±2.14	77.44 ±1.75
smo2	52.43 ±3.06	54.54 ±2.37	57.41 ±2.41	57.41 ±2.41	**69.53** ±0.10	**69.53** ±0.10	53.13 ±2.34	**69.53** ±0.10

Also this factor can be easily incorporated into the quality function

$$q_4(t) = w(t)^\alpha \cdot h(t) \cdot P(t)^\beta, \tag{7}$$

where $\beta > 0$ controls the influence of predictive accuracy factor to the whole quality value.

5. Empirical Evaluation

There were carried out some experiments in order to evaluate various aspects of the decomposition method. In these experiments a genetic algorithm was used for generation of the best template with respect to the selected decomposition criterion. Results were obtained from the ten-fold Cross-Validation (CV10) evaluation. The experiments were performed with different decomposition approaches as well as without using decomposition method at all. Almost all data sets utilized in evaluation of the decomposition method were taken from *Recursive-Partitioning.com* [11]. The RoboCup Soccer visual data were extracted from the log files of RoboCup Soccer Server [2].

- att — AT&T telemarketing data, 2 classes, 5 numerical attribute, 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.

Table 3. Comparison of the decomposition method that use various template evaluation criteria with the decision tree method. The numbers are averages over ten Cross-Validation folds with their standard deviations and the best classification accuracy for each table is marked with bold face.

Table	–	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot H$	$w \cdot h \cdot H^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$
att	58.80 ±5.79	58.70 ±4.53	56.60 ±4.41	55.50 ±3.21	59.70 ±4.13	60.00 ±4.14	60.70 ±4.56	**69.40** ±4.39
ech	59.54 ±12.88	64.89 ±12.03	64.12 ±10.15	61.07 ±11.91	65.65 ±6.58	63.36 ±6.32	77.10 ±7.50	**81.68** ±8.20
edu	51.60 ±3.51	51.90 ±3.14	53.60 ±3.48	54.40 ±3.15	53.90 ±0.40	53.90 ±0.40	53.60 ±3.19	**59.60** ±2.94
hco	78.80 ±7.54	81.79 ±9.13	81.79 ±8.69	82.88 ±7.17	68.75 ±6.44	66.58 ±8.65	84.24 ±6.24	**91.03** ±7.50
hep	79.35 ±9.02	78.06 ±7.37	77.42 ±7.56	75.48 ±7.43	81.94 ±5.25	81.29 ±4.01	87.10 ±4.92	**93.55** ±5.17
hin	64.50 ±2.44	69.10 ±4.77	73.20 ±4.35	72.80 ±5.22	65.80 ±3.62	70.60 ±3.36	71.30 ±5.19	**74.20** ±3.20
hur2	78.47 ±7.04	82.78 ±8.01	82.78 ±8.01	80.38 ±9.12	82.78 ±8.01	68.90 ±11.52	83.25 ±7.25	**89.47** ±5.44
hyp	**98.83** ±0.49	97.34 ±0.97	95.51 ±0.13	97.19 ±0.55	95.23 ±0.09	95.23 ±0.09	98.70 ±0.52	98.48 ±1.06
inf2	63.87 ±7.13	63.45 ±7.13	63.03 ±7.13	62.61 ±7.32	41.18 ±3.06	40.34 ±3.35	68.91 ±10.88	**82.35** ±7.24
pid2	68.75 ±5.49	69.92 ±5.87	71.09 ±5.77	65.10 ±5.22	70.05 ±3.55	68.10 ±3.61	71.09 ±6.58	**79.17** ±2.73
rcd128		86.12 ±4.26	82.50 ±6.81	75.39 ±4.90	78.34 ±12.78	81.90 ±6.07	**88.11** ±1.55	86.53 ±2.55
smo2	60.28 ±2.13	55.13 ±1.93	61.93 ±2.75	61.93 ±2.75	**69.53** ±0.10	**69.53** ±0.10	58.04 ±2.60	69.46 ±0.73

- 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.
- rcd128 — RoboCup Soccer visual data, prediction of the head direction, 128 classes, 26 numerical attributes, 20000 observations, 100.0% incomplete cases, 75.1% missing values. With this data set it was impossible to compute the results without use of the decomposition method due to the enormous number of missing values. This is the result of the fact that the missing values satisfy all tests in decision and discretization trees (see e.g. [20] for details).
- 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 experiments a rule induction method and a decision tree method that base on conflict (discernibility) measure were used to induce classifiers from the decision subtables. Both methods are implemented in the *RSES-Lib* software (see [1]). These methods were chosen in order that we could

Table 4. Comparison of the number of induced rules induced with various template evaluators and without the decomposition method at all. The smallest model for each table is marked with bold face.

Table	-	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot H$	$w \cdot h \cdot H^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$
att	3241.0	5921.6	5764.3	13594.6	**0.0**	**0.0**	5933.1	2693.9
ech	304.6	450.7	466.3	392.6	122.8	**0.0**	308.0	263.2
edu	5199.5	4377.8	6547.2	10705.0	**0.0**	49.8	4333.3	4644.2
hco	13370.2	5138.7	3073.4	3585.3	2269.2	**1395.4**	4458.7	7475.1
hep	5016.3	3326.9	3035.3	**1586.0**	2556.7	1844.7	3453.8	3479.1
hin	736.0	309.9	2898.3	4853.2	**7.9**	14.2	540.0	531.8
hur2	363.2	293.8	293.8	**148.2**	293.8	192.8	302.2	297.1
hyp	11374.7	921.0	2827.8	3683.7	3.8	**1.4**	640.7	632.4
inf2	5629.0	6725.2	6725.2	7146.5	3.6	**2.0**	4659.5	2478.5
pid2	5643.6	9015.8	8631.5	**3821.7**	6664.1	5619.7	7670.9	5324.2
rcd128		12552.8	9130.6	8958.0	8316.5	**6204.1**	10040.9	10569.0
smo2	10746.8	11892.9	12490.2	12490.2	**0.0**	**0.0**	9508.0	139.1

compare several approaches to missing attribute values handling. We compare the standard approach to ignore discernibility on missing values (implemented in RSES-lib) with the decomposition method that does not relay on any other missing attribute values handling. The empirical evaluation provided by Grzymała-Busse in [6] (for decision rules) and Quinlan in [20] (for decision trees) showed that assuming indiscernibility of a missing value with any other one is a very effective mechanism for missing attribute values handling.

We used a simple voting mechanism and the decision tree method to resolve conflicts between partial answers. The initial experiments showed that application of voting for the decomposition method based on rule induction is not enough to achieve good results. Partially this is a consequence of *positive region* (see [8, 22]) reduction in subtables of original data, as described in section 3.2. The expressiveness of a regular classifier makes it possible to combine partial answers induced from inconsistent decision subtables in much more sophisticated way.

5.1. General experiments

The aim of the first group of experiments is to test the decomposition method on the standard data sets that originally contain missing attribute values. The experiments were carried out with various template evaluation functions and also without decomposition.

Table 1 presents the average number of generated templates. The number of templates corresponds to the number of subclassifiers that are a result of the data decomposition. The number of templates remains in reasonable limits and very rarely exceeds 10. As we can compare with two following tables there is no strong general correlation between the number of templates and classifier accuracy. For some data sets a better classification is related to the increase of the number of templates, while for the other data sets a better accuracy is achieved without any increase of the number of templates.

During experiments we tested also a hypothesis that huge number of subclassifiers can, from statisti-

Table 5. Comparison of the number of leaves in induced decision trees with various template evaluators and without the decomposition method at all. The smallest model is marked with bold face.

Table	-	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot H$	$w \cdot h \cdot H^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$
att	494.7	911.2	1194.2	3100.5	3.4	**3.0**	941.0	657.7
ech	43.0	92.8	102.7	121.4	15.1	**2.0**	77.1	91.7
edu	515.0	450.3	735.3	1626.1	**2.0**	7.5	433.6	562.1
hco	174.9	153.5	348.0	530.6	17.6	**13.4**	185.5	249.2
hep	28.7	91.1	93.3	107.5	20.1	**13.0**	96.4	119.6
hin	234.2	242.5	1826.4	2816.2	**11.7**	24.3	343.7	445.9
hur2	40.2	52.3	52.3	**31.3**	52.3	76.1	52.5	67.8
hyp	48.6	86.8	773.3	883.7	2.3	**2.0**	58.8	62.6
inf2	58.0	139.9	139.9	146.6	3.4	**2.1**	132.6	171.2
pid2	153.3	352.0	400.2	294.0	199.2	**148.9**	333.3	394.3
rcd128		1458.9	1232.6	**1165.4**	1324.3	1621.3	1240.3	1269.7
smo2	1278.9	1536.7	2424.9	2424.9	**2.0**	**2.0**	1607.1	196.7

cal point of view, increase the accuracy of a compound classifier. In this case, however, it does not seem to be justified. The classifier accuracy decreases significantly, when we are taking not only carefully selected templates, but also a big number of others, randomly selected or all possible templates.

Tables 2 and 3 present classification results of the decomposition method. In the first column there are results of classification without the decomposition method. In the following columns the results of the decomposition method are presented with various template quality function, as described in the header of each column. The table 2 presents results of the rule induction based method with the decision tree used for conflict resolving. In the table 3 there are the results of the decision tree based method with the simple voting mechanism used for conflict resolving. The decision tree utilized as a conflict resolving method and as a partial classifier induction uses conflict measure for evaluation of the tests in tree nodes.

The decomposition method performs better than the both methods without decomposition, especially when the predictive quality is included in the template quality function. We should consider that evaluation of the predictive quality is very time-consuming, even in spite of partial result caching and other optimizations. The functions G and H based on indiscernibility are much easier to compute, however, the results not always overcome these obtained without the decomposition method. From the other hand, the data decomposition is usually computed only once and even after small data modifications (e.g. data increment) it remains nearly optimal. Therefore we can spend some time once to achieve better results many times.

Table 4 presents the average number of induced rules[3] and table 5 presents the average number of leaves in induced decision tree. The number of induced rules and leaves was computed by summing, respectively, numbers of induced rules and leaves in all partial subclassifiers. This number can be interpreted as a size of a model describing a hypothesis. We expect the decomposition done in according to data regularities to decrease complexity of induced model, so also its size. We should take into consid-

[3]For completely inconsistent decision tables (i.e. very narrow ones) the number of induced rules is 0 and the decision is made only by the conflict resolving method (majority class).

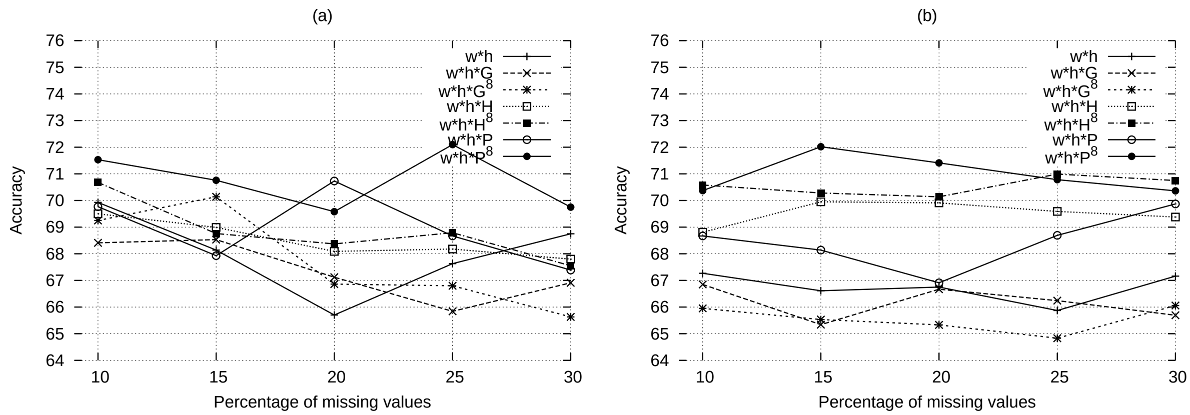

Figure 2. Classification accuracy of the calibration experiments with rule induction method used as the partial classifier. On the first diagram (a) are results with random missing values adding, while on the second diagram (b) are results with template-based missing values adding. The results are averaged over six data sets with 10%, 15%, 20%, 25% and 30% of missing attribute values.

eration that in the decomposition method one object is usually covered by more than one template. This could potentially yield to the great increase of the model size with use of the decomposition method. As we can observe, the number of rules and leaves induced with the decomposition method is frequently smaller than the number of rules induced without the decomposition at all. What is more important, this model size reduction is frequently related with improvement of classification accuracy. In these cases we can certainly say that the decomposition was performed in according to data regularities and it helped in construction of better concept hypothesis.

5.2. Calibration experiments

The aim of the second group of experiments is to present the properties of the decomposition method with variable ratio of missing attribute values. The results of these experiments can also be viewed the experimental justification of the main assumptions that we made in the decomposition method.

It is not feasible to obtain the standard and freely available data sets that contain variable number of missing attribute values of a natural origin. Because of that, we had to add artificially the missing attribute values by a pseudo-random algorithm. There were constructed two completely different algorithms to add the missing values. The first algorithm represents a bad approach to adding missing values. The values were removed completely at random, up to the required number of missing attribute values. The second algorithm performed the template-based removal. At the beginning all existing total templates (schemes) were extracted from a data set. With each total scheme its frequency in original data was related. On the basis of those frequencies the algorithm constructs the empirical distribution of total schemes. Then, each total scheme was selected according to the empirical distribution and the algorithm tried to apply this total scheme to a randomly selected object. The application was successful, when the selected object satisfied selected total scheme (template). In other words, the selected object should not have missing values at the places, where the original object, from which a total scheme was taken, did not have ones. In the case of successful application the values of attributes were removed, but only those

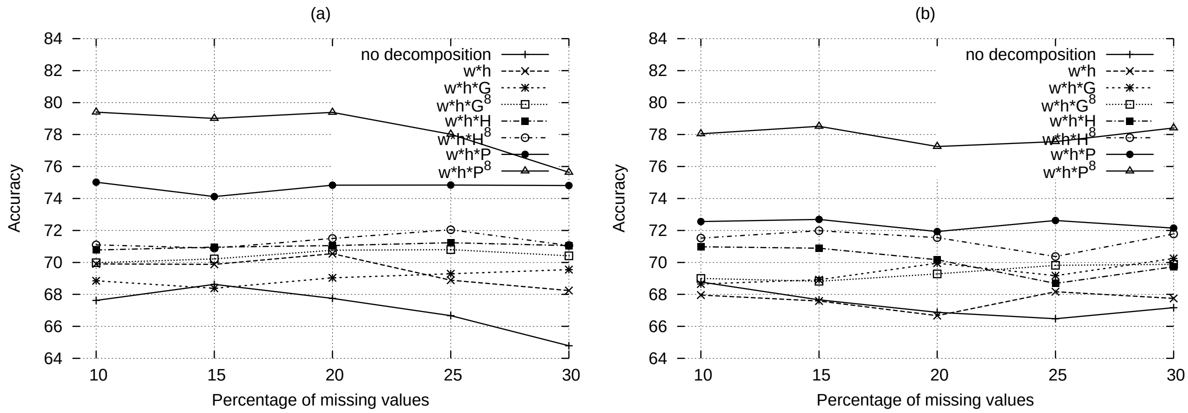

Figure 3. Classification accuracy of the calibration experiments with decision tree method used as the partial classifier. On the first diagram (a) are results with random missing values adding, while on the second diagram (b) are results with template-based missing values adding. The results are averaged over six data sets with 10%, 15%, 20%, 25% and 30% of missing attribute values.

that were not included in total descriptors of selected total scheme. Such an algorithm, although is not ideal, simulates the missing attribute values adding, according to the nature of investigated problem. Of course some data artifacts are created and distribution of missing values does not remind the original one with subsequent applications of removal algorithm. In each application of template-based removal algorithm we used only initial frequencies of total schemes to minimize this effect, but the main problem remains unsolvable.

The results of experiments with 10%, 15%, 20%, 25% and 30% of missing attribute values are presented in the figures 2 and 3. The figure 2 corresponds to experiments with decision rules used as a partial classifier and a decision tree used for conflict resolving. The figure 3 corresponds to the results of experiments with a decision tree used as a partial classifier and voting method used for conflict resolving. The results are averaged over six data sets, namely: att, ech, hep, hin, hyp and pid2. Only on these data sets it is feasible to get the number of missing values between 10% and 30% by applying of above algorithms. The other data sets either contain significantly more than 10% of missing values or the application of second, template-based removal algorithm was not successful. Main trend of results obtained on the data sets with random addition shows that, when the number of missing values increases, the classification accuracy decreases. From the other hand, the results obtained on the data sets with template-based missing values addition shows that there is no general trend of classification accuracy reduction when the number of missing values increases. This trend is more evident in results where rule induction method was used, at the figure 2. In the table 6 the summary of classification accuracy is provided. The classification methods that base on decision trees are denoted with word “Trees”, while the classification methods that base on rule induction are denoted with “Rules”. Similarly, the two described above algorithms of missing values adding are denoted with “Random” (random removal) and “Template” (template-based removal). The differences presented in this table shows clearly, that the degradation of classification accuracy on data with random missing values adding is higher. If we compare the degradation of results without the decomposition method (first column of numbers), the

Table 6. Decrease of the classification accuracy. The classification accuracies on data with 30% of missing values were subtracted from the classification accuracies on data with 10% of missing values.

	-	$w \cdot h$	$w \cdot h \cdot G$	$w \cdot h \cdot G^8$	$w \cdot h \cdot H$	$w \cdot h \cdot H^8$	$w \cdot h \cdot P$	$w \cdot h \cdot P^8$
Trees, Random	-2.83%	-1.67%	0.70%	0.43%	0.27%	-0.04%	-0.21%	-3.76%
Trees, Template	-1.61%	-0.21%	1.60%	0.88%	-1.25%	0.26%	-0.41%	0.36%
Rules, Random	-25.92%	-1.18%	-1.50%	-3.62%	-1.70%	-3.13%	-2.38%	-1.78%
Rules, Template	-6.94%	-0.11%	-1.16%	0.11%	0.57%	0.17%	1.20%	-0.01%

proportion of worsening of the results on data with random removal still remains higher.

Tables 7 and 8 show the averaged number of partitions used in the decomposition method over data sets obtained by random missing value addition algorithm and template-based addition respectively. The averages are provided with their standard deviations. The deviation values are relatively high, because the number of templates was averaged over different data sets. As we can observe the number of partition — the number of generated templates — increases when the number of missing values increases on data with random addition. For data sets with template-based missing values addition the numbers of partitions remain in very reasonable limits. We can even observe that for template-based removal the standard deviation is getting lower when the number of missing values increases. Although the data sets after application of the template-based removal algorithm are each time farther from reality, we can notice that the decomposition behaves very stable. Such a stability in the terms of averages and standard deviations can not be found in case of data sets after application random removal algorithm.

The fact that numbers in the table 6 are positive implies that after the removal of some meaningful information, the classification accuracy increases. This is an argument, that there is still room for optimizations in the decomposition method. The most important would be the improvement of template evaluation functions. Also, the computing of the decomposition i.e., generation of set of total templates could be improved. Nevertheless, we should consider that the decomposition method can sometimes induce the final classifier quicker than induction methods alone. The selection of these particular algorithms and functions is motivated by a trade-off between results and computation time.

6. Conclusions

The decomposition method turned out to be an efficient tool for adapting existing methods to deal with missing attribute values in decision tables. It can be applied to any arbitral algorithm for classifier induction to enrich it with capabilities of incomplete information systems processing. The time-consuming predictive quality evaluation can be replaced with easier to compute measures of the template quality. The application of the rule-based inductive learning demonstrated that decomposition can simplify the model describing induced hypothesis. The results of calibration experiments justify our expectations that missing values appearance is strongly related with the whole process of data creation and the regularities discovered in the decomposition method are significant to the inductive learning. The further research will focus on application of rule-based inductive learning with uniform conflict resolving method at the subtables and the whole system level. We believe that decomposition done in accordance to the natural structure of analyzed data can result in classifier close to the common sense reasoning.

Table 7. Average number of partitions in calibration experiments using template-based missing values adding. The numbers are averages over ten Cross-Validation folds with their standard deviations.

	10%	15%	20%	25%	30%
$w \cdot h$	3.50 ±0.84	3.27 ±0.50	3.03 ±1.05	2.48 ±0.85	2.32 ±0.77
$w \cdot h \cdot G$	6.73 ±6.45	5.95 ±5.71	5.48 ±4.73	5.07 ±4.53	4.82 ±4.03
$w \cdot h \cdot G^8$	9.23 ±9.40	8.88 ±9.29	7.98 ±7.94	7.10 ±6.63	6.13 ±5.86
$w \cdot h \cdot H$	2.35 ±0.87	2.48 ±1.11	2.38 ±0.87	2.42 ±0.93	2.33 ±0.89
$w \cdot h \cdot H^8$	3.07 ±2.11	3.65 ±2.99	4.53 ±4.04	4.68 ±3.28	4.60 ±2.95
$w \cdot h \cdot P$	3.50 ±0.91	3.63 ±0.81	3.05 ±1.29	2.82 ±1.17	2.57 ±0.96
$w \cdot h \cdot P^8$	4.58 ±2.15	4.25 ±1.93	4.35 ±2.35	4.17 ±2.46	4.25 ±2.32

Table 8. Average number of partitions in calibration experiments using random missing values adding. The numbers are averages over ten Cross-Validation folds with their standard deviations.

	10%	15%	20%	25%	30%
$w \cdot h$	6.30 ±2.81	7.75 ±2.08	8.13 ±1.75	8.05 ±1.95	8.30 ±1.67
$w \cdot h \cdot G$	13.38 ±9.40	18.15 ±12.96	21.47 ±15.74	24.65 ±17.96	26.78 ±19.61
$w \cdot h \cdot G^8$	19.10 ±16.26	26.43 ±22.44	32.70 ±27.78	37.67 ±31.97	42.90 ±36.87
$w \cdot h \cdot H$	3.37 ±0.39	4.98 ±0.88	5.78 ±1.66	6.70 ±3.10	6.87 ±1.22
$w \cdot h \cdot H^8$	6.17 ±6.35	11.62 ±16.34	15.65 ±25.26	16.38 ±23.93	13.22 ±11.76
$w \cdot h \cdot P$	6.23 ±2.63	8.25 ±1.92	8.62 ±1.91	8.80 ±1.78	9.12 ±1.84
$w \cdot h \cdot P^8$	7.95 ±3.07	10.88 ±3.16	11.98 ±3.67	13.08 ±3.86	14.08 ±4.71

Acknowledgments

I would like to thank professor Andrzej Skowron for a great support while writing this paper and professor Jerzy Stefanowski for insightful comments, which helped to improve the experimental evaluation. This work was partially supported by the Polish State Committee for Scientific Research grant No. 8T11C02519 and by the Katholischer Akademischer Ausländer-Dienst.

References

[1] Bazan, J. G., Szczuka, M. S., Wróblewski, J.: A New Version of Rough Set Exploration System, *Rough Sets and Current Trends in Computing, 3rd International Conference, RSCTC 2002, Malvern, PA, USA, October 14-16, 2002, Proceedings* (J. J. Alpigini, J. F. Peters, A. Skowron, N. Zhong, Eds.), Springer, 2002.

[2] Birk, A., Coradeschi, S., Tadokoro, S., Eds.: *RoboCup 2001: Robot Soccer World Cup V*, Springer, 2002.

[3] Dash, M., Liu, H.: Feature Selection for Classification, *Intelligent Data Analysis*, **1**(3), 1997.

[4] Freund, Y., Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting, *EuroCOLT '95* (P. M. B. Vitányi, Ed.), Springer, 1995.

[5] Fujikawa, Y., Ho, T. B.: Cluster-Based Algorithms for Dealing with Missing Values, *PAKDD-2002* (M.-S. Chen, P. S. Yu, B. Liu, Eds.), Springer, 2002.

[6] Grzymała-Busse, J. W., Hu, M.: A Comparison of Several Approaches to Missing Attribute Values in Data Mining, *Rough Sets and Current Trends in Computing, Second International Conference, RSCTC 2000 Banff, Canada, October 16-19, 2000, Revised Papers* (W. Ziarko, Y. Y. Yao, Eds.), Springer, 2001.

[7] Grzymała-Busse, J. W., Wang, A. Y.: Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values, *Proceedings of 5th Workshop on Rough Sets and Soft Computing (RSSC'97) at the 3rd Joint Conference on Information Sciences*, Research Triangle Park (NC, USA), 1997.

[8] Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough Sets: A Tutorial, *Rough Fuzzy Hybridization. A New Trend in Decision Making* (S. K. Pal, A. Skowron, Eds.), Springer, Singapore, 1999.

[9] Kryszkiewicz, M.: Properties of Incomplete Information Systems in the Framework of Rough Sets, in: Polkowski and Skowron [18], 422–450.

[10] Latkowski, R.: Application of Data Decomposition to Incomplete Information Systems, *Intelligent Information Systems XI, 2002, Sopot, Poland* (M. A. Kłopotek, S. T. Wierzchoń, Eds.), Physica-Verlag, 2002.

[11] Lim, T.: *Missing covariate values and classification trees*, http://www.recursive-partitioning.com/mv.shtml, Recursive-Partitioning.com, 2000.

[12] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Progams*, Springer, 1996.

[13] Nguyen, H. S., Nguyen, S. H.: From Optimal Hyperplanes to Optimal Decision Trees, *RSFD'96* (S. Tsumoto, S. Kobayashi, T. Yokomori, H. Tanaka, A. Nakamura, Eds.), Tokyo University, 1996.

[14] Nguyen, H. S., Nguyen, S. H.: Discretization Methods in Data Mining, in: Polkowski and Skowron [18], 451–482.

[15] Nguyen, S. H.: *Regularity Analysis and its Application in Data Mining*, Ph.D. Thesis, Warsaw University, Faculty of Mathematics, Computer Science and Mechanics, 1999.

[16] Nguyen, S. H., Skowron, A., Synak, P.: Discovery of data patterns with applications to decomposition and classification problems, in: Polkowski and Skowron [19], 55–97.

[17] Pawlak, Z.: Rough Sets, *International Journal of Computer and Information Sciences*, **11**, 1982, 341–356.

[18] Polkowski, L., Skowron, A., Eds.: *Rough Sets in Knowledge Discovery 1: Methodology and Applications*, Physica-Verlag, 1998.

[19] Polkowski, L., Skowron, A., Eds.: *Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, 1998.

[20] Quinlan, J. R.: Unknown Attribute Values in Induction, *Proceedings of the Sixth International Machine Learning Workshop* (A. M. Segre, Ed.), Morgan Kaufmann, 1989.

[21] Skowron, A.: Synthesis of Adaptive Decision Systems from Experimental Data, *SCAI 1995* (A. Aamodt, J. H. Komorowski, Eds.), IOS Press, 1995.

[22] Skowron, A., Rauszer, C.: The Discernibility Matrices and Functions in Information Systems, *Intelligent Decision Support. Handbook of Applications and Advances in Rough Sets Theory* (R. Słowiński, Ed.), Kluwer, Dordrecht, 1992.

[23] Stefanowski, J.: On rough set based approaches to induction of decision rules, in: Polkowski and Skowron [18], 500–529.

[24] Stefanowski, J., Tsoukiàs, A.: Incomplete Information Tables and Rough Classification, *International Journal of Computational Intelligence*, **17**(3), August 2001, 545–566.

[25] Weiss, S. M., Indurkhya, N.: Lightweight Rule Induction, *Proceedings of the International Conference on Machine Learning ICML'2000*, 2000.