Control and Cybernetics

vol. **39** (2010) No. 2

An improved comparison of three rough set approaches to missing attribute values*

by

Jerzy W. Grzymala-Busse[1], Witold J. Grzymala-Busse[2], Zdzisław S. Hippe[3] and Wojciech Rząsa[4]

[1] Department of Electrical Engineering and Computer Science
University of Kansas, Lawrence, KS 66045, USA and
Institute of Computer Science, Polish Academy of Sciences
01–237 Warsaw, Poland

[2] Touchnet Information Systems, Inc., Lenexa, KS 66219, USA

[3] Department of Expert Systems and Artificial Intelligence,
University of Information Technology and Management,
35–225 Rzeszow, Poland

[4] Department of Computer Science, University of Rzeszow,
35–310 Rzeszow, Poland

Abstract: In a previous paper three types of missing attribute values: lost values, attribute-concept values and "do not care" conditions were compared using six data sets. Since previous experimental results were affected by large variances due to conducting experiments on different versions of a given data set, we conducted new experiments, using the same pattern of missing attribute values for all three types of missing attribute values and for both certain and possible rules. Additionally, in our new experiments, the process of incremental replacing specified values by missing attribute values was terminated when entire rows of the data sets were full of missing attribute values. Finally, we created new, incomplete data sets by replacing the specified values starting from 5% of all attribute values, instead of 10% as in the previous experiments, with an increment of 5% instead of the previous increment of 10%. As a result, it is becoming more clear that the best approach to missing attribute values is based on lost values, with small difference between certain and possible rules, and that the worst approach is based on "do not care" conditions, certain rules. With our improved experimental setup it is also more clear that for a given data set the type of the missing attribute values should be selected individually.

Keywords: incomplete data sets, missing attribute values, approximations for incomplete data, LERS data mining system, MLEM2 algorithm.

*Submitted: December 2008; Accepted: August 2009.

1. Introduction

The main objective of this paper is to compare three different interpretations of missing attribute values, all three based on rough sets.

In a previous paper (Grzymala-Busse and Grzymala-Busse, 2007) results of experiments conducted to compare three types of missing attribute values: *lost values*, *attribute-concept values*, and *"do not care" conditions* were reported.

Lost values are interpreted as currently unavailable, even though originally they were available in the data set. Such values might be incidentally erased. Another possibility is that they were given, but were not recorded. A rough set approach to incomplete data sets in which all attribute values were lost was presented for the first time in Grzymala-Busse and Wang (1997), where two algorithms for rule induction, modified to handle lost attribute values, were introduced.

In attribute-concept values we are assuming that the missing attribute value, where the corresponding case belongs to a concept C, could be any attribute value for all cases from the same concept C. A *concept* (class) is a set of all cases classified (or diagnosed) in the same way. For example, if for a patient the value of attribute *Temperature* is missing, this patient is sick with *flu*, and all remaining patients sick with *flu* have values *high* or *very_high* for *Temperature* then using the interpretation of the missing attribute value as the *attribute-concept value*, we will replace the missing attribute value with *high* and *very_high*. This approach was introduced in Grzymala-Busse (2004c).

For "do not care" conditions a missing attribute value is replaced by all specified values for that attribute. For example, if all possible values of the attribute *Temperature* are: *normal*, *high*, *very_high*, then a missing attribute value will be replaced by all three values: *normal*, *high*, *very_high*. A rough set approach to incomplete data sets in which all attribute values were "do not care" conditions was presented for the first time in Grzymala-Busse (1991), where a method for rule induction was introduced in which each missing attribute value was replaced by all values from the domain of the same attribute.

In general, incomplete decision tables are described by characteristic relations, in a similar way as complete decision tables are described by indiscernibility relations (Grzymala-Busse, 2003, 2004a,b).

In rough set theory (Pawlak, 1982, 1991), one of the basic notions is the idea of lower and upper approximations. For complete decision tables, once the indiscernibility relation is fixed and the concept (a set of cases) is given, the lower and upper approximations are unique.

For incomplete decision tables, for a given characteristic relation and concept, there are three important and different possibilities to define lower and upper approximations, called singleton, subset, and concept approximations (Grzymala-Busse, 2003). Singleton lower and upper approximations were studied in Kryszkiewicz (1995, 1999); Slowinski and Vanderpooten (2000); Stefanowski and Tsoukias (1999, 2001). Note that similar definitions of lower and

Table 1. An incomplete decision table

	Attributes			Decision
Case	Temperature	Headache	Cough	Flu
1	high	?	yes	yes
2	?	yes	*	yes
3	–	no	*	no
4	high	–	yes	yes
5	*	yes	no	no
6	normal	no	?	no

upper approximations, though not for incomplete decision tables, were studied in Lin (1992); Yau (1998); Yao and Lin (1996). Some other rough-set approaches to missing attribute values were presented in Grzymala-Busse (1991); Grzymala-Busse and Hu (2000); Hong, Tseng and Chien (2004), Nakata and Sakai (2005); Wang (2002) as well.

This paper first introduces briefly the methodology used in our research and then presents results of experiments aimed at comparison of three different interpretations of missing attribute values. Finally, we conclude that the best approach to missing attribute values is based on the interpretation of lost values.

A preliminary version of this paper was prepared for the 16-th International Conference on Intelligent Information Systems, Zakopane, Poland, June 16–18, 2008 (Grzymala-Busse et al., 2008).

2. Mining incomplete data

We assume that the input data sets are presented in the form of a *decision table*. An example of a decision table is shown in Table 1. Rows of the decision table represent *cases*, while columns are labeled by *variables*. The set of all cases will be denoted by U. In Table 1, $U = \{1, 2,..., 6\}$. Independent variables are called *attributes* and a dependent variable is called a *decision* and is denoted by d. The set of all attributes will be denoted by A. In Table 1, $A = \{$ *Temperature*, *Headache*, *Cough*$\}$ and $d =$ *Flu*. Any decision table defines a function ρ that maps the direct product of U and A into the set of all values. For example, in Table 1, $\rho(1, Temperature) = high$. A decision table with completely specified function ρ will be called *completely specified*, or, for the sake of simplicity, *complete*. In practice, input data for data mining are frequently affected by missing attribute values. In other words, the corresponding function ρ is incompletely specified (partial). A decision table with an incompletely specified function ρ will be called *incomplete*. Function ρ describing Table 1 is incompletely specified.

An important tool to analyze complete decision tables is a block of the attribute-value pair. Let a be an attribute, i.e., $a \in A$, and let v be a value of a for some case. For complete decision tables if $t = (a, v)$ is an attribute-value pair then a *block* of t, denoted $[t]$, is a set of all cases from U that for attribute a have value v. For incomplete decision tables the definition of a block of an attribute-value pair must be modified in the following way:

- If for an attribute a there exists a case x such that $\rho(x, a) = ?$, i.e., the corresponding value is lost, then the case x should not be included in any blocks $[(a, v)]$ for all values v of attribute a,
- If for an attribute a there exists a case x such that the corresponding value is a "do not care" condition, i.e., $\rho(x, a) = *$, then the case x should be included in blocks $[(a, v)]$ for all specified values v of attribute a.
- If for an attribute a there exists a case x such that the corresponding value is an attribute-concept value, i.e., $\rho(x, a) = -$, then the corresponding case x should be included in blocks $[(a, v)]$ for all specified values $v \in V(x, a)$ of attribute a, where

$$V(x, a) = \{\rho(y, a) \mid \rho(y, a) \textit{ is specified}, y \in U, \ \rho(y, d) = \rho(x, d)\}.$$

For Table 1, $V(3, Temperature) = \{normal\}$ and $V(4, Headache) = \{yes\}$, so the blocks of attribute-value pairs are:

[(Temperature, high)] = {1, 4, 5},
[(Temperature, normal)] = {3, 5, 6},
[(Headache, yes)] = {2, 4, 5},
[(Headache, no)] = {3, 6},
[(Cough, yes)] = {1, 2, 3, 4},
[(Cough, no)] = {2, 3, 5}.

For a case $x \in U$ the *characteristic set* $K_B(x)$ is defined as the intersection of the sets $K(x, a)$, for all $a \in B$, where the set $K(x, a)$ is defined in the following way:

- If $\rho(x, a)$ is specified, then $K(x, a)$ is the block $[(a, \rho(x, a)]$ of attribute a and its value $\rho(x, a)$,
- If $\rho(x, a) = ?$ or $\rho(x, a) = *$ then the set $K(x, a) = U$,
- If $\rho(x, a) = -$, then the corresponding set $K(x, a)$ is equal to the union of all blocks of attribute-value pairs (a, v), where $v \in V(x, a)$ if $V(x, a)$ is nonempty. If $V(x, a)$ is empty, $K(x, a) = U$.

For Table 1 and $B = A$,

$K_A(1) = \{1, 4, 5\} \cap U \cap \{1, 2, 3, 4\} = \{1, 4\}$,
$K_A(2) = U \cap \{2, 4, 5\} \cap U = \{2, 4, 5\}$,
$K_A(3) = \{3, 5, 6\} \cap \{3, 6\} \cap U = \{3, 6\}$,
$K_A(4) = \{1, 4, 5\} \cap \{2, 4, 5\} \cap \{1, 2, 3, 4\} = \{4\}$,
$K_A(5) = U \cap \{2, 4, 5\} \cap \{2, 3, 5\} = \{2, 5\}$,
$K_A(6) = \{3, 5, 6\} \cap \{3, 6\} \cap U = \{3, 6\}$.

Characteristic set $K_B(x)$ may be interpreted as the set of cases that are indistinguishable from x using all attributes from B and using a given interpretation of missing attribute values. Thus, $K_A(x)$ is the set of all cases that cannot be distinguished from x using all attributes. The characteristic relation $R(B)$ is a relation on U defined for $x, y \in U$ as follows

$(x, y) \in R(B)$ *if and only if* $y \in K_B(x)$.

Thus, the relation $R(B)$ may be defined by $(x, y) \in R(B)$ if and only if y is indistinguishable from x by all attributes from B.

For decision tables, in which all missing attribute values are lost, a special characteristic relation was defined in Stefanowski and Tsoukias (1999), see also, e.g., Stefanowski and Tsoukias (2001). For decision tables where all missing attribute values are "do not care" conditions a special characteristic relation was defined in Kryszkiewicz (1995), see also, e.g., Kryszkiewicz (1999). For a completely specified decision table, the characteristic relation $R(B)$ is reduced to the indiscernibility relation (Pawlak, 1982, 1991). For some other approaches to missing attribute values see, e.g., Dardzinska and Ras (2005); Little and Rubin (2002).

3. Definability

For completely specified decision tables, any union of elementary sets of B is called a B-definable set, see, e.g., Pawlak (1991). Definability for completely specified decision tables should be modified to fit into incomplete decision tables. For incomplete decision tables, a union of some intersections of attribute-value pair blocks, where such attributes are members of B and are distinct, will be called B-*locally definable* sets. A union of characteristic sets $K_B(x)$, where $x \in X \subseteq U$, will be called a B-*globally definable* set. Any set X that is B-globally definable is B-locally definable, the converse is not true. For example, the set $\{5\}$ is A-locally definable since $\{5\} = [(Temperature, normal)] \cap [(Headache, yes)]$. However, the set $\{5\}$ is not A-globally definable. On the other hand, the set $\{1\}$ is not even A-locally definable. Obviously, if a set is not B-locally definable then it cannot be expressed by rule sets using attributes from B. This is why it is important to distinguish between B-locally definable sets and those that are not B-locally definable.

4. Lower and upper approximations

For completely specified decision tables lower and upper approximations are defined on the basis of the indiscernibility relation, see Pawlak (1982, 1991).

For incomplete decision tables lower and upper approximations may be defined in a few different ways. In this paper we will discuss three different definitions of lower and upper approximations for incomplete decision tables, following Grzymala-Busse (2003, 2004a,b). Let X be a concept, let B be a subset of the set A of all attributes, and let $R(B)$ be the characteristic relation of the incomplete decision table with characteristic sets $K_B(x)$, where $x \in U$. Our first definition uses a similar idea as in the previous articles on incomplete decision tables, Stefanowski and Tsoukias (1999, 2001); Kryszkiewicz (1995, 1999) i.e., lower and upper approximations are sets of singletons from the universe U satisfying some properties. We will call these approximations *singleton*. A singleton B-lower approximation of X is defined as follows:

$$\underline{B}X = \{x \in U \mid K_B(x) \subseteq X\}.$$

A singleton B-upper approximation of X is

$$\overline{B}X = \{x \in U \mid K_B(x) \cap X \neq \emptyset\}.$$

In our example of the decision table presented in Table 1 let us say that $B = A$. Then the singleton A-lower and A-upper approximations of the two concepts: {1, 2, 3} and {4, 5, 6, 7, 8} are:

$$\underline{A}\{1,2,4\} = \{1,4\},$$
$$\underline{A}\{3,5,6\} = \{3,6\},$$
$$\overline{A}\{1,2,4\} = \{1,2,4,5\},$$
$$\overline{A}\{3,5,6\} = \{2,3,5,6\}.$$

We may easily observe that the set $\{1\} = \underline{A}\{1,2,4\}$ is not A-locally definable since in all blocks of attribute-value pairs cases 1 and 4 are inseparable. Thus, as it was observed in, e.g., Grzymala-Busse (2003, 2004a,b), singleton approximations should not be used, theoretically, for data mining and, in particular, for rule induction.

The second method of defining lower and upper approximations for complete decision tables uses another idea: lower and upper approximations are unions of elementary sets, subsets of U. Therefore, we may define lower and upper approximations for incomplete decision tables by analogy with the second method, using characteristic sets instead of elementary sets. There are two ways to do this. Using the first way, a *subset* B-lower approximation of X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \subseteq X\}.$$

A *subset* B-upper approximation of X is

$$\overline{B}X = \cup\{K_B(x) \mid x \in U, K_B(x) \cap X \neq \emptyset\}.$$

Since any characteristic relation $R(B)$ is reflexive; for any concept X, singleton B-lower and B-upper approximations of X are subsets of the subset B-lower and B-upper approximations of X, respectively , Grzymala-Busse (2004a). For the same decision table, presented in Table 1, the subset A-lower and A-upper approximations are

$$\underline{A}\{1,2,4\} = \{1,4\},$$
$$\underline{A}\{3,5,6\} = \{3,6\},$$
$$\overline{A}\{1,2,4\} = \{1,2,4,5\},$$
$$\overline{A}\{3,5,6\} = \{2,3,4,5,6\}.$$

The second possibility is to modify the subset definition of lower and upper approximation by replacing the universe U from the subset definition by a concept X. A *concept* B-lower approximation of the concept X is defined as follows:

$$\underline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \subseteq X\}.$$

Obviously, the subset B-lower approximation of X is the same set as the concept B-lower approximation of X. A *concept* B-upper approximation of the concept X is defined as follows:

$$\overline{B}X = \cup\{K_B(x) \mid x \in X, K_B(x) \cap X \neq \emptyset\} =$$
$$= \cup\{K_B(x) \mid x \in X\}.$$

The concept upper approximations were defined in Lin (1992) and Slowinski and Vanderpooten (2000), as well. The concept B-upper approximation of X is a subset of the subset B-upper approximation of X, Grzymala-Busse (2004). For the decision table presented in Table 1, the concept A-upper approximations are

$$\overline{A}\{1,2,4\} = \{1,2,4,5\},$$
$$\overline{A}\{3,5,6,7,8\} = \{2,3,5,6\}.$$

Note that for complete decision tables, all three definitions of lower approximations, singleton, subset and concept, coalesce to the same definition. Also, for complete decision tables, all three definitions of upper approximations coalesce to the same definition. This is not true for incomplete decision tables, as our example shows.

5. LERS and LEM2

The data system LERS (Learning from Examples based on Rough Sets), Grzymala-Busse (1992), induces rules from inconsistent data, i.e., data with conflicting cases. Two cases are conflicting when they are characterized by the same values of all attributes, but they belong to different concepts (classes).

Rules induced from the lower approximation of the concept *certainly* describe the concept, hence such rules are called *certain*. On the other hand, rules induced from the upper approximation of the concept describe the concept *possibly*, so these rules are called *possible*.

The LEM2 algorithm, a part of LERS, is most frequently used for rule induction. LEM2 explores the search space of attribute-value pairs. Its input data set is a lower or upper approximation of a concept, so its input data set is always consistent. In general, LEM2 computes a local covering and then converts it into a rule set. We will quote a few definitions to describe the LEM2 algorithm, see. e.g., Chan and Grzymala-Busse (1991); Grzymala-Busse (1992, 2002).

The LEM2 algorithm is based on the idea of an attribute-value pair block. Let X be a nonempty lower or upper approximation of a concept represented by a decision-value pair (d, w). Set X *depends* on a set T of attribute-value pairs $t = (a, v)$ if and only if

$$\emptyset \neq [T] = \bigcap_{t \in T} [t] \subseteq X.$$

Set T is a *minimal complex* of X if and only if X depends on T and no proper subset T' of T exists such that X depends on T'. Let $\mathcal{T}$ be a nonempty collection of nonempty sets of attribute-value pairs. Then $\mathcal{T}$ is a *local covering* of X if and only if the following conditions are satisfied:

- each member T of $\mathcal{T}$ is a minimal complex of X,
- $\bigcup_{t \in \mathcal{T}} [T] = X$, and
- $\mathcal{T}$ is minimal, i.e., $\mathcal{T}$ has the smallest possible number of members.

MLEM2, a modified version of LEM2, processes numerical attributes differently than symbolic attributes. For numerical attributes MLEM2 sorts all values of a numerical attribute. Then it computes cutpoints as averages for any two consecutive values of the sorted list. For each cutpoint q MLEM2 creates two blocks, the first block contains all cases, for which values of the numerical attribute are smaller than q, the second block contains remaining cases, i.e., all cases, for which values of the numerical attribute are larger than q. The search space of MLEM2 is the set of all blocks computed in this way, together with blocks defined by symbolic attributes. Starting from that point, rule induction in MLEM2 is conducted, the same way as in LEM2.

Table 2. Data sets used for experiments

Data set	Number of		
	cases	attributes	concepts
Bankruptcy	66	5	2
Breast cancer - Slovenia	277	9	2
Hepatitis	155	19	2
Image segmentation	210	19	7
Iris	150	4	3
Lymphography	148	18	4
Wine	178	12	3

In our data sets, lost values were denoted by "?", "do not care" conditions by "*", and attribute-concept values by "−". We assumed that for each case at least one attribute value was specified.

For rule induction from incomplete data we used the MLEM2 data mining algorithm, for details see Grzymala-Busse (1992, 2002). Additionally, we used *concept* lower and upper approximations to induce *certain* and *possible* rules.

6. Experiments

In our experiments seven typical data sets were used, see Table 2. In two data sets: *bankruptcy* and *iris* all attributes were numerical. These data sets were processed as numerical (i.e., discretization was done during rule induction by MLEM2). The *image segmentation* data set was converted into symbolic using a discretization method based on agglomerative cluster analysis (this method was described, e.g., in Chmielewski and Grzymala-Busse, 1996).

Since previous experimental results were affected by large variances due to conducting experiments on different versions of a given data set, we conducted new experiments, using the same pattern of missing attribute values for all three types of missing attribute values and for both certain and possible rules.

For every data set a set of templates was created. Templates were formed by replacing incrementally (with 5% increment) existing specified attribute values by *lost values*. Thus, we started each series of experiments with no *lost values*, then we added 5% of *lost values*, then we added additional 5% of *lost values*, etc., until at least one entire row of the data sets was full of *lost values*. Then, three attempts were made to change configuration of new *lost values* and either a new data set with extra 5% of *lost values* was created or the process was terminated. Additionally, the same formed templates were edited for further experiments by replacing question marks, representing *lost values* by "−" representing *attribute-concept values* and, separately, by "*", representing *"do not care" conditions*.

6.1. A single ten-fold cross validation scheme

For each data set with some percentage of missing attribute values of a given type, experiments were conducted separately for certain and possible rule sets, using *concept* lower and upper approximations, respectively. Ten-fold cross validation was used to compute error rate. Rule sets were induced by the MLEM2 option of the LERS data mining system. Results of our experiments are presented in Figs. 1–7.

In four data sets: *image segmentation*, *iris*, *lymphography* and *wine*, strategies based on *lost values* were the best, irrespective of whether certain or possible rule sets were used. The strategy based on *do not care conditions* and certain rule sets was the worst strategy, while *attribute-concept value* combined with the certain rule sets was the next bad strategy.

Remaining data sets show different patterns. In the *bankruptcy* data set, for *lost values* and *attribute-concept values*, there is no difference in performance between certain and possible rule sets. Both strategies, based on *attribute-concept values* and *"do not care" conditions*, show gradual increase of the error rate with increasing percentage of missing attribute values, while a strategy based on *"do not care" conditions* seems to be the worst. The strategy based on *lost values*, starting from 20% of missing attribute values, surprisingly, show decrease in the error rate with increasing percentage of missing attribute values.

In the *breast cancer - Slovenia* data set there is no preference among all six strategies, except that starting from 35% of missing attribute values, the strategy based on *"do not care" conditions* and certain rule sets is the worst strategy.

The *hepatitis* data set shows gradual increase of the error rate with increasing percentage of missing attribute values up to 45%, with no clear pattern of any interpretation of missing attribute values being better or worse. Starting from 50% of the percentage of missing attribute values, the strategy based on *attribute-concept values* combined with the certain rule sets seems to be the worst strategy, while the strategy based on *attribute-concept values* combined with the possible rule sets seems to be the best strategy.

6.2. Multiple ten-fold cross validation

Additional experiments were conducted on some data sets by repeating the 10-fold cross validation scheme for 30 times, changing the ordering and partitioning every time. In these experiments all six approaches to missing attribute values, i.e., interpreting missing attribute values as *lost*, *attribute-concept values*, and *"do not care" conditions*, combined with inducing two kind of rules: certain and possible, were applied. Since such experiments are time-consuming, only three data sets were selected: *bankruptcy* (an example of a data set with all numerical attributes), *breast cancer* (with all symbolic attributes), and *lymphography* (a

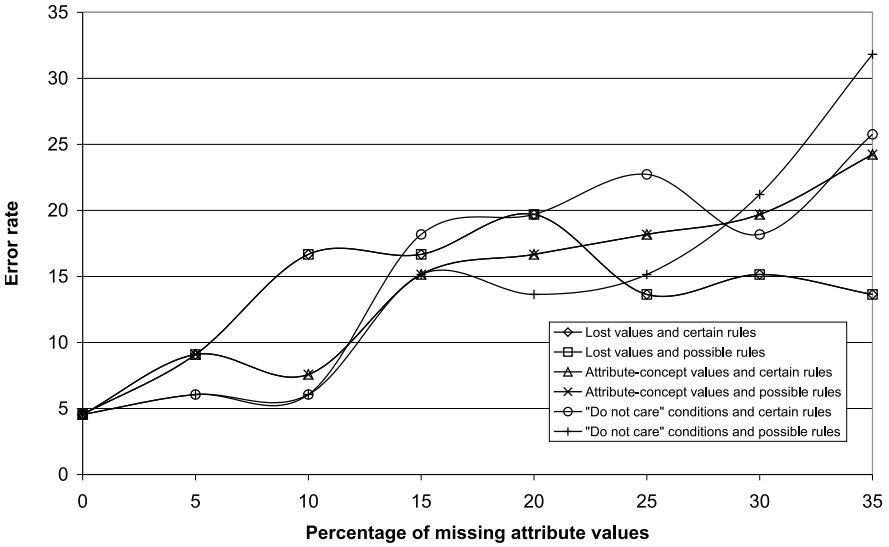

Figure 1. Bankruptcy data set

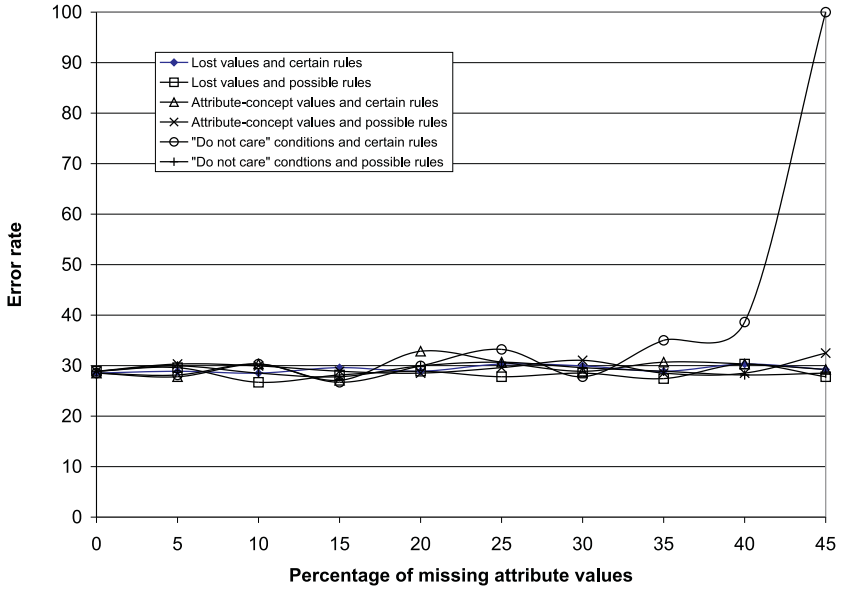

Figure 2. Breast cancer - Slovenia data set

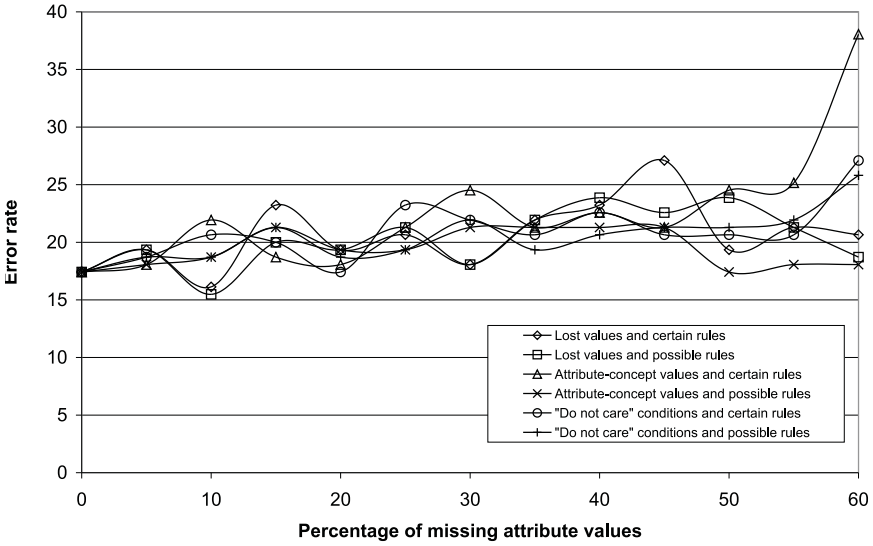

Figure 3. Hepatitis data set

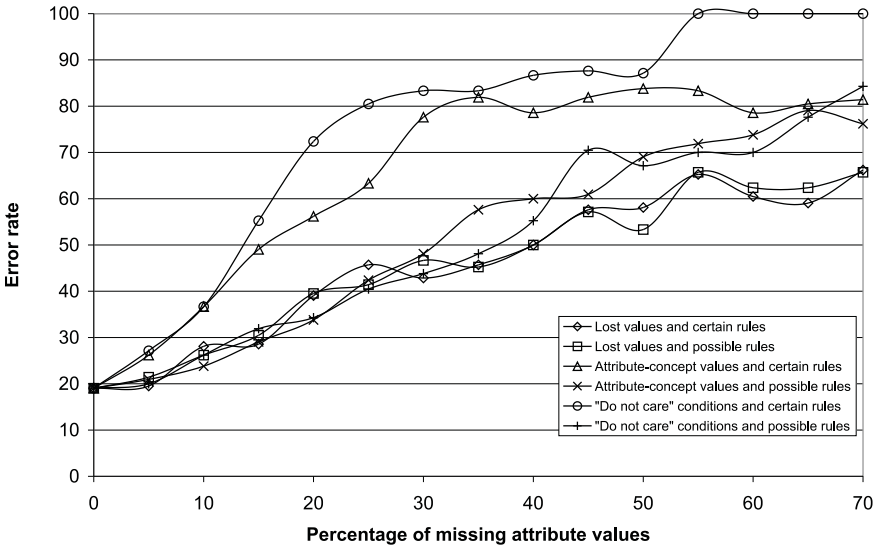

Figure 4. Image segmentation data set

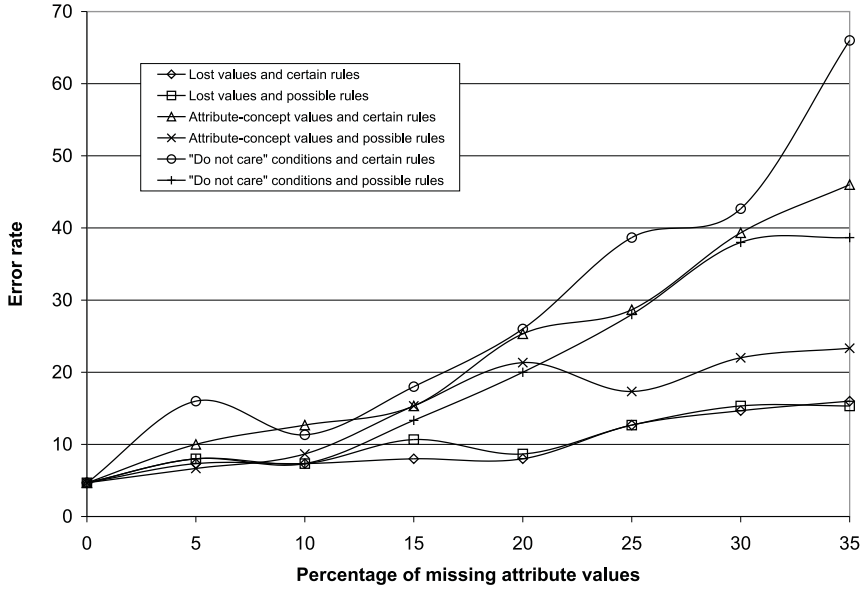

Figure 5. Iris data set

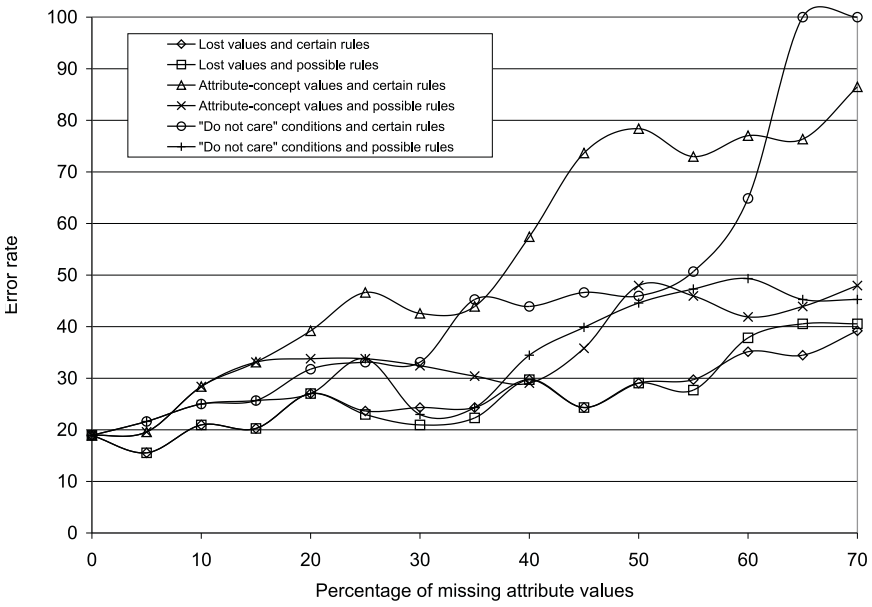

Figure 6. Lymphography data set

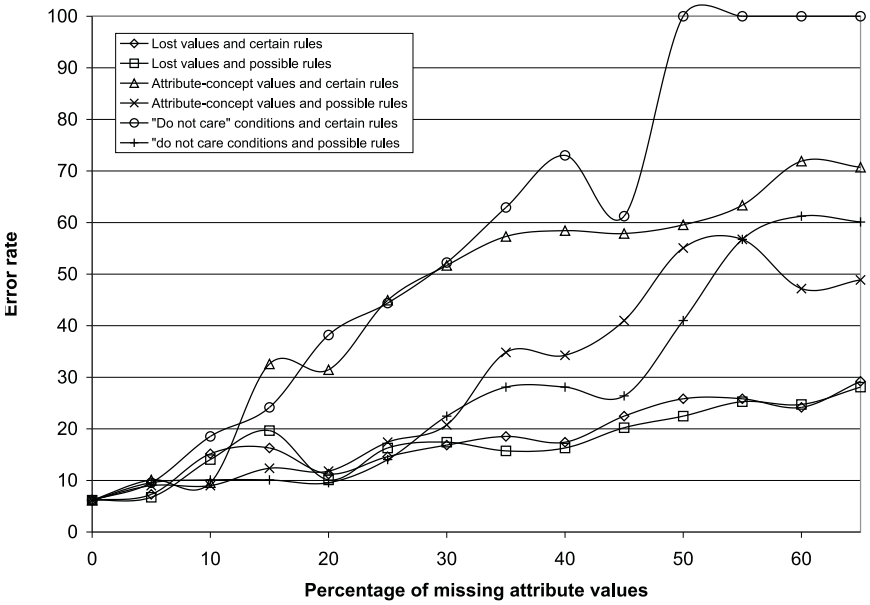

Figure 7. Wine data set

data set with relatively many attributes and concepts). Additionally, all three data sets had 35% of missing attribute values (35% was selected since for the *bankruptcy* data set it is the maximum percentage of missing attribute values for which every case has at least one specified attribute value). Results of our experiments are presented in Table 3.

We compared all 15 pairs of the six methods listed in Table 3 using the standard statistical test (two-tailed) on the difference between two means, based on the following well-known formula

$$Z = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2 + s_2^2}{30}}},$$

where $\overline{X_1}$ and $\overline{X_2}$ are the means of 30 ten-fold cross validation experiments and s_1 and s_2 are the corresponding sample standard deviations.

In general, for all three data sets, there is no significant difference (5% significance level) between certain and possible rules for *lost* values. Moreover, the *lost* interpretation of missing attribute values is the best approach for all three data sets.

For the *bankruptcy* data set, the worst method is based on *"do not care" conditions* and the *attribute-concept values* is worse that *lost values* and better than *"do not care" conditions*. For all three types of missing attribute values

Table 3. Additional experiments—error rates

Method	Bankruptcy		Breast cancer		Lymphography	
	error rate	standard deviation	error rate	standard deviation	error rate	standard deviation
(?, C)	18.13	2.85	28.11	0.85	25.11	3.25
(?, P)	17.32	3.22	28.38	1.14	25.65	2.79
(−, C)	23.38	3.73	30.10	0.71	51.71	3.59
(−, P)	25.05	4.03	29.13	0.52	28.85	3.02
(*, C)	29.44	5.47	34.24	2.50	42.16	3.45
(*, P)	27.58	4.40	28.81	0.75	27.32	2.75

there is no difference between certain and possible rules, all observations with 5% significance level. The last observation is most likely due to the fact that *bankruptcy* is a numerical data set.

For the *breast cancer* data set there is the following order of performance for the different methods: the best one are *lost values*, no significant difference between certain and possible rules, then *attribute-concept values*, certain rules, then *attribute-concept values*, possible rules and *"do not care" conditions*, possible rules, no significant difference between the two, and the worst method is *"do not care" conditions* and certain rules.

For the *lymphography* data set, for only two methods there is no significant difference: *lost values*, no matter whether certain or possible rules. These two methods are also the best. Then there is the following order of performance, from better to worse: *"do not care" conditions*, possible rules, then *attribute-concept values*, possible rules, then *"do not care" conditions*, certain rules and, finally, *attribute-concept values*, certain rules. The only surprise is poor performance of the *attribute-concept value* approach. It may be explained by the fact that the *lymphography* data set has many concepts, many attributes, and a few cases. All of these facts contribute to poor performance of *attribute-concept value* interpretation of missing attribute values since with few cases and many concepts it is difficult to guess correctly the proper attribute value. Furthermore, during classification of a testing case against a rule set, *attribute-concept values* are treated in the same way as *"do not care" conditions*.

7. Conclusions

Overall, it seems that the interpretation of missing attribute values as *lost values* is the best approach among our three types of missing attribute value interpretations, and that the worst approach is based on *"do not care" conditions* and certain rules. This was caused by the fact that lower approximations of con-

cepts, with large number of missing attribute values, were empty. Our additional experiments (see Table 3) support this idea as well. With our improved experimental setup it is also clearer that for a given data set the type of the missing attribute value should be selected individually.

References

CHAN, C.C. and GRZYMALA-BUSSE, J.W. (1991) On the attribute redundancy and the learning programs ID3, PRISM, and LEM2. Technical report. Department of Computer Science, University of Kansas.

CHMIELEWSKI, M.R. and GRZYMALA-BUSSE, J.W. (1996) Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning* **15**, 319–331.

DARDZINSKA, A. and RAS, Z.W. (2005) CHASE-2: Rule based chase algorithm for information systems of type lambda. In: *Proceedings of the Second International Workshop on Active Mining (AM'2003)*, 258–270.

GRZYMALA-BUSSE, J.W. (1992) LERS–A system for learning from examples based on rough sets. In: R. Slowinski, ed., *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory.* Kluwer Academic Publishers, Dordrecht, Boston, London, 3–18.

GRZYMALA-BUSSE, J.W. (2002) MLEM2: A new algorithm for rule induction from imperfect data. In: *Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU 2002)*, 243–250.

GRZYMALA-BUSSE, J.W. (2003) Rough set strategies to data with missing attribute values. In: *Workshop Notes, Foundations and New Directions of Data Mining, in conjunction with the 3-rd International Conference on Data Mining*, 56–63.

GRZYMALA-BUSSE, J.W. (2004a) Characteristic relations for incomplete data: A generalization of the indiscernibility relation. In: *Proceedings of the Fourth International Conference on Rough Sets and Current Trends in Computing*, 244–253.

GRZYMALA-BUSSE, J.W. (2004b) Data with missing attribute values: Generalization of indiscernibility relation and rule induction. *Transactions on Rough Sets*, **1**, 78–95.

GRZYMALA-BUSSE, J.W. (2004c) Three approaches to missing attribute values —A rough set perspective. In: *Proceedings of the Workshop on Foundation of Data Mining, associated with the Fourth IEEE International Conference on Data Mining*, 55–62.

GRZYMALA-BUSSE, J.W. and GRZYMALA-BUSSE, W.J. (2007) An experimental comparison of three rough set approaches to missing attribute values. In: J.F. Peters and A. Skowron, eds. Springer-Verlag, Berlin, Heidelberg, 31–50.

Grzymala-Busse, J.W., Grzymala-Busse, W.J., Hippe, Z.S. and Rząsa, W. (2008) An improved comparison of three rough set approaches to missing attribute values. In: *Proceeedings of the 16-th International Conference on Intelligent Information Systems*, 141–150.

Grzymala-Busse, J.W. and Hu, M. (2000) A comparison of several approaches to missing attribute values in data mining. In: *Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing*, 340–347.

Grzymala-Busse, J.W. and Wang, A.Y. (1997) Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In: *Proceedings of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97) at the Third Joint Conference on Information Sciences (JCIS'97)*, 69–72.

Grzymala-Busse, J.W. (1991) On the unknown attribute values in learning from examples. In: *Proceedings of the ISMIS-91, 6th International Symposium on Methodologies for Intelligent Systems*, 368–377.

Hong, T.P., Tseng, L.H. and Chien, B.C. (2004) Learning coverage rules from incomplete data based on rough sets. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 3226–3231.

Kryszkiewicz, M. (1995) Rough set approach to incomplete information systems. In: *Proceedings of the Second Annual Joint Conference on Information Sciences*, 194–197.

Kryszkiewicz, M. (1999) Rules in incomplete information systems. *Information Sciences* **113**, 271–292.

Lin, T.Y. (1992) Topological and fuzzy rough sets. In: R. Slowinski, ed. *Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory.* Kluwer Academic Publishers, Dordrecht, Boston, London, 287–304.

Little, R.J.A. and Rubin, D.B. (2002) *Statistical Analysis with Missing Data. Second Edition.* John Wiley & Sons, Hoboken, N.J.

Nakata, M. and Sakai, H. (2005) Rough sets handling missing values probabilistically interpreted. In: *Proceedings of the 10-th International Conference RSFDGrC'2005 on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Springer-Verlag, Berlin, Heidelberg, 325–334.

Pawlak, Z. (1982) Rough sets. *International Journal of Computer and Information Sciences* **11**, 341–356.

Pawlak, Z. (1991) *Rough Sets. Theoretical Aspects of Reasoning about Data.* Kluwer Academic Publishers, Dordrecht, Boston, London.

Slowinski, R. and Vanderpooten, D. (2000) A generalized definition of rough approximations based on similarity. *IEEE Transactions on Knowledge and Data Engineering* **12**, 331–336.

Stefanowski, J. and Tsoukias, A. (1999) On the extension of rough sets under incomplete information. In: *Proceedings of the RSFDGrC'1999, 7th International Workshop on New Directions in Rough Sets, Data Mining,*

and Granular-Soft Computing, 73–81.

STEFANOWSKI, J. and TSOUKIAS, A. (2001) Incomplete information tables and rough classification. *Computational Intelligence* **17**, 545–566.

WANG, G. (2002) Extension of rough set under incomplete information systems. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*, 1098–1103.

YAO, Y.Y. (1998) Relational interpretations of neighborhood operators and rough set approximation operators. *Information Sciences* **111**, 239–259.

YAO, Y.Y. and LIN, T.Y. (1996) Generalization of rough sets using modal logics. *Intelligent Automation and Soft Computing* **2**, 103–119.