Chapter 3

HANDLING MISSING ATTRIBUTE VALUES

Jerzy W. Grzymala-Busse
University of Kansas

Witold J. Grzymala-Busse
FilterLogix

Abstract In this chapter methods of handling missing attribute values in Data Mining are described. These methods are categorized into sequential and parallel. In sequential methods, missing attribute values are replaced by known values first, as a preprocessing, then the knowledge is acquired for a data set with all known attribute values. In parallel methods, there is no preprocessing, i.e., knowledge is acquired directly from the original data sets. In this chapter the main emphasis is put on rule induction. Methods of handling attribute values for decision tree generation are only briefly summarized.

Keywords: Missing attribute values, lost values, do not care conditions, incomplete data, imputation, decision tables.

1. Introduction

We assume that input data for Data Mining are presented in a form of a *decision table* (or *data set*) in which *cases* (or *records*) are described by *attributes* (independent variables) and a *decision* (dependent variable). A very simple example of such a table is presented in Table 3.1, with the attributes *Temperature*, *Headache*, and *Nausea* and with the decision *Flu*. However, many real-life data sets are incomplete, i.e., some attribute values are missing. In Table 3.1 missing attribute values are denoted by "?"s.

The set of all cases with the same decision value is called a *concept*. For Table 3.1, case set $\{1, 2, 4, 8\}$ is a concept of all cases such that the value of *Flu* is *yes*.

Table 3.1. An Example of a Data Set with Missing Attribute Values.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	?	no	yes
2	very_high	yes	yes	yes
3	?	no	no	no
4	high	yes	yes	yes
5	high	?	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	?	yes	?	yes

There is variety of reasons why data sets are affected by missing attribute values. Some attribute values are not recorded because they are irrelevant. For example, a doctor was able to diagnose a patient without some medical tests, or a home owner was asked to evaluate the quality of air conditioning while the home was not equipped with an air conditioner. Such missing attribute values will be called *"do not care" conditions.*

Another reason for missing attribute values is that the attribute value was not placed into the table because it was forgotten or it was placed into the table but later on was mistakenly erased. Sometimes a respondent refuse to answer a question. Such a value, that matters but that is missing, will be called *lost.*

The problem of missing attribute values is as important for data mining as it is for statistical reasoning. In both disciplines there are methods to deal with missing attribute values. Some theoretical properties of data sets with missing attribute values were studied in (Imielinski and Lipski, 1984; Lipski, 1979; Lipski, 1981).

In general, methods to handle missing attribute values belong either to *sequential methods* (called also *preprocessing methods*) or to *parallel methods* (methods in which missing attribute values are taken into account during the main process of acquiring knowledge).

Sequential methods include techniques based on deleting cases with missing attribute values, replacing a missing attribute value by the most common value of that attribute, assigning all possible values to the missing attribute value, replacing a missing attribute value by the mean for numerical attributes, assigning to a missing attribute value the corresponding value taken from the closest fit case, or replacing a missing attribute value by a new vale, computed from a new data set, considering the original attribute as a decision.

The second group of methods to handle missing attribute values, in which missing attribute values are taken into account during the main process of acquiring knowledge is represented, for example, by a modification of the

LEM2 (Learning from Examples Module, version 2) rule induction algorithm in which rules are induced form the original data set, with missing attribute values considered to be ”do not care” conditions or lost values. C4.5 (Quinlan, 1993) approach to missing attribute values is another example of a method from this group. C4.5 induces a decision tree during tree generation, splitting cases with missing attribute values into fractions and adding these fractions to new case subsets. A method of *surrogate splits* to handle missing attribute values was introduced in CART (Breiman *et al.*, 1984), yet another system to induce decision trees. Other methods of handling missing attribute values while generating decision trees were presented in (Brazdil and Bruha, 1992) and (Bruha, 2004)

In statistics, *pairwise deletion* (Allison, 2002) (Little and Rubin, 2002) is used to evaluate statistical parameters from available information.

In this chapter we assume that the main process is rule induction. Additionally for the rest of the chapter we will assume that all decision values are known, i.e., specified. Also, we will assume that for each case at least one attribute value is known.

2. Sequential Methods

In sequential methods to handle missing attribute values original incomplete data sets, with missing attribute values, are converted into complete data sets and then the main process, e.g., rule induction, is conducted.

2.1 Deleting Cases with Missing Attribute Values

This method is based on ignoring cases with missing attribute values. It is also called *listwise deletion* (or *casewise deletion*, or *complete case analysis*) in statistics. All cases with missing attribute values are deleted from the data set. For the example presented in Table 3.1, a new table, presented in Table 3.2, is created as a result of this method.

Table 3.2. Dataset with Deleted Cases with Missing Attribute Values.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	very_high	yes	yes	yes
2	high	yes	yes	yes
3	normal	yes	no	no
4	normal	no	yes	no

Obviously, a lot of information is missing in Table 3.2. However, there are some reasons (Allison, 2002), (Little and Rubin, 2002) to consider it a good method.

2.2 The Most Common Value of an Attribute

In this method, one of the simplest methods to handle missing attribute values, such values are replaced by the most common value of the attribute. In different words, a missing attribute value is replaced by the most probable known attribute value, where such probabilities are represented by relative frequencies of corresponding attribute values. This method of handling missing attribute values is implemented, e.g., in CN2 (Clark, 1989). In our example from Table 3.1, a result of using this method is presented in Table 3.3.

Table 3.3. Dataset with Missing Attribute Values replaced by the Most Common Values.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	high	no	no	no
4	high	yes	yes	yes
5	high	yes	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	high	yes	yes	yes

For case 1, the value of *Headache* in Table 3.3 is *yes* since in Table 3.1 the attribute *Headache* has four values *yes* and two values *no*. Similarly, for case 3, the value of *Temperature* in Table 3.3 is *high* since the attribute *Temperature* has the value *very_high* once, *normal* twice, and *high* three times.

2.3 The Most Common Value of an Attribute Restricted to a Concept

A modification of the method of replacing missing attribute values by the most common value is a method in which the most common value of the attribute restricted to the concept is used instead of the most common value for all cases. Such a concept is the same concept that contains the case with missing attribute value.

Let us say that attribute a has missing attribute value for case x from concept C and that the value of a for x is missing. This missing attribute value is exchanged by the known attribute value for which the conditional probability $P(known\ value\ of\ a\ for\ case\ x\ |C)$ is the largest. This method was implemented, e.g., in ASSISTANT (Kononenko *et al.*, 1984). In our example from Table 3.1, a result of using this method is presented in Table 3.4.

Table 3.4. Dataset with Missing Attribute Values Replaced by the Most Common Value of the Attribute Restricted to a Concept

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3	normal	no	no	no
4	high	yes	yes	yes
5	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	high	yes	yes	yes

For example, in Table 3.1, case 1 belongs to the concept {1, 2, 4, 8}, all known values of *Headache*, restricted to {1, 2, 4, 8}, are *yes*, so the missing attribute value is replaced by *yes*. On the other hand, in Table 3.1, case 3 belongs to the concept {3, 5, 6, 7}, and the value of *Temperature* is missing. The known values of *Temperature*, restricted to {3, 5, 6, 7} are: *high* (once) and *normal* (twice), so the missing attribute value is exchanged by *normal.*

2.4 Assigning All Possible Attribute Values to a Missing Attribute Value

This approach to missing attribute values was presented for the first time in (Grzymala-Busse, 1991) and implemented in LERS. Every case with missing attribute values is replaced by the set of cases in which every missing attribute value is replaced by all possible known values. In the example from Table 3.1, a result of using this method is presented in Table 3.5.

In the example of Table 3.1, the first case from Table 3.1, with the missing attribute value for attribute *Headache*, is replaced by two cases, 1^i and 1^{ii}, where case 1^i has value *yes* for attribute *Headache*, and case 1^{ii} has values *no* for the same attribute, since attribute *Headache* has two possible known values, *yes* and *no*. Case 3 from Table 3.1, with the missing attribute value for the attribute *Temperature*, is replaced by three cases, 3^i, 3^{ii}, and 3^{iii}, with values *high*, *very_high*, and *normal*, since the attribute *Temperature* has three possible known values, *high*, *very_high*, and *normal*, respectively. Note that due to this method the new table, such as Table 3.5, may be inconsistent. In Table 3.5, case 1^{ii} conflicts with case 3^i, case 4 conflicts with case 5^i, etc. However, rule sets may be induced from inconsistent data sets using standard rough-

Table 3.5. Dataset in Which All Possible Values are Assigned to Missing Attribute Values.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1^i	high	yes	no	yes
1^{ii}	high	no	no	yes
2	very_high	yes	yes	yes
3^i	high	no	no	no
3^{ii}	very_high	no	no	no
3^{iii}	normal	no	no	no
4	high	yes	yes	yes
5^i	high	yes	yes	no
5^{ii}	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8^i	high	yes	yes	yes
8^{ii}	high	yes	no	yes
8^{iii}	very_high	yes	yes	yes
8^{iv}	very_high	yes	no	yes
8^v	normal	yes	yes	yes
8^{vi}	normal	yes	no	yes

set techniques, see, e.g., (Grzymala-Busse, 1988), (Grzymala-Busse, 1991), (Grzymala-Busse, 1992), (Grzymala-Busse, 1997), (Grzymala-Busse, 2002), (Polkowski and Skowron, 1998).

2.5 Assigning All Possible Attribute Values Restricted to a Concept

This method was described, e.g., in (Grzymala-Busse and Hu, 2000). Here, every case with missing attribute values is replaced by the set of cases in which every attribute a with the missing attribute value has its every possible known value restricted to the concept to which the case belongs. In the example from Table 3.1, a result of using this method is presented in Table 3.6.

In the example of Table 3.1, the first case from Table 3.1, with the missing attribute value for attribute *Headache*, is replaced by one with value *yes* for attribute *Headache*, since attribute *Headache*, restricted to the concept $\{1, 2, 4, 8\}$ has one possible known value, *yes*. Case 3 from Table 3.1, with the missing attribute value for the attribute *Temperature*, is replaced by two cases, 3^i and 3^{ii}, with values *high* and *very_high*, since the attribute *Temperature*, restricted to the concept $\{3, 5, 6, 7\}$ has two possible known values, *normal* and *high*, respectively. Again, due to this method the new table, such as Table 3.6, may be inconsistent. In Table 3.6, case 4 conflicts with case 5^i, etc.

Table 3.6. Dataset in which All Possible Values, Restricted to the Concept, are Assigned to Missing Attribute Values.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	yes	no	yes
2	very_high	yes	yes	yes
3^{i}	normal	no	no	no
3^{ii}	high	no	no	no
4	high	yes	yes	yes
5^{i}	high	yes	yes	no
5^{ii}	high	no	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8^{i}	high	yes	yes	yes
8^{ii}	high	yes	no	yes
8^{iii}	very_high	yes	yes	yes
8^{iv}	very_high	yes	no	yes

2.6 Replacing Missing Attribute Values by the Attribute Mean

This method is used for data sets with numerical attributes. An example of such a data set is presented in Table 3.7.

Table 3.7. An Example of a Dataset with a Numerical Attribute.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	?	no	yes
2	102.6	yes	yes	yes
3	?	no	no	no
4	99.6	yes	yes	yes
5	99.8	?	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	?	yes	?	yes

In this method, every missing attribute value for a numerical attribute is replaced by the arithmetic mean of known attribute values. In Table 3.7, the mean of known attribute values for Temperature is 99.2, hence all missing attribute values for *Temperature* should be replaced by 99.2. The table with missing attribute values replaced by the mean is presented in Table 3.8. For symbolic attributes *Headache* and *Nausea*, missing attribute values were replaced using the most common value of the attribute.

Table 3.8. Data set in which missing attribute values are replaced by the attribute mean and the most common value

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	yes	no	yes
2	102.6	yes	yes	yes
3	99.2	no	no	no
4	99.6	yes	yes	yes
5	99.8	yes	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	99.2	yes	yes	yes

2.7 Replacing Missing Attribute Values by the Attribute Mean Restricted to a Concept

Similarly as the previous method, this method is restricted to numerical attributes. A missing attribute value of a numerical attribute is replaced by the arithmetic mean of all known values of the attribute restricted to the concept. For example from Table 3.7, case 3 has missing attribute value for *Temperature*. Case 3 belong to the concept $\{3, 5, 6, 7\}$. The arithmetic mean of known values of *Temperature* restricted to the concept, i.e., 99.8, 96.4, and 96.6 is 97.6, so the missing attribute value is replaced by 100.8. On the other hand, case 8 belongs to the concept $\{1, 2, 4, 8\}$, the arithmetic mean of 100.2, 102.6, and 99.6 is 100.8, so the missing attribute value for case 8 should be replaced by 100.8. The table with missing attribute values replaced by the mean restricted to the concept is presented in Table 3.9. For symbolic attributes *Headache* and *Nausea*, missing attribute values were replaced using the most common value of the attribute restricted to the concept.

2.8 Global Closest Fit

The global closes fit method (Grzymala-Busse *et al.*, 2002) is based on replacing a missing attribute value by the known value in another case that resembles as much as possible the case with the missing attribute value. In searching for the closest fit case we compare two vectors of attribute values, one vector corresponds to the case with a missing attribute value, the other vector is a candidate for the closest fit. The search is conducted for all cases, hence the name global closest fit. For each case a distance is computed, the case for which the

Table 3.9. Data set in which missing attribute values are replaced by the attribute mean and the most common value, both restricted to the concept

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	yes	no	yes
2	102.6	yes	yes	yes
3	97.6	no	no	no
4	99.6	yes	yes	yes
5	99.8	no	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	100.8	yes	yes	yes

distance is the smallest is the closest fitting case that is used to determine the missing attribute value. Let x and y be two cases. The distance between cases x and y is computed as follows

$$distance(x, y) = \sum_{i=1}^{n} distance(x_i, y_i),$$

where

$$distance(x_i, y_i) = \begin{cases} 0 & \text{if } x_i = y_i \text{ ,} \\ 1 & \text{if } x \text{ and } y \text{ are symbolic and } x_i \neq y_i, \\ & \text{or } x_i = ? \text{ or } y_i = ?, \\ \frac{|x_i - y_i|}{r} & \text{if } x_i \text{ and } y_i \text{ are numbers and } x_i \neq y_i \text{ ,} \end{cases}$$

where r is the difference between the maximum and minimum of the known values of the numerical attribute with a missing value. If there is a tie for two cases with the same distance, a kind of heuristics is necessary, for example, select the first case. In general, using the global closest fit method may result in data sets in which some missing attribute values are not replaced by known values. Additional iterations of using this method may reduce the number of missing attribute values, but may not end up with all missing attribute values being replaced by known attribute values.

Table 3.10. Distance (1, x)

d(1, 2)	d(1, 3)	d(1, 4)	d(1, 5)	d(1, 6)	d(1, 7)	d (1, 8)
2.39	2.0	2.10	1.06	1.61	1.58	3.00

For the data set in Table 3.7, distances between case 1 and all remaining cases are presented in Table 3.10. For example, the distance $d(1, 2) =$

$\frac{|100.2-102.6|}{|102.6-96.4|} + 1 + 1 = 2.39$. For case 1, the missing attribute value (for attribute *Headache*) should be the value of *Headache* for case 5, i.e., *yes*, since for this case the distance is the smallest. However, the value of *Headache* for case 5 is still missing. The table with missing attribute values replaced by the value computed on the basis of the global closest fit is presented in Table 3.11. Some missing attribute values are still present in this table. In such cases it is recommended to use another method of handling missing attribute value to replace all missing attribute values by known attribute values.

Table 3.11. Data Set Processed by the Global Closest Fit Method.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	?	no	yes
2	102.6	yes	yes	yes
3	96.4	no	no	no
4	99.6	yes	yes	yes
5	99.8	?	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	96.4	yes	no	yes

2.9 Concept Closest Fit

This method is similar to the global closest fit method. The difference is that the original data set, containing missing attribute values, is first split into smaller data sets, each smaller data set corresponds to a concept from the original data set. More precisely, every smaller data set is constructed from one of the original concepts, by restricting cases to the concept. For the data set from Table 3.7, two smaller data sets are created, presented in Tables 3.12 and 3.13.

Table 3.12. Dataset Restricted to the Concept $\{1, 2, 4, 8\}$.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	?	no	yes
2	102.6	yes	yes	yes
4	99.6	yes	yes	yes
8	?	yes	?	yes

Following the data set split, the same global closest fit method is applied to both tables separately. Eventually, both tables, processed by the global fit

Table 3.13. Dataset Restricted to the Concept {3, 5, 6, 7}.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
3	?	no	no	no
5	99.8	?	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no

method, are merged into the same table. In our example from Table 3.7, the final, merged table is presented in Table 3.14.

Table 3.14. Dataset Processed by the Concept Closest Fit Method.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	100.2	yes	no	yes
2	102.6	yes	yes	yes
3	96.4	no	no	no
4	99.6	yes	yes	yes
5	99.8	no	yes	no
6	96.4	yes	no	no
7	96.6	no	yes	no
8	100.2	yes	no	yes

2.10 Other Methods

There is a number of other methods to handle missing attribute values. One of them is *event-covering method* (Chiu and Wong, 1986), (Wong and Chiu, 1987), based on an interdependency between known and missing attribute values. The interdependency is computed from contingency tables. The outcome of this method is not necessarily a complete data set (with all attribute values known), just like in the case of closest fit methods.

Another method of handling missing attribute values, called D^3RJ was discussed in (Latkowski, 2003; Latkowski and Mikolajczyk, 2004). In this method a data set is decomposed into complete data subsets, rule sets are induced from such data subsets, and finally these rule sets are merged.

Yet another method of handling missing attribute values was refereed to as Shapiro's method in (Quinlan, 1989), where for each attribute with missing attribute values a new data set is created, such attributes take place of the decision and vice versa, the decision becomes one of the attributes. From such a table missing attribute values are learned using either a rule set or decision tree techniques. This method, identified as a *chase* algorithm, was also discussed in (Dardzinska and Ras, 2003A; Dardzinska and Ras, 2003B).

Learning missing attribute values from summary constraints was reported in (Wu and Barbara, 2002; Wu and Barbara, 2002). Yet another approach to handling missing attribute values was presented in (Greco *et al.*, 2000).

There is a number of statistical methods of handling missing attribute values, usually known under the name of *imputation* (Allison, 2002; Little and Rubin, 2002; Schikuta, 1996), such as maximum likelihood and the EM algorithm. Recently *multiple imputation* gained popularity. It is a Monte Carlo method of handling missing attribute values in which missing attribute values are replaced by many plausible values, then many complete data sets are analyzed and the results are combined.

3. Parallel Methods

In this section we will concentrate on handling missing attribute values in parallel with rule induction. We will distinguish two types of missing attribute values: *lost* and *do not care* conditions (for respective interpretation, see Introduction). First we will introduce some useful ideas, such as blocks of attribute-value pairs, characteristic sets, characteristic relations, lower and upper approximations. Later we will explain how to induce rules using the same blocks of attribute-value pairs that were used to compute lower and upper approximations. Input data sets are not preprocessed the same way as in sequential methods, instead, the rule learning algorithm is modified to learn rules directly from the original, incomplete data sets.

3.1 Blocks of Attribute-Value Pairs and Characteristic Sets

In this subsection we will quote some basic ideas of the rough set theory. Any decision table defines a function ρ that maps the direct product of the set U of all cases and the set A of all attributes into the set of all values. For example, in Table 3.1, $\rho(1, Temperature) = high$. In this section we will assume that all missing attribute values are denoted either by "?" or by "*", lost values will be denoted by "?", "do not care" conditions will be denoted by "*". Thus, we assume that all missing attribute values from Table 3.1 are lost. On the other hand, all attribute values from Table 3.15 are do not care conditions.

Let (a, v) be an attribute-value pair. For complete decision tables, a block of (a, v), denoted by $[(a, v)]$, is the set of all cases x for which $\rho(x, a) = v$. For incomplete decision tables the definition of a block of an attribute-value pair is modified. 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 is not included in any block

Table 3.15. An Example of a Dataset with Do Not Care Conditions.

Case	Attributes			Decision
	Temperature	Headache	Nausea	Flu
1	high	*	no	yes
2	very_high	yes	yes	yes
3	*	no	no	no
4	high	yes	yes	yes
5	high	*	yes	no
6	normal	yes	no	no
7	normal	no	yes	no
8	*	yes	*	yes

$[(a, v)]$ for every value 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 corresponding case x should be included in blocks $[(a, v)]$ for all known values v of attribute a. This modification of the attribute-value pair block definition is consistent with the interpretation of missing attribute values, lost and "do not care" conditions. Thus, for Table 3.1

[(Temperature, high)] = $\{1, 4, 5\}$,
[(Temperature, very_high)] = $\{2\}$,
[(Temperature, normal)] = $\{6, 7\}$,
[(Headache, yes)] = $\{2, 4, 6, 8\}$,
[(Headache, no)] = $\{3, 7\}$,
[(Nausea, no)] = $\{1, 3, 6\}$,
[(Nausea, yes)] = $\{2, 4, 5, 7\}$,

and for Table 3.15

[(Temperature, high)] = $\{1, 3, 4, 5, 8\}$,
[(Temperature, very_high)] = $\{2, 3, 8\}$,
[(Temperature, normal)] = $\{3, 6, 7, 8\}$,
[(Headache, yes)] = $\{1, 2, 4, 5, 6, 8\}$,
[(Headache, no)] = $\{1, 3, 5, 7\}$,
[(Nausea, no)] = $\{1, 3, 6, 8\}$,
[(Nausea, yes)] = $\{2, 4, 5, 7, 8\}$.

The *characteristic set* $K_B(x)$ is the intersection of blocks of attribute-value pairs (a, v) for all attributes a from B for which $\rho(x, a)$ is known and $\rho(x, a) = v$. For Table 3.1 and $B = A$,

$K_A(1) = \{1,4,5\} \cap \{1,3,6\} = \{1\}$,
$K_A(2) = \{2\} \cap \{2,4,6,8\} \cap \{2,4,5,7\} = \{2\}$,
$K_A(3) = \{3,7\} \cap \{1,3,6\} = \{3\}$,

$K_A(4) = \{1,4,5\} \cap \{2,4,6,8\} \cap \{2,4,5,7\} = \{4\}$,
$K_A(5) = \{1,4,5\} \cap \{2,4,5,7\} = \{4,5\}$,
$K_A(6) = \{6,7\} \cap \{2,4,6,8\} \cap \{1,3,6\} = \{6\}$,
$K_A(7) = \{6,7\} \cap \{3,7\} \cap \{2,4,5,7\} = \{7\}$, and
$K_A(8) = \{2,4,6,8\}$.

and for Table 3.15 and $B = A$,

$K_A(1) = \{1,3,4,5,8\} \cap \{1,3,6,8\} = \{1,3,8\}$,
$K_A(2) = \{2,3,8\} \cap \{1,2,4,5,6,8\} \cap \{2,4,5,7,8\} = \{2,8\}$,
$K_A(3) = \{1,3,5,7\} \cap \{1,3,6,8\} = \{1,3\}$,
$K_A(4) = \{1,3,4,5,8\} \cap \{1,2,4,5,6,8\} \cap \{2,4,5,7,8\} = \{4,5,8\}$,
$K_A(5) = \{1,3,4,5,8\} \cap \{2,4,5,7,8\} = \{4,5,8\}$,
$K_A(6) = \{3,6,7,8\} \cap \{1,2,4,5,6,8\} \cap \{1,3,6,8\} = \{6,8\}$,
$K_A(7) = \{3,6,7,8\} \cap \{1,3,5,7\} \cap \{2,4,5,7,8\} = \{7\}$, and
$K_A(8) = \{1,2,4,5,6,8\}$.

The characteristic set $K_B(x)$ may be interpreted as the smallest set of cases that are indistinguishable from x using all attributes from B, 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. For further properties of characteristic sets see (Grzymala-Busse, 2003; Grzymala-Busse, 2004A; Grzymala-Busse, 2004B; Grzymala-Busse, 2004C). Incomplete decision tables in which all attribute values are lost, from the viewpoint of rough set theory, were studied for the first time in (Grzymala-Busse and Wang, 1997), where two algorithms for rule induction, modified to handle lost attribute values, were presented. This approach was studied later in (Stefanowski, 2001; Stefanowski and Tsoukias, 1999; Stefanowski and Tsoukias, 2001).

Incomplete decision tables in which all missing attribute values are "do not care" conditions, from the view point of rough set theory, were studied 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 attribute. Originally such values were replaced by all values from the entire domain of the attribute, later, by attribute values restricted to the same concept to which a case with a missing attribute value belongs. Such incomplete decision tables, with all missing attribute values being "do not care conditions", were also studied in (Kryszkiewicz, 1995; Kryszkiewicz, 1999). Both approaches to missing attribute values were generalized in (Grzymala-Busse, 2003; Grzymala-Busse, 2004A; Grzymala-Busse, 2004B; Grzymala-Busse, 2004C).

3.2 Lower and Upper Approximations

Any finite union of characteristic sets of B is called a *B-definable* set. The lower approximation of the concept X is the largest definable sets that is contained in X and the upper approximation of X is the smallest definable set that contains X. In general, for incompletely specified decision tables lower and upper approximations may be defined in a few different ways (Grzymala-Busse, 2003; Grzymala-Busse, 2004A; Grzymala-Busse, 2004B; Grzymala-Busse, 2004C). Here we will quote the most useful definition of lower and upper approximations from the view point of Data Mining. A *concept* B-lower approximation of the concept X is defined as follows:

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

A concept B-upper approximation of the concept X is defined as follows:

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

For the decision table presented in Table 3.1, the concept A-lower and A-upper approximations are

$$\begin{aligned} \underline{A}\{1,2,4,8\} &= \{1,2,4\}, \\ \underline{A}\{3,5,6,7\} &= \{3,6,7\}, \\ \overline{A}\{1,2,4,8\} &= \{1,2,4,6,8\}, \\ \overline{A}\{3,5,6,7\} &= \{3,4,5,6,7\}, \end{aligned}$$

and for the decision table from Table 3.15, the concept A-lower and A-upper approximations are

$$\begin{aligned} \underline{A}\{1,2,4,8\} &= \{2,8\}, \\ \underline{A}\{3,5,6,7\} &= \{7\}, \\ \overline{A}\{1,2,4,8\} &= \{1,2,3,4,5,6,8\}, \\ \overline{A}\{3,5,6,7\} &= \{1,3,4,5,6,7,8\}. \end{aligned}$$

3.3 Rule Induction—MLEM2

The MLEM2 rule induction algorithm is a modified version of the algorithm LEM2, see chapter 13 in this volume. Rules induced from the lower approximation of the concept *certainly* describe the concept, so they are called *certain*. On the other hand, rules induced from the upper approximation of the concept describe the concept only *possibly* (or *plausibly*), so they are called *possible*

(Grzymala-Busse, 1988). MLEM2 may induce both certain and possible rules from a decision table with some missing attribute values being lost and some missing attribute values being "do not care" conditions, while some attributes may be numerical. For rule induction from decision tables with numerical attributes see (Grzymala-Busse, 2004A). MLEM2 handles missing attribute values by computing (in a different way than in LEM2) blocks of attribute-value pairs, and then characteristic sets and lower and upper approximations. All these definitions are modified according to the two previous subsections, the algorithm itself remains the same.

Rule sets in the LERS format (every rule is equipped with three numbers, the total number of attribute-value pairs on the left-hand side of the rule, the total number of examples correctly classified by the rule during training, and the total number of training cases matching the left-hand side of the rule), induced from the decision table presented in Table 3.1 are:
certain rule set:

2, 1, 1
(Temperature, high) & (Nausea, no) -> (Flu, yes)
2, 2, 2
(Headache, yes) & (Nausea, yes) -> (Flu, yes)
1, 2, 2
(Temperature, normal) -> (Flu, no)
1, 2, 2
(Headache, no) -> (Flu, no)

and possible rule set:

1, 3, 4
(Headache, yes) -> (Flu, yes)
2, 1, 1
(Temperature, high) & (Nausea, no) -> (Flu, yes)
2, 1, 2
(Temperature, high) & (Nausea, yes) -> (Flu, no)
1, 2, 2
(Temperature, normal) -> (Flu, no)
1, 2, 2
(Headache, no) -> (Flu, no)

Rule sets induced from the decision table presented in Table 3.15 are:
certain rule set:

2, 2, 2
(Temperature, very_high) & (Nausea, yes) -> (Flu, yes)
3, 1, 1
(Temperature, normal) & (Headache, no) & (Nausea, yes) -> (Flu, no)

and possible rule set:

1, 4, 6
(Headache, yes) -> (Flu, yes)
1, 2, 3
(Temperature, very_high) -> (Flu, yes)
1, 2, 5
(Temperature, high) -> (Flu, no)
1, 3, 4
(Temperature, normal) -> (Flu, no)

3.4 Other Approaches to Missing Attribute Values

Through this section we assumed that the incomplete decision tables may only consist of lost values or do not care conditions. Note that the MLEM2 algorithm is able to handle not only these two types of tables but also decision tables with a mixture of these two cases, i.e., tables with some lost attribute values and with other missing attribute values being do not care conditions. Furthermore, other interpretations of missing attribute values are possible as well, see (Grzymala-Busse, 2003; Grzymala-Busse, 2004A).

4. Conclusions

In general, there is no best, universal method of handling missing attribute values. On the basis of existing research on comparison such methods (Grzymala-Busse and Hu, 2000; Grzymala-Busse and Siddhaye, 2004; Lakshminarayan *et al.*, 1999) we may conclude that for every specific data set the best method of handling missing attribute values should be chosen individually, using as the criterion of optimality the arithmetic mean of many multi-fold cross validation experiments (Weiss and Kulikowski, 1991). Similar conclusions may be drawn for decision tree generation (Quinlan, 1989).

References

Allison P.D. *Missing Data.* Sage Publications, 2002.

Brazdil P. and Bruha I. Processing unknown attribute values by ID3. Proceedings of the 4-th Int. Conference Computing and Information, Toronto, 1992, 227 - 230

Breiman L., Friedman J.H., Olshen R.A., Stone C.J. *Classification and Regression Trees.* Wadsworth & Brooks, Monterey, CA, 1984.

Bruha I. Meta-learner for unknown attribute values processing: Dealing with inconsistency of meta-databases. *Journal of Intelligent Information Systems* **22** 71–87, 2004.

Chiu, D. K. and Wong A. K. C. Synthesizing knowledge: A cluster analysis approach using event-covering. *IEEE Trans. Syst., Man, and Cybern.* **SMC-16** 251–259, 1986.

Clark P. and Niblett T. The CN2 induction algorithm. *Machine Learning* **3** 261–283, 1989.

Dardzinska A. and Ras Z.W. Chasing unknown values in incomplete information systems. Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining, Melbourne, FL, November 1922, 24–30, 2003A.

Dardzinska A. and Ras Z.W. On rule discovery from incomplete information systems. Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining, Melbourne, FL, November 1922, 31–35, 2003B.

Greco S., Matarazzo B., and Slowinski R. Dealing with missing data in rough set analysis of multi-attribute and multi-criteria decision problems. In *Decision Making: Recent developments and Worldwide Applications*, ed. by S. H. Zanakis, G. Doukidis, and Z. Zopounidis, Kluwer Academic Publishers, Dordrecht, Boston, London, 2000, 295–316.

Grzymala-Busse J.W. Knowledge acquisition under uncertainty—A rough set approach. *Journal of Intelligent & Robotic Systems* **1** (1988) 3–16.

Grzymala-Busse J.W. On the unknown attribute values in learning from examples. Proc. of the ISMIS-91, 6th International Symposium on Methodologies for Intelligent Systems, Charlotte, North Carolina, October 16–19, 1991. Lecture Notes in Artificial Intelligence, vol. 542, Springer-Verlag, Berlin, Heidelberg, New York, 1991, 368–377.

Grzymala-Busse J.W. LERS—A system for learning from examples based on rough sets. In Intelligent Decision Support. Handbook of Applications and Advances of the Rough Sets Theory, ed. by R. Slowinski, Kluwer Academic Publishers, Dordrecht, Boston, London, 1992, 3–18.

Grzymala-Busse J.W. A new version of the rule induction system LERS, *Fundamenta Informaticae* **31** (1997) 27–39.

Grzymala-Busse J.W. MLEM2: A new algorithm for rule induction from imperfect data. Proceedings of the 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, Annecy, France, July 1–5, 2002, 243–250.

Grzymala-Busse J.W. Rough set strategies to data with missing attribute values. Proceedings of the Workshop on Foundations and New Directions in Data Mining, associated with the third IEEE International Conference on Data Mining, Melbourne, FL, November 1922, 2003, 56–63.

Grzymala-Busse J.W. Data with missing attribute values: Generalization of indiscernibility relation and rule induction. Transactions on Rough Sets, Lecture Notes in Computer Science Journal Subline, Springer-Verlag, vol. **1** 78–95, 2004A.

Grzymala-Busse J.W. Characteristic relations for incomplete data: A generalization of the indiscernibility relation. Proceedings of the RSCTC'2004, the Fourth International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden, June 15, 2004. Lecture Notes in Artificial Intelligence 3066, Springer-Verlag pp.244–253, 2004B.

Grzymala-Busse J.W. Rough set approach to incomplete data. Proceedings of the ICAISC'2004, the Seventh International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, June 711, 2004. Lecture Notes in Artificial Intelligence 3070, Springer-Verlag pp.50–55, 2004.

Grzymala-Busse J.W., Grzymala-Busse W.J., and Goodwin L.K. A comparison of three closest fit approaches to missing attribute values in preterm birth data. *International Journal of Intelligent Systems* **17** (2002) 125–134.

Grzymala-Busse, J.W. and Hu, M. A comparison of several approaches to missing attribute values in Data Mining. Proceedings of the Second International Conference on Rough Sets and Current Trends in Computing RSCTC'2000, Banff, Canada, October 16–19, 2000, 340–347.

Grzymala-Busse, J.W. and Wang A.Y. Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. Proc. of the Fifth International Workshop on Rough Sets and Soft Computing (RSSC'97) at the Third Joint Conference on Information Sciences (JCIS'97), Research Triangle Park, NC, March 2-5, 1997, 69–72.

Grzymala-Busse J.W. and Siddhaye S. Rough set approaches to rule induction from incomplete data. Proceedings of the IPMU'2004, the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Perugia, Italy, July 49, 2004, vol. 2, 923930.

Imielinski T. and Lipski W. Jr. Incomplete information in relational databases, *Journal of the ACM* **31** (1984) 761–791.

Kononenko I., Bratko I., and Roskar E. Experiments in automatic learning of medical diagnostic rules. Technical Report, Jozef Stefan Institute, Lljubljana, Yugoslavia, 1984

Kryszkiewicz M. Rough set approach to incomplete information systems. Proceedings of the Second Annual Joint Conference on Information Sciences, Wrightsville Beach, NC, September 28–October 1, 1995, 194–197.

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

Lakshminarayan K., Harp S.A., and Samad T. Imputation of missing data in industrial databases. *Applied Intelligence* **11** (1999) 259 – 275.

Latkowski, R. On decomposition for incomplete data. *Fundamenta Informaticae* **54** (2003) 1-16.

Latkowski R. and Mikolajczyk M. Data decomposition and decision rule joining for classification of data with missing values. Proceedings of the RSCTC'2004, the Fourth International Conference on Rough Sets and Current Trends in Computing, Uppsala, Sweden, June 1–5, 2004. Lecture Notes in Artificial Intelligence 3066, Springer-Verlag 2004, 254–263.

Lipski W. Jr. On semantic issues connected with incomplete information databases. *ACM Transactions on Database Systems* **4** (1979), 262–296.

Lipski W. Jr. On databases with incomplete information. *Journal of the ACM* **28** (1981) 41–70.

Little R.J.A. and Rubin D.B. *Statistical Analysis with Missing Data,* Second Edition, J. Wiley & Sons, Inc., 2002.

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

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

Pawlak Z., Grzymala-Busse J.W., Slowinski R., and Ziarko, W. Rough sets. *Communications of the ACM* **38** (1995) 88–95.

Polkowski L. and Skowron A. (eds.) *Rough Sets in Knowledge Discovery, 2, Applications, Case Studies and Software Systems,* Appendix 2: Software Systems. Physica Verlag, Heidelberg New York (1998) 551–601.

Quinlan J.R. Unknown attribute values in induction. Proc. of the 6-th Int. Workshop on Machine Learning, Ithaca, NY, 1989, 164 – 168.

Quinlan J. R. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, San Mateo CA (1993).

Schafer J.L. *Analysis of Incomplete Multivariate Data.* Chapman and Hall, London, 1997.

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

Stefanowski J. Algorithms of Decision Rule Induction in Data Mining. Poznan University of Technology Press, Poznan, Poland (2001).

Stefanowski J. and Tsoukias A. On the extension of rough sets under incomplete information. Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining, and Granular-Soft Computing, RSFDGrC'1999, Ube, Yamaguchi, Japan, November 8–10, 1999, 73–81.

Stefanowski J. and Tsoukias A. Incomplete information tables and rough classification. *Computational Intelligence* **17** (2001) 545–566.

Weiss S. and Kulikowski C.A. *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems,* chapter *How to Estimate the True Performance of a Learning System,* pp. 17–49, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1991.

Wong K.C. and Chiu K.Y. Synthesizing statistical knowledge for incomplete mixed-mode data. IEEE Transactions on Pattern Analysis and Machine Intelligence **9** (1987) 796805.

Wu X. and Barbara D. Learning missing values from summary constraints. *ACM SIGKDD Explorations Newsletter* **4** (2002) 21 – 30.

Wu X. and Barbara D. Modeling and imputation of large incomplete multidimensional datasets. Proc. of the 4-th Int. Conference on Data Warehousing and Knowledge Discovery, Aix-en-Provence, France, 2002, 286 – 295

Yao Y.Y. On the generalizing rough set theory. Proc. of the 9th Int. Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'2003), Chongqing, China, October 19–22, 2003, 44–51.