ITC 2/46
Journal of Information Technology and Control
Vol. 46 / No. 2 / 2017
pp. 183-193
DOI 10.5755/j01.itc.46.2.17330

A Comparison of Mining Incomplete and Inconsistent Data

Received 2016/12/23 | Accepted after revision 2017/05/04

http://dx.doi.org/10.5755/j01.itc.46.2.17330

A Comparison of Mining Incomplete and Inconsistent Data

Patrick G. Clark, Cheng Gao

University of Kansas, Department of Electrical Engineering and Computer Science, Lawrence, KS 66045, USA, e-mails: patrick.g.clark@gmail.com; cheng.gao@ku.edu

Jerzy W. Grzymala-Busse

University of Kansas, Department of Electrical Engineering and Computer Science, Lawrence, KS 66045, USA, University of Information Technology and Management, Department of Expert Systems and Artificial Intelligence, 35-225 Rzeszow, Poland, e-mail: jerzy@ku.edu

Corresponding author: patrick.g.clark@gmail.com

We present experimental results on a comparison of incompleteness and inconsistency. We used two interpretations of missing attribute values: lost values and "do not care" conditions. Our experiments were conducted on 204 data sets, including 71 data sets with lost values, 71 data sets with "do not care" conditions and 62 inconsistent data sets, created from eight original numerical data sets. We used the Modified Learning from Examples Module version 2 (MLEM2) rule induction algorithm for data mining, combined with three types of probabilistic approximations: lower, middle and upper. We used an error rate, computed by ten-fold cross validation, as the criterion of quality. There is experimental evidence that incompleteness is worse than inconsistency for data mining (two-tailed test, 5% level of significance). Additionally, lost values are better than "do not care" conditions, again, with regards to the error rate, and there is a little difference in an error rate between three types of probabilistic approximations.

KEYWORDS: Incomplete data, lost values, "do not care" conditions, inconsistent data, rough set theory, probabilistic approximations, MLEM2 rule induction algorithm.

Introduction

A complete data set, i.e., a data set having all attribute values specified, is consistent if for any two cases with the same attribute values, both cases belong to the same concept (class). Another definition of consistency is based on rough set theory: a complete data set is consistent if for any concept its

lower and upper approximations are equal [10, 11].

In some situations the data set being mined is incomplete, some of the attribute values are missing. We use two interpretations of missing attribute values, lost values and "do not care" conditions [5, 6]. Lost values are, e.g., erased, for data mining we use the existing, specified attribute values. "Do not care" conditions are interpreted differently, e.g., an expert refused to tell the attribute value, so such value may be replaced by any value from the attribute domain.

The main objective of our paper is to compare mining incomplete and inconsistent data in terms of an error rate computed as a result of ten-fold cross validation. Using eight numerical data sets, we discretized each of them and then converted to a symbolic and consistent data set with intervals as attribute values. We then randomly replaced some of the intervals with "?"s, representing lost values. This process was conducted incrementally, starting by randomly replacing 5% of the intervals with missing attribute values, and then an additional 5%, until a case occurred with all attribute values missing. The process was then attempted twice more with the maximum percentage and if again a case occurred with all attribute values missing, the process was terminated for that data set. Finally, in all incomplete data sets, we replaced all "?"s by "*"s, representing "do not care" conditions. The new data sets, with missing attribute values, were as close as possible to the original data sets, having the same number of attributes, cases, and concepts.

Additionally, any original data set was discretized with a controlled level of inconsistency, starting from about 5%, with the same increment of about 5%. Due to the nature of discretization, the levels of inconsistency were only approximately equal to 5%, 10%, etc. Our way of generation of inconsistent data preserved as much as possible the original data set. Again, the number of attributes, cases and concepts were not changed.

All such incomplete and inconsistent data sets were validated using the same setup, based on rule induction by the MLEM2 rule induction algorithm and the same system for ten-fold cross validation.

To the best of our knowledge, no research comparing incompleteness with inconsistency was ever undertaken. However, our results should be taken with a grain of salt since the measures of incompleteness and inconsistency are different. We measure both of them in the most natural way: for a data set, incompleteness is measured by the percentage of missing attribute values, or percentage of missing attribute values to the total number of cases in the data set. Inconsistency is measured by the level of inconsistency, i.e., percentage of conflicting cases to the number of cases. Yet the former measure is local, it is associated with the attribute-value pairs, while the latter is global, it is computed by comparing entire cases. On the other hand, if we want to compare incompleteness with inconsistency, there is no better way than using these two measures.

In our experiments we used the idea of a probabilistic approximation, with a probability α, as an extension of the standard approximation, well known in rough set theory. For $\alpha = 1$, the probabilistic approximation is identical with the lower approximation; for very small α, it is identical with the upper approximation. Research on properties of probabilistic approximations was first reported in [13] and then was continued in many other papers, for example, [12, 15–17].

Incomplete data sets are usually analyzed using special approximations such as singleton, subset and concept [5, 6]. For incomplete data sets probabilistic approximations were used for the first time in [7]. The first experimental results using probabilistic approximations were published in [3]. In experiments reported in this paper, we used concept probabilistic approximations.

A preliminary version of this paper was presented at the ICIST 2016, the 22nd International Conference on Information and Software Technologies [2].

Incomplete Data

Data sets may be presented in the form of a decision table. An example of such a decision table is shown in Table 1. Rows of the decision table represent cases and columns represent variables. The set of all cases will be denoted by U. In Table 1, $U = \{1, 2, 3, 4, 5, 6, 7\}$. 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, $A = \{Age, Cholesterol, Weight\}$. The value for a case x and an attribute a will be denoted by $a(x)$.

Table 1

A data set with numerical attributes

Attributes				Decision
Case	Age	Cholesterol	Weight	Risk
1	20	180	140	low
2	60	200	180	low
3	40	220	160	low
4	50	200	180	low
5	60	220	180	high
6	40	220	180	high
7	50	180	220	high

Table 2 presents an example of the discretized and consistent data set. All attribute values are intervals and as such are considered symbolic.

Table 2

A discretized, consistent data set

Attributes				Decision
Case	Age	Cholesterol	Weight	Risk
1	20..45	180..210	140..170	low
2	45..60	180..210	170..210	low
3	20..45	210..220	140..170	low
4	45..60	180..210	170..210	low
5	45..60	210..220	170..210	high
6	20..45	210..220	170..210	high
7	45..60	180..210	210..220	high

Table 3 presents an example of an incomplete data set with lost values, denoted by "?"s [9, 14]. The percentage of missing attribute values is the total number of missing attribute values, equal to eight, divided by the total number of attribute values, equal to 21, i.e., the percentage of missing attribute values is 38.1%.

Table 4 presents an example of an incomplete data set with "do not care" conditions, denoted by "*"s [9, 14].

Table 5 represent an inconsistent data set. This data set was created from the data set from Table 1. The numerical data set from Table 1 was discretized with 30% level of inconsistency. Cases 3 and 6 are conflicting, so the level of inconsistency is $2/7 \approx 30\%$.

Table 3

An incomplete data set with lost values

Attributes				Decision
Case	Age	Cholesterol	Weight	Risk
1	?	180..210	140..170	low
2	45..60	?	170..210	low
3	20..45	?	?	low
4	45..60	180..210	170..210	low
5	45..60	?	170..210	high
6	?	210..220	?	high
7	45..60	180..210	?	high

Table 4

An incomplete data set with "do not care" conditions

Attributes				Decision
Case	Age	Cholesterol	Weight	Risk
1	*	180..210	140..170	low
2	45..60	*	170..210	low
3	20..45	*	*	low
4	45..60	180..210	170..210	low
5	45..60	*	170..210	high
6	*	210..220	*	high
7	45..60	180..210	*	high

Table 5

An inconsistent data set

Attributes				Decision
Case	Age	Cholesterol	Weight	Risk
1	20..45	180..210	140..210	low
2	45..60	180..210	140..210	low
3	20..45	210..220	140..210	low
4	45..60	180..210	140..210	low
5	45..60	210..220	140..210	high
6	20..45	210..220	140..210	high
7	45..60	180..210	210..220	high

A fundamental idea of rough set theory [10] is an indiscernibility relation, defined for complete data sets. Let B be a nonempty subset of the set A for all attributes. The indiscernibility relation $R(B)$ is a relation on U defined for $x, y \in U$ by

$$(x, y) \in R(B) \text{ if and only if } \forall a \in B(a(x) = a(y)).$$

The indiscernibility relation $R(B)$ is an equivalence relation. Equivalence classes of $R(B)$ are called *elementary sets* of B and are denoted by $[x]_B$. A subset of U is called *B-definable* if it is a union of *elementary sets* of B.

The set X of all cases defined by the same value of the decision d is called a *concept*. The set of all concepts is denoted by $\{d\}^*$. For example, a concept associated with the value *low* of the decision *Risk* is the set {1, 2, 3, 4}. The largest B-definable set contained in X is called the *B-lower approximation* of X, denoted by $\underline{appr_B}(X)$, and defined as follows

$$\cup \{[x]_B | [x]_B \subseteq X\}.$$

The smallest B-definable set containing X, denoted by $\overline{appr_B}(X)$ is called the *B-upper approximation* of X, and is defined by

$$\cup \{[x]_B | [x]_B \cap X \neq \emptyset\}.$$

For Table 5,

$$\underline{appr_A}(\{1, 2, 3, 4\}) = \{1, 2, 4\},$$

and

$$\overline{appr}_A(\{1, 2, 3, 4\}) = \{1, 2, 3, 4, 6\}.$$

The level of inconsistency may be defined as follows

$$1 - \frac{\sum_{X \in \{d\}^*} \left| \underline{appr_A}(X) \right|}{|U|}$$

where $|S|$ denotes the cardinality of the set S.

For a variable a and its value v, (a, v) is called a variable-value pair. A *block* of (a, v), denoted by $[(a, v)]$, is the set $\{x \in U | a(x) = v\}$ [4]. For incomplete decision tables the definition of a block of an attribute-value pair is modified in the following way.

- If for an attribute a and a case x, if $a(x) =?$, i. e., the attribute value is lost, the case x should not be included in any blocks $[(a,v)]$ for all values v of attribute a,
- If for an attribute a and a case x, if $a(x) = *$, i.e., the attribute values is a "do not care" condition, the case x should be included in blocks $[(a, v)]$ for all specified values v of attribute a.

For the data set with lost values from Table 3 the blocks of attribute-value pairs are:

[(Age, 20..45)] = {3},

[(Age, 45..60)] = {2, 4, 5, 7},

[(Cholesterol, 180..210)] = {1, 4, 7},

[(Cholesterol, 210..220)] = {6},

[(Weight, 180..210)] = {1}, and

[(Weight, 170..220)] = {2, 4, 5}.

For a case $x \in U$ and $B \subseteq A$, 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 $a(x)$ is specified, then $K(x, a)$ is the block $[(a, a(x))]$ of attribute a and its value $a(x)$,
- If $a(x) =$? then the set $K(x, a) = U$, where U is the set of all cases.

For Table 3 and $B = A$,

$K_A(1) = \{1\}$,

$K_A(2) = \{2, 4, 5\}$,

$K_A(3) = \{3\}$,

$K_A(4) = \{4\}$,

$K_A(5) = \{2, 4, 5\}$,

$\underline{K_A}(6) = \{6\}$, and

$K_A(7) = \{4, 7\}$.

On the other hand, for the data set with "do not care" conditions from Table 4 the blocks of attribute-value pairs are:

[(Age, 20..45)] = {1, 3, 6},

[(Age, 45..60)] = {1, 2, 4, 5, 6, 7},

[(Cholesterol, 180..210)] = {1, 2, 3, 4, 5, 7},

[(Cholesterol, 210..220)] = {2, 3, 5, 6},

[(Weight, 180..210)] = {1, 3, 6, 7}, and

[(Weight, 170..220)] = {2, 3, 4, 5, 6, 7}.

For a case $x \in U$ and $B \subseteq A$, 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 $a(x)$ is specified, then $K(x, a)$ is the block $[(a, a(x))]$ of attribute a and its value $a(x)$,
- If $a(x) = *$ then the set $K(x,a) = U$, where U is the set of all cases.

For Table 4 and $B = A$,

$K_A(1) = \{1, 3, 7\}$,

$K_A(2) = \{2, 4, 5, 6, 7\}$,

$K_A(3) = \{1, 3, 6\}$,

$K_A(4) = \{2, 4, 5, 7\}$,

$K_A(5) = \{2, 4, 5, 6, 7\}$,

$K_A(6) = \{2, 3, 5, 6\}$, and

$K_A(7) = \{1, 2, 4, 5, 7\}$.

First we will quote some definitions from [8]. Let X be a subset of U. The *B-singleton lower approximation* of X, denoted by $\underline{appr}_B^{singleton}(X)$, is defined by

$$\{x | x \in U, K_B(x) \subseteq X\}.$$

The *B-singleton upper approximation* of X, denoted by $\overline{appr}_B^{singleton}(X)$, is defined by

$$\{x | x \in U, K_B(x) \cap X \neq \emptyset\}.$$

The *B-subset lower approximation* of X, denoted by $\underline{appr}_B^{subset}(X)$, is defined by

$$\cup \{K_B(x) | x \in U, K_B(x) \subseteq X\}.$$

The *B-subset upper approximation* of X, denoted by $\overline{appr}_B^{subset}(X)$, is defined by

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

The *B-concept lower approximation* of X, denoted by $\underline{appr}_B^{concept}(X)$, is defined by

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

The *B-concept upper approximation* of X, denoted by $\overline{appr}_B^{concept}(X)$, is defined by

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

For Table 3 and X = {5, 6, 7}, all A-singleton, A-subset and A-concept lower and upper approximations are:

$\underline{appr}_A^{singleton}(X) = \{6\}$,
$\overline{appr}_A^{singleton}(X) = \{2, 5, 6, 7\}$,
$\underline{appr}_A^{subset}(X) = \{6\}$,
$\overline{appr}_A^{subset}(X) = \{2, 4, 5, 6, 7\}$,
$\underline{appr}_A^{concept}(X) = \{6\}$,
$\overline{appr}_A^{concept}(X) = \{2, 4, 5, 6, 7\}$.

On the other hand, for Table 4 and $X = \{5, 6, 7\}$, all A-singleton, A-subset and A-concept lower and upper approximations are:

$\underline{appr}_A^{singleton}(X) = \emptyset$,
$\overline{appr}_A^{singleton}(X) = U$,
$\underline{appr}_A^{subset}(X) = \emptyset$,
$\overline{appr}_A^{subset}(X) = U$,
$\underline{appr}_A^{concept}(X) = \emptyset$,
$\overline{appr}_A^{concept}(X) = U$.

Probabilistic approximations

Definitions of lower and upper approximations may be extended to the probabilistic approximations [7]. In our experiments we used only concept approximations, so we will cite the corresponding definition only for the concept approximation. A B-concept probabilistic approximation of the set X with the threshold α, $0 < \alpha \leq 1$, denoted by $appr_{\alpha,B}^{concept}(X)$, is defined by

$$\cup \{K_B(x) | x \in X, \Pr(X | K_B(x)) \geq \alpha\},$$

where $Pr(X|K_B(x)) = \frac{|X \cap K_B(x)|}{|K_B(x)|}$ is the conditional probability of X given $K_B(x)$.

Since we are using only B-concept probabilistic approximations, for the sake of simplicity we will call them B-probabilistic approximations. Additionally, if $B = A$, B-probabilistic approximations will be called simply probabilistic approximations and will be denoted by $appr_\alpha(X)$.

Note that if $\alpha = 1$, the probabilistic approximation is equal to the concept lower approximation and if α is small, close to 0, in our experiments it is 0.001, the probabilistic approximation is equal to the concept upper approximation.

For Table 3 and the concept $X = \{5, 6, 7\}$, there exist the following distinct probabilistic approximations:

$appr_{1.0}(X) = \{6\}$,
$appr_{0.5}(X) = \{4, 6, 7\}$,
$appr_{0.333}(X) = \{2, 4, 5, 6, 7\}$.

For Table 4 and the concept $X = \{5, 6, 7\}$, there exist the

following distinct probabilistic approximations:

$$appr_{1.0}(X) = \emptyset,$$
$$appr_{0.6}(X) = \{2, 4, 5, 6, 7\},$$
$$appr_{0.5}(X) = \{2, 3, 4, 5, 6, 7\},$$
$$appr_{0.4}(X) = U.$$

A special probabilistic approximations with $\alpha = 0.5$ will be called *middle* approximations.

Figure 1

Error rates for two series of data sets originated from the *Australian* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"

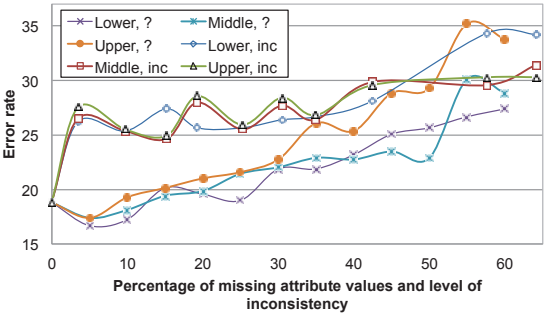
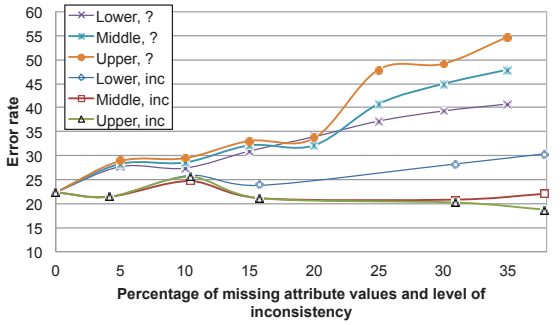


Figure 2

Error rates for two series of data sets originated from the *Ecoli* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"



Experiments

Our experiments are based on eight data sets, all taken from the University of California at Irvine *Machine Learning Repository*. Essential information about these data sets is presented in Table 6. All eight data sets are numerical.

For any data set we created a series of incomplete data sets in the following way: first, the numerical data set was discretized using the agglomerative cluster analysis method [1]. Then we randomly replaced 5% of specified attribute values by symbols of "?", denoting missing attribute values. After that, we replaced randomly and incrementally, with an increment equal to 5%, new specified attribute values by symbols "?", preserving old ones. The process continued until we reached the point of having a case with all attribute values being "?"s. Then we returned to the one but last step and tried to add, randomly, 5% of "?"s again. If after three such attempts the result was still a case with "?"s as values for all attributes, the process was terminated. For example, for the *australian* data set such maximum for missing attribute values is 60%. New incomplete data sets, with "do not care" conditions, were created by replacing all "?"s by "*"s, in respective data sets.

For each original numerical data set, a series of inconsistent data sets was created by discretization, using the same agglomerative cluster analysis method as for the missing data sets. However, different levels of inconsistency were used as a stopping condition for discretization. Note that due to the nature of discretization, only some levels of inconsistency were possible to accomplish, so the levels of inconsistency are not as regular as percentage of missing attribute values. For example, for the *australian* data set these

Figure 3

Error rates for two series of data sets originated from the *Hepatitis* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"

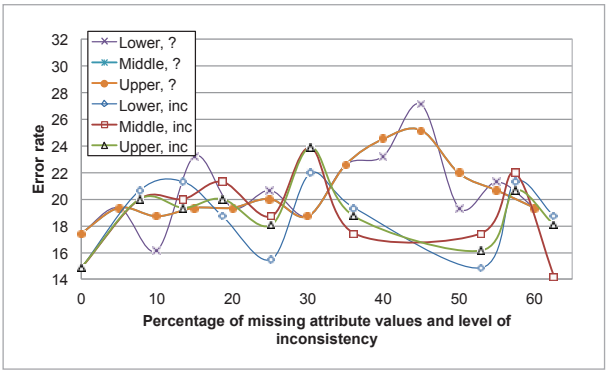


Figure 4

Error rates for two series of data sets originated from the *Image Segmentation* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"

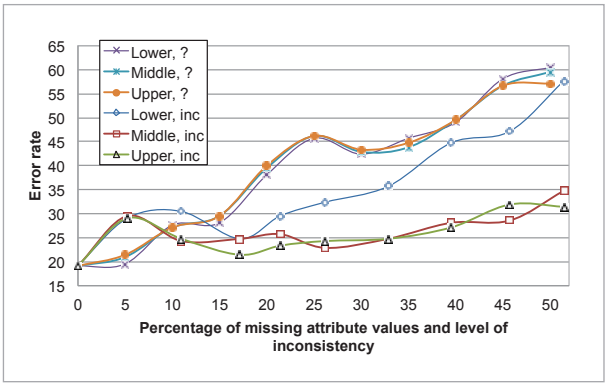
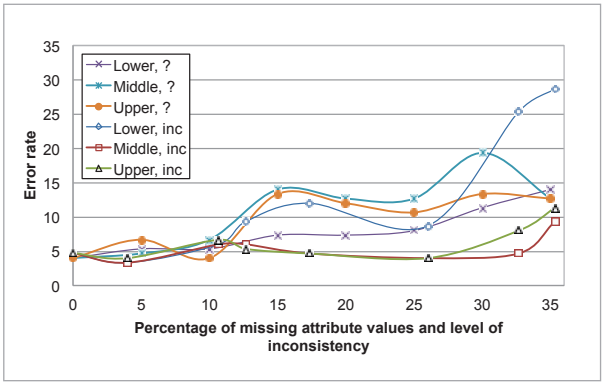


Figure 6

Error rates for two series of data sets originated from the *Iris* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"



levels are 3.48. 9.71, 15.22 etc. instead of 5, 10, 15, as for the percentage of missing attribute values, though we tried to keep both series as close as possible.

Our experiments were conducted on 204 data sets, including 71 data sets with lost values, 71 data sets with "do not care" conditions and 62 inconsistent data sets, created from eight original numerical data sets, among these data sets, eight discretized and consistent data sets were used as special cases for both incomplete and inconsistent data sets.

For every data set we used three different probabilistic approximations for rule induction (lower, middle and upper). Thus we had 24 different approaches to rule induction from data sets with lost values. Obviously, for "do not care" conditions we had the same number of 24 distinct approaches to rule induction. For rule induction we used the MLEM2 rule induction algorithm, a part of the Learning from Examples based on Rough Sets (LERS) data mining system [4].

For lost values we compared incomplete data with inconsistent ones for the same type of probabilistic approximations, using the Wilcoxon matched-pairs signed rank test, with 5% level of significance, two-tailed test. Since we had 71 incomplete data sets and 62 inconsistent data sets, missing pairs were constructed by interpolation. Results of experiments rates for which there were no matching results, either incomplete or inconsistent, are not depicted in Figures 1–8.

Figure 5

Error rates for two series of data sets originated from the *Ionosphere* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"

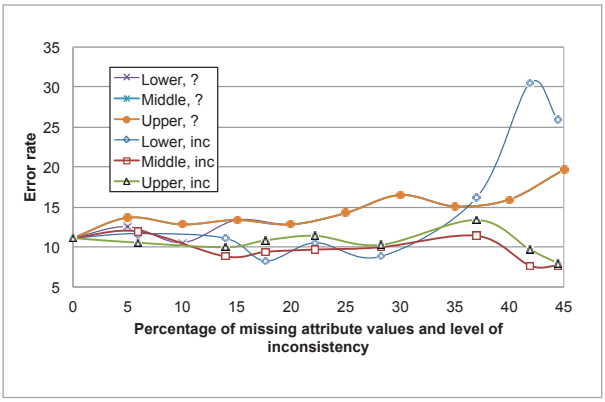


Figure 7

Error rates for two series of data sets originated from the *Pima* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"

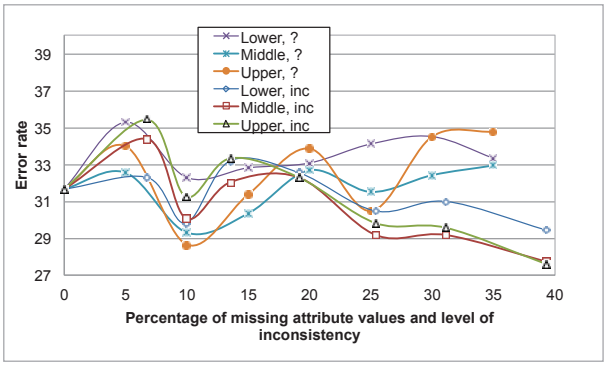
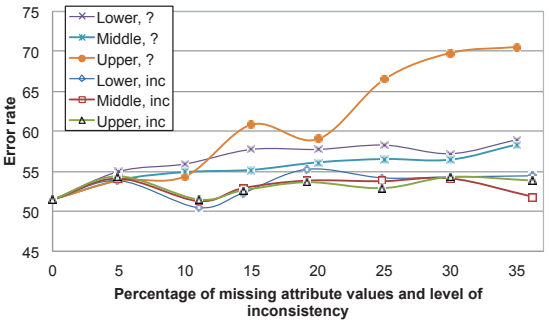


Figure 8

Error rates for two series of data sets originated from the *Yeast* data set. Lost values are denoted by "?", inconsistent data are denoted by "inc"



Results of our experiments, presented in Figures 1–8, are: among 24 approaches, in 12 inconsistency was better (the error rate was smaller for inconsistent data). The *australian* data set was an exception, for all three probabilistic approximations the error rate was significantly smaller for incomplete data sets. For remaining nine approaches the difference between incompleteness and inconsistency was statistically insignificant.

For incomplete data sets with "do not care" conditions, results, presented in Figures 9–16, were more decisive. For 15 out of 24 combinations, inconsistency was better than incompleteness, for remaining nine combinations the difference between incompleteness and inconsistency was statistically insignificant.

Table 6

Data sets

Data set	Cases	Number of attributes	Concepts
Australian	690	14	2
Ecoli	336	8	8
Hepatitis	155	19	2
Image Segmentation	210	19	7
Ionosphere	351	34	2
Iris	150	4	3
Pima	768	8	2
Yeast	1484	8	9

Figure 9

Error rates for two series of data sets originated from the *Australian* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"

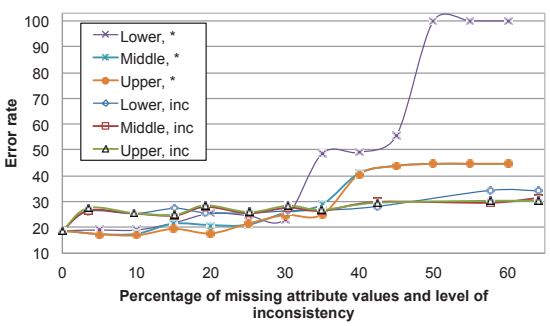


Figure 10

Error rates for two series of data sets originated from the *Ecoli* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"

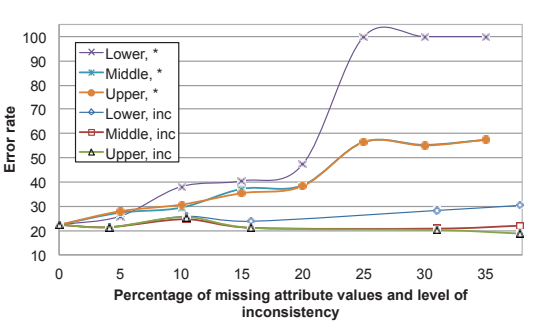


Figure 11

Error rates for two series of data sets originated from the *Hepatitis* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"

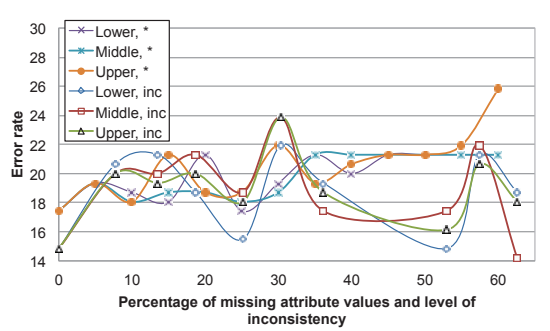
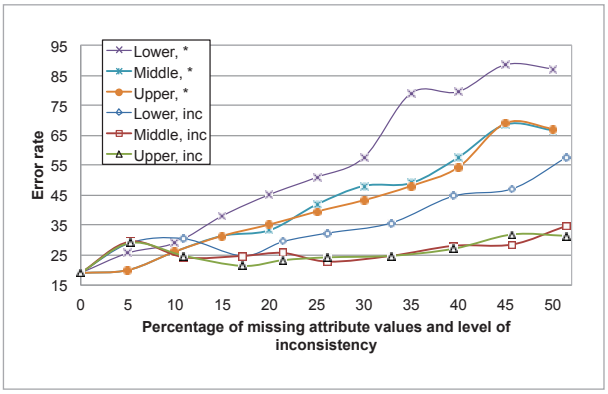


Figure 12

Error rates for two series of data sets originated from the *Image Segmentation* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"



Taking onto account both interpretations of missing attribute values, lost values and "do not care" conditions, we conclude that incompleteness is worse than inconsistency for data mining.

With our experimental results, we also compared the error rate, computed by ten-fold cross validation, for two interpretations of missing attribute values: lost values and "do not care" conditions.

For 14 out of 24 combinations, the error rate for data sets with lost values was smaller than the error rate for data sets with "do not care" conditions. For remaining ten approaches the difference between lost values and "do not care" conditions was statistically insignificant.

Finally, we compared all three types of probabilistic approximations: lower, middle and upper, separately for lost values and for "do not care" conditions. For this comparison we used the Friedman Rank Sums test with 5% of significance level. For a fixed interpretation of missing attribute value, the total number of combinations was again 24 (three type of approximations and eight data sets).

For lost values, lower approximations were better than middle approximation for two combinations, middle approximations were better than upper approximations for one combination and upper approximations were better than lower approximations for one combination, for remaining 20 combinations the difference in performance was statistically insignificant. For "do not care" conditions, for four combinations middle approximations were better than lower approximations, for other four combinations upper approximations were better than lower approximations, for remaining 16 combinations the difference in performance was statistically insignificant. Thus, there is some evidence that for "do not care" conditions, the lower approximations should not be used for data mining.

It is not surprising since for data sets with large number of "do not care" conditions, the lower approximations are frequently empty, with the corresponding error rate equal to 100%.

Figure 13

Error rates for two series of data sets originated from the *Ionosphere* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"

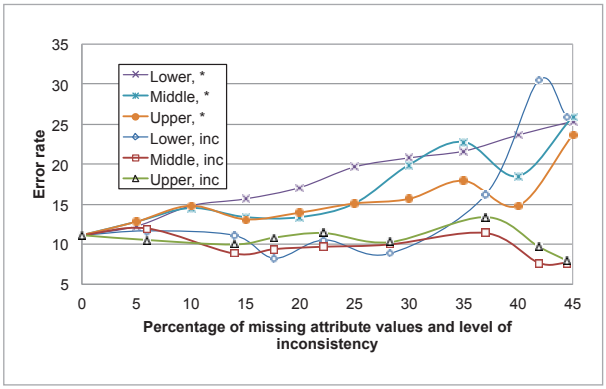


Figure 14

Error rates for two series of data sets originated from the *Iris* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"

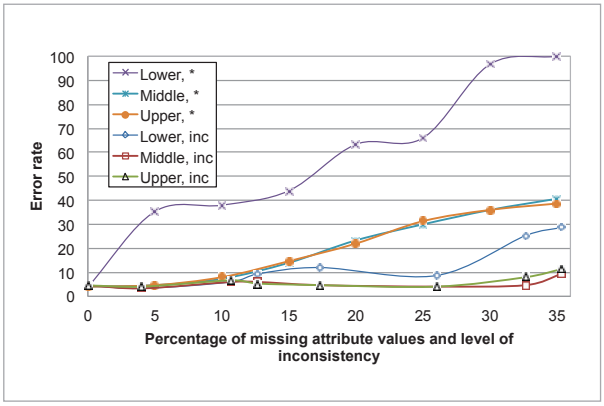
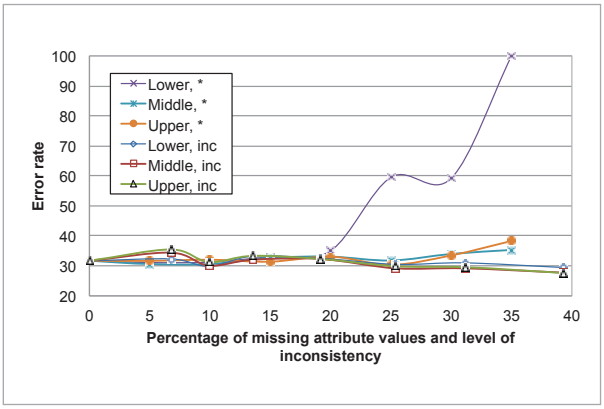


Figure 15

Error rates for two series of data sets originated from the *Pima* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"

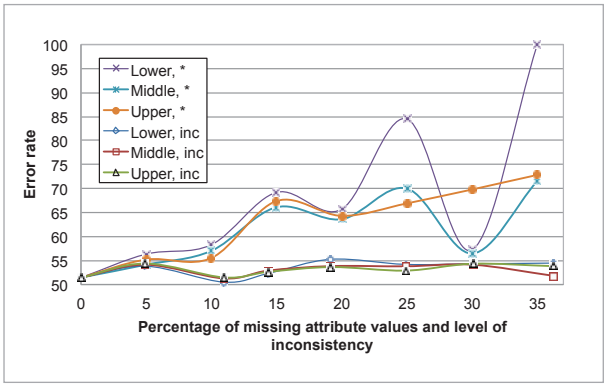


Conclusions

Our main conclusion is that there is experimental evidence that incompleteness is worse than inconsistency for data mining, in terms of an error rate. Additionally, lost values are better than "do not care" conditions, and there is a little difference between the three types of probabilistic approximations, except that for data sets with "do not care" conditions, lower approximations should not be used.

Figure 16

Error rates for two series of data sets originated from the *Yeast* data set. "Do not care" conditions are denoted by "*", inconsistent data are denoted by "inc"



References

1. Chmielewski, M. R., Grzymala-Busse, J. W. Global discretization of continuous attributes as preprocessing for machine learning. International Journal of Approximate Reasoning, 1996, 15(4), 319-331.https://doi.org/10.1016/S0888-613X(96)00074-6
2. Clark, P. G., Gao, C., Grzymala-Busse, J. W. A comparison of mining incomplete and inconsistent data. In: Proceedings of ICIST 2016 International Conference on Information and Software Technologies, 2016, 414-425. https://doi.org/10.1007/978-3-319-46254-7_33
3. Clark, P. G., Grzymala-Busse, J. W. Experiments on probabilistic approximations. In: Proceedings of the 2011 IEEE International Conference on Granular Computing, 2011, 144-149. https://doi.org/10.1109/GRC.2011.6122583
4. Grzymala-Busse, J. W. A new version of the rule induction system LERS. Fundamenta Informaticae, 1997, 31, 27-39.
5. Grzymala-Busse, J. W. Rough set strategies to data with missing attribute values. In: Notes of the Workshop on Foundations and New Directions of Data Mining, in conjunction with the Third International Conference on Data Mining, 2003, 56-63.
6. Grzymala-Busse, J. W. Data with missing attribute values: Generalization of indiscernibility relation and rule induction. Transactions on Rough Sets, 2004, 1, 78-95. https://doi.org/10.1007/978-3-540-27794-1_3
7. Grzymala-Busse, J. W. Generalized parameterized approximations. In: Proceedings of the 6-th International Conference on Rough Sets and Knowledge Technology, 2011, 136-145. https://doi.org/10.1007/978-3-642-24425-4_20
8. Grzymala-Busse, J. W., Rzasa, W. Definability and other properties of approximations for generalized indiscernibility relations. Transactions on Rough Sets, 2010, 11, 14-39. https://doi.org/10.1007/978-3-642-11479-3_2
9. Grzymala-Busse, J. W., Wang, A. Y. Modified algorithms LEM1 and LEM2 for rule induction from data with missing attribute values. In: Proceedings of the 5th International Workshop on Rough Sets and Soft Computing in conjunction with the Third Joint Conference on Information Sciences, 1997, 69-72.

10. Pawlak, Z. Rough sets. International Journal of Computer and Information Sciences, 1982, 11, 341-356. https://doi.org/10.1007/BF01001956
11. Pawlak, Z. Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, Boston, London, 1991.https://doi.org/10.1007/978-94-011-3534-4
12. Pawlak, Z., Skowron, A. Rough sets: Some extensions. Information Sciences, 2007, 177, 28-40. https://doi.org/10.1016/j.ins.2006.06.006
13. Pawlak, Z., Wong, S. K. M., Ziarko, W. Rough sets: Probabilistic versus deterministic approach. International Journal of Man-Machine Studies, 1988, 29, 81-95. https://doi.org/10.1016/S0020-7373(88)80032-4
14. Stefanowski, J., Tsoukias, A. Incomplete information tables and rough classification. Computational Intelligence, 2001, 17(3), 545-566. https://doi.org/10.1111/0824-7935.00162
15. Yao, Y. Y. Probabilistic rough set approximations. International Journal of Approximate Reasoning, 2008, 49, 255-271. https://doi.org/10.1016/j.ijar.2007.05.019
16. Yao, Y. Y., Wong, S. K. M. A decision theoretic framework for approximate concepts. International Journal of Man-Machine Studies, 1992, 37, 793-809. https://doi.org/10.1016/0020-7373(92)90069-W
17. Ziarko, W. Probabilistic approach to rough sets. International Journal of Approximate Reasoning, 2008, 49, 272-284. https://doi.org/10.1016/j.ijar.2007.06.014

Summary / Santrauka

We present experimental results on a comparison of incompleteness and inconsistency. We used two interpretations of missing attribute values: lost values and "do not care" conditions. Our experiments were conducted on 204 data sets, including 71 data sets with lost values, 71 data sets with "do not care" conditions and 62 inconsistent data sets, created from eight original numerical data sets. We used the Modified Learning from Examples Module version 2 (MLEM2) rule induction algorithm for data mining, combined with three types of probabilistic approximations: lower, middle and upper. We used an error rate, computed by ten-fold cross validation, as the criterion of quality. There is experimental evidence that incompleteness is worse than inconsistency for data mining (two-tailed test, 5% level of significance). Additionally, lost values are better than "do not care" conditions, again, with regards to the error rate, and there is a little difference in an error rate between three types of probabilistic approximations.

Straipsnyje pateikiami neužbaigtumo ir nenuoseklumo palyginimo eksperimento rezultatai. Naudotos dvi trūkstamų požymių įverčių interpretacijos: prarastos vertės ir „nesvarbu" sąlygos. Eksperimentui atlikti naudoti 204 duomenų rinkiniai, iš kurių 71 – su prarastomis vertėmis, 71 – su „nesvarbu" sąlygomis ir likę 62 nenuoseklūs duomenų rinkiniai, sukurti iš aštuonių originalių kiekybinių duomenų rinkinių. Duomenų gavybai naudotas pakeistas mokymasis (angl. *Modified Learning*) iš pavyzdžių modulio 2 versijos MLEM2 (angl. *Examples Module version*) taisyklių indukcijos algoritmo kartu su trijų tipų tikimybiniais priartėjimais: žemesniuoju, vidutiniu ir aukštesniuoju. Naudotas klaidų lygis, kuris apskaičiuotas kaip kokybės kriterijumi remiantis dešimties sluoksnių kryžmine validacija. Eksperimento rezultatai rodo, kad duomenims gauti neužbaigtumas blogiau nei nenuoseklumas (atliktas testas, gautas 5 % statistinio reikšmingumo lygmuo). Atsižvelgiant į klaidų lygmenį, prarastos vertės yra geriau nei „nesvarbu" sąlygos. Rezultatai taip pat rodo, kad nėra didelio skirtumo tarp trijų tikimybinių priartėjimų tipų klaidų lygmenų.