Proc. of the ISMIS-91, 6th International Symposium on Methodologies for Intelligent Systems, Charlotte, North Carolina, October 16–19, 1991, 368–377, *Lecture Notes in Artificial Intelligence*, vol. 542, 1991, Springer-Verlag

On the Unknown Attribute Values in Learning from Examples

Jerzy W. Grzymala-Busse
Department of Computer Science
University of Kansas
Lawrence, KS 66045

Abstract. In machine learning many real-life applications data are characterized by attributes with unknown values. This paper shows that the existing approaches to learning from such examples are not sufficient. A new method is suggested, which transforms the original decision table with unknown values into a new decision table in which every attribute value is known. Such a new table, in general, is inconsistent. This problem is solved by a technique of learning from inconsistent examples, based on rough set theory. Thus, two sets of rules: certain and possible are induced. Certain rules are categorical, while possible rules are supported by existing data, although conflicting data may exist as well. The presented approach may be combined with any other approach to uncertainty when processing of possible rules is concerned.

1. Introduction

In this paper it is assumed that input data for machine learning are stored in a *decision table*, in which *attributes* characterize *examples*. The decision table provides information about real world phenomena. Each example is described by *values* of attributes. Also, each example belongs to some *class*, also called a *concept*. Such a class is represented by a set of all examples having the same value of a variable *decision*. In many real-life applications an attribute may have unknown value for an example. More specifically, such value may exist, but is unknown. For example, the value has been not recorded, mistakenly erased, or forgotten by an expert. In the theory of databases such value is called a *null* [9].

Learning from examples is one of the most explored areas of machine learning. Until recently, many algorithms of learning from examples were developed assuming that input information is complete and free from errors or conflicts. As it was observed a few years ago [5], “very little attention has been paid to the problem of developing methods that work well in noisy environments. There is need for research on methods of learning from uncertain input information, from incomplete information, and from information containing errors.” The situation has improved greatly since that time. Many methods of machine learning under uncertainty have been invented, most of them based on probability theory. However, surprisingly little research has been done in the area of learning from incomplete information. Up-to-date methods to deal with unknown attribute values in learning from examples were presented in [15], see also [2, 10, 13, 14, 16]. They are based on the following ideas:

(1) ignoring examples with unknown values of attributes [13],

(2) assuming additional special value for an unknown value of attributes,

(3) using probability theory. For example, using relative frequencies of known values of a given attribute *A* for assigning them to unknown values [8]. Another possibility is based on replacing unknown values by the most common value of *A*, as in CN2 [4]. Yet another possibility is based on inclusion of an example with unknown value of attribute *A* to all subsets for which values of *A* are known [6],

(4) A. Shapiro suggested in a private communication to J. R. Quinlan a method in which an attribute with unknown values is assumed to be a decision and vice versa. Unknown values are determined as values of the new decision from known attributes and the old decision.

All of the above approaches have serious drawbacks. Approach (1), based on ignoring examples with unknown values, induces rules that may not cover all cases, or even worse, false rules, as may be showed by the following simple example.

Table 1

	Attributes			Decision
	Feel	Cuddliness	Material	Attitude
1	soft	–	plastic	negative
2	hard	–	plastic	positive
3	soft	furry	wool	neutral
4	hard	furry	wool	negative

Suppose that the attribute *Cuddliness* from Table 1 may assume values *smooth*, *furry*, and *fuzzy*. After ignoring the first two examples, as containing unknown values denoted '–' of attribute *Cuddliness*, two rules may be induced

$$(\text{Feel, soft}) \rightarrow (\text{Attitude, neutral}),$$

$$(\text{Feel, hard}) \rightarrow (\text{Attitude, negative}).$$

The above rules may be verified using the examples 1 and 2. Both rules should be rejected, since they are false. Thus, the approach (1) does not provide acceptable rules.

Using the approach (2), in which the unknown value of an attribute is considered an additional value, the following rules may be induced from Table 1:

$$(\text{Feel, hard}) \wedge (\text{Cuddliness, –}) \rightarrow (\text{Attitude, positive}),$$

$$(\text{Feel, hard}) \wedge (\text{Cuddliness, furry}) \rightarrow (\text{Attitude, negative}).$$

These rules have the following interpretation: if *Feel* is *hard* and it is not known what is the value of *Cuddliness* (but the value of *Cuddliness* is one of the three: *smooth*, *furry*, or *fuzzy*) then *Attitude* is *positive*, and if *Feel* is *hard* and *Cuddliness* is *furry* then *Attitude* is *negative*. Obviously, these two rules are conflicting, i.e., the approach (2) does not provide acceptable rules either.

It is difficult to use any probabilistic approach to Table 1 since the table is so small. In any case, it is clear that all probabilistic approaches, like the two preceding approaches, inevitably produce errors [13–16].

In order to use approach (4), only one attribute may have unknown values—a serious restriction. Moreover, even then, it is not always possible to use this approach. For example, using this method for Table 1, Table 2 must be created.

Table 2

	Attributes			Decision
	Feel	Attitude	Material	Cuddliness
1	soft	negative	plastic	–
2	hard	positive	plastic	–
3	soft	neutral	wool	furry
4	hard	negative	wool	furry

Approach (4) is useless in this case, since there is no way to guess what are values of *Cuddliness* from Table 2.

2. A new approach to unknown attribute values

The suggested here method presents the most cautious approach to unknown attribute value problem. The main idea of the method is to replace each example with an unknown value of attribute *A* by the set of examples, in which attribute *A* has its every possible value. Thus, if attribute *A* has an unknown value for example *E*, and attribute *A* has *m* possible values, then *E* will be replaced by *m* new examples E', E'',..., $E^{(m)}$. When example *E* has two unknown values of attributes *A* and *B*, and there is *m* possible values of *A* and *n* possible values of *B*, then *E* will be replaced by $m \cdot n$ examples, and so on. The most obvious rationale of the method is the following: since the value of an attribute *A* for a given example *E* is unknown, every possible value of *A* is considered, and every such value corresponds to a new example. On the other hand, the fact that attribute *A* has an unknown value for example *E*, and that *E* is a member of some class *C* may be interpreted in yet another way: an expert classified *E* as a member of class *C* not knowing the value of *A* , i.e., that such a value was not necessary for classification. This implies that it does not matter what a value it was, hence, *A* may assume any value from its domain.

Using this method, a consistent decision table may be converted into inconsistent one. The decision table is inconsistent when it contains at least one pair of inconsistent examples, i.e., examples characterized by the same values of all attributes yet with different values of a decision.

Table 3

	Attributes			Decision
	Feel	Cuddliness	Material	Attitude
1’	soft	smooth	plastic	negative
1’’	soft	furry	plastic	negative
1’’’	soft	fuzzy	plastic	negative
2’	hard	smooth	plastic	positive
2’’	hard	furry	plastic	positive
2’’’	hard	fuzzy	plastic	positive
3	soft	furry	wool	neutral
4	hard	furry	wool	negative

Table 3 was created by applying the method to Table 1. It is not difficult to see that Table 3 is consistent. Moreover, the following rules may be induced from Table 3:

(Feel, soft) ∧ (Material, plastic) → (Attitude, negative),
(Feel, hard) ∧ (Material, wool) → (Attitude, negative),
(Feel, soft) ∧ (Material, wool) → (Attitude, neutral),
(Feel, hard) ∧ (Material, plastic) → (Attitude, positive).

These rules cover all four original examples from Table 1, even though *Cuddliness* has two unknown values. Note that these rules are absolutely correct—no error analysis is required. Also, note that the attitude *Cuddliness* from Table 1 is irrelevant for inducing rules. This fact may be easily recognized from Table 3, from which rules are actually induced.

Table 4

	Attributes		Decision
	Color	Size	Attitude
1	blue	–	negative
2	–	big	negative
3	red	big	positive
4	red	–	positive

The next example, more general, is presented in Table 4. In this table it is assumed that attribute *Color* has three values: *blue*, *red*, and *yellow*, and that attribute *Size* has two values:

small and *big*. The new table, in which every example with unknown values of attribute *A* is replaced by the set of examples such that attribute *A* has its every possible value is presented in Table 5.

Table 5

	Attributes		Decision
	Color	Size	Attitude
1’	blue	small	negative
1’’	blue	big	negative
2’	blue	big	negative
2’’	red	big	negative
2’’’	yellow	big	negative
3	red	big	positive
4’	red	small	positive
4’’	red	big	positive

In Table 5, pairs of examples (2’’, 3) and (2’’, 4’’) are inconsistent (they are described by the same values of both attributes, yet corresponding values of decision for example 2’’ are different than these for examples 3 and 4’’). Thus Table 5 is inconsistent. An approach for learning rules from inconsistent tables, presented in the next section, follows ideas from [7].

3. Rough Set Approach for Inconsistent Examples

In the early eighties Z. Pawlak introduced a new tool to deal with uncertainty, called rough set theory [12]. The main advantage of rough set theory is that it does not need any preliminary or additional information about data (like prior probability in probability theory, basic probability number in Dempster-Shafer theory, grade of membership or value of possibility in fuzzy set theory). Other advantages of the rough set approach include its ease of handling and its simple algorithms.

Rough set theory is especially well suited to deal with inconsistencies in the process of machine learning. In the presented approach, inconsistencies are not corrected. The key issue is to compute lower and upper approximations of concepts, the fundamental notions of rough set theory. On the basis of lower and upper approximations, two different sets of rules are computed: certain and possible. Certain rules are categorical and may be further employed using classical logic. Possible rules are supported by existing data, although conflicting data may exist as well. Possible rules may be processed further using either classical logic or any theory to deal with uncertainty [3, 7]. There exist other methods of machine learning using rough set theory, see e.g. [1, 17].

Note that the presented approach may be combined with any other approach to uncertainty when processing of possible rules is concerned. An advantage of the method is that certain and possible rules are processed separately, i.e. two parallel inference engines may be used.

Let U be a nonempty set, called the *universe*, and let R be an equivalence relation on U, called an *indiscernibility relation*. An ordered pair (U, R) is called an *approximation space*. For any element x of U, the equivalence class of R containing x will be denoted by $[x]_R$. Equivalence classes of R are called *elementary sets in (U, R)*. We assume that the empty set is also elementary.

Any finite union of elementary sets in (U, R) is called a *definable set in (U, R).*

Let X be a subset of U. We wish to define X in terms of definable sets in (U, R). Thus, we need two more concepts.

A *lower approximation of X in (U, R),* denoted by $\underline{R}X$, is the set

$$\{x \in U \mid [x]_R \subseteq X \}.$$

An *upper approximation of X in (U, R),* denoted by $\overline{R}X$, is the set

$$\{x \in U \mid [x]_R \cap X \neq \emptyset \}.$$

The lower approximation of X in (U, R) is the greatest definable set in (U, R), contained in X. The upper approximation of X in (U, R) is the least definable set in (U, R) containing X. Time complexity of algorithms for computing lower and upper approximations of any set X is $O(n^2)$, where n is the cardinality of set U of examples. A *rough set in (U, R)* is the family of all subsets of U having the same lower and upper approximations in (U, R).

Let x be in U. We say that x is *certainly in X* iff x $\in \underline{R}X$, and that x is *possibly in X* iff $x \in \overline{R}X$. Our terminology originates from the fact that we want to decide if x is in X on the basis of a definable set in (U, R) rather than on the basis of X . This means that we deal with $\underline{R}X$ and $\overline{R}X$ instead of X. Since $\underline{R}X \subseteq X \subseteq \overline{R}X$, if x is in $\underline{R}X$ it is certainly in X . On the other hand, if x is in $\overline{R}X$, it is possibly in X .

With any decision table an approximation space may be associated. Let Q be a set of all attributes and let U be a set of all examples of the decision table. For any nonempty subset P of Q, an ordered pair $(U, \underset{P}{\sim})$ is an approximation space (U, R), where $\underset{P}{\sim}$ is an indiscernibility relation on U, defined as follows. For $x, y \in U$, $x \underset{P}{\sim} y$ if and only if x and y have the same value on all attributes in P. The indiscernibility relation, associated with P, is an equivalence relation on U. As such, it induces a *partition* of U, generated by P, denoted P^*.

For the sake of convenience, for any $X \subseteq U$, the lower approximation of X in (U, R) and the upper approximation of X in (U, R) are called *P-lower approximation of X* and *P-upper approximation of X,* and are denoted $\underline{P}X$ and $\overline{P}X$, respectively.

Measures of uncertainty based on rough set theory have auxiliary value only, since in the rough-set approach, the set X is described by its lower and upper approximations. One of such measures is a *quality of lower approximation of X by P*. It is equal to

$$\frac{|\underline{P}X|}{|U|} .$$

Thus, the quality of lower approximation of X by P is the ratio of the number of all examples certainly classified by attributes from P as being in concept X to the number of all examples. It is a kind of relative frequency. Note that the quality of lower approximation of X by P is a belief function according to Dempster-Shafer theory.

A quality of upper approximation of X by P is equal to

$$\frac{|\overline{P}X|}{|U|} .$$

The quality of upper approximation of X by P is the ratio of the number of all possibly classified objects by attributes from P as being in X to the number of all objects of the system. Therefore, it is again a kind of relative frequency. The quality of upper approximation of X by P is a plausibility function from the Dempster-Shafer theory viewpoint.

The difference between $\frac{|\overline{P}X|}{|U|}$ and $\frac{|\underline{P}X|}{|U|}$ is called an *error* ε. It is the ratio of the number of all examples, possibly but not certainly properly classified by attributes from P as being in X, to the number of all examples. Thus defined error ε is an estimate of the worst case of actual error of classification. The actual error is always smaller than ε.

4. Certain and Possible Rules

The main idea of use of rough set theory for learning from examples is presented in Figure 1. A set of examples is given, e.g. in the form of decision table. All results of uncertainty are manifested finally by inconsistent information in the decision table.

For any concept X, every block of $\underline{P}X$ or $\overline{P}X$ is definable, hence it may be represented by rules using attributes of set R. Rules induced on the basis of the lower approximation $\underline{P}X$ are *certain*. *Possible* rules, on the other hand, are induced on the basis of upper approximation $\overline{P}X$.

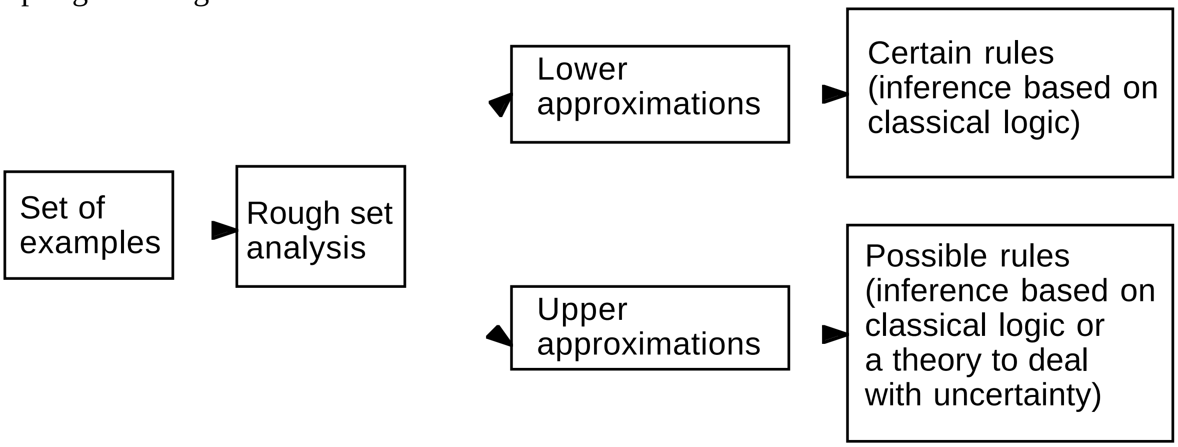
Lower approximations
Certain rules (inference based on classical logic)
Set of examples
Rough set analysis
Upper approximations
Possible rules (inference based on classical logic or a theory to deal with uncertainty)

Figure 1. Principle of use of rough set theory for learning from examples

For Table 5, the partition *P** is equal to

{{1’}, {1”, 2’}, {2”, 3, 4”}, {2”, 3, 4”}, {2’”}, {4’}},

where $P = \{Color, Size\}$. The lower approximation of class {1’, 1’’, 2’, 2’’, 2’’’}, corresponding to value *negative* of *Attitude* is {1’, 1’’, 2’, 2’’’}. Similarly, the lower approximation of the class {3, 4’, 4’’} is {4’}.

The upper approximation of the class {1’, 1’’, 2’, 2’’, 2’’’} is {1’, 1’’, 2’, 2’’, 2’’’, 3, 4’’}, and the upper approximation of the class {3, 4’, 4’’} is {2’’, 3, 4’, 4’’}. Thus, for both classes, error ε is the same and equal to

$$\frac{7 - 4}{8} = \frac{4 - 1}{8} = 0.375.$$

The expression for an error may be interpreted as follows: three examples (2’’, 3, and 4’’) out of eight are possibly but not certainly correctly classified. The error would be 0.375 when **all** three examples: 2’’, 3, and 4’’ are mistakenly classified, i.e., when none of them belong to the corresponding class (however, that is impossible).

Lower approximation of the classes imply Tables 6 and 7.

Table 6

	Attributes		Decision
	Color	Size	Attitude
1’	blue	small	negative
1’’	blue	big	negative
2’	blue	big	negative
2’’	red	big	positive
2’’’	yellow	big	negative
3	red	big	positive
4’	red	small	positive
4’’	red	big	positive

Table 7

	Attributes		Decision
	Color	Size	Attitude
1’	blue	small	negative
1’’	blue	big	negative
2’	blue	big	negative
2’’	red	big	negative
2’’’	yellow	big	negative
3	red	big	negative
4’	red	small	positive
4’’	red	big	negative

Note that Tables 6 and 7 are consistent. Thus, from Table 6, rules describing the class {1’, 1’’, 2’, 2’’’}, i.e. certain rules for value *negative* of *Attitude*, may be induced as follows:

(Color, blue) → (Attitude, negative),

(Color, yellow) → (Attitude, negative).

Similarly, from Table 7, the rule for the class {4’}, i.e. a certain rule for value *positive* of *Attitude*, is induced:

(Color, red) ∧ (Size, small) → (Attitude, positive).

Upper approximations of the classes imply Tables 7 and 6 (these tables are the same as implied by lower approximations because *Attitude* has two values). From Table 7, rules for the class {1’, 1’’, 2’, 2’’, 2’’’, 3, 4’’}, i.e., possible rules for value *negative* of *Attitude* are induced:

(Color, blue) → (Attitude, negative),

(Size, big) → (Attitude, negative),

(Color, yellow) → (Attitude, negative).

Finally, from Table 6, a rule for the class {2'', 3, 4', 4''}, i.e., a possible rule for value *positive* of *Attitude* is induced:

(Color, red) → (Attitude, positive).

The certain rules, listed above, are absolutely correct—no error analysis is required. The error for the possible rules is always smaller than 37.5%. The above rules, certain and possible, are presented in the minimal discriminant form [11].

Also, note that using an approach based on ignoring examples with unknown values of attributes to Table 3 will produce a new table with just one example, from which very little can be learned. For Table 3 the remaining approaches, listed in [15], are also of a very little help.

4. Conclusions

A very simple method is proposed to deal with unknown values of attributes: every example with unknown values of attribute *A* is replaced by the set of examples having every possible value for *A*. This is the most conservative approach because an unknown value is replaced by every possible value. This method produces, in general, inconsistent decision tables. However, the problem of learning rules from inconsistent examples may be easily solved using rough set theory. Thus, two different sets of rules are computed: certain and possible. Certain rules are categorical and may be further employed using classical logic. Possible rules are supported by existing data, although conflicting data may exist as well. For possible rules an estimate for the worst case of error is presented.

Certain and possible rules may be propagated separately during an inference process in an expert system, producing thus new certain and possible rules, respectively. Therefore, the inference engine of an expert system may be divided into two parallel subsystems, for certain and possible rules, in which certain and possible rules are processed separately. Both subsystems will operate in the same way as those based on classical logic. For example, standard strategies, like forward and backward chaining, are then applicable.

Proc. of the ISMIS-91, 6th International Symposium on Methodologies for Intelligent Systems, Charlotte, North Carolina, October 16–19, 1991, 368–377, *Lecture Notes in Artificial Intelligence*, vol. 542, 1991, Springer-Verlag.

References

[1] T. Arciszewski, M. Mustafa, and W. Ziarko. A methodology of design knowledge acquisition for use in learning expert systems. *Int. J. Man-Machine Studies* 27, 1987, 23–32.

[2] J. Catlett. Induction using the Shafer representation. *Technical Report. Basser Department of Computer Science, University of Sydney, Australia*, 1985.

[3] C.-C. Chan and J. W. Grzymala-Busse. Rough-set boundaries as a tool for learning rules from examples. *Proc. ISMIS-89, 4th Int. Symposium on Methodologies for Intelligent Systems*, 1989, 281–288.

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

[5] T. G. Dietterich, R. S. Michalski. A comparative review of selected methods for learning from examples. In *Machine Learning. An Artificial Intelligence Approach*, ed. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, Morgan Kauffman, 1983, 41–81.

[6] J. H. Friedman. A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, 1977, 404–408.

[7] J. W. Grzymala-Busse. Knowledge acquisition under uncertainty—a rough set approach. *Journal of Intelligent & Robotic Systems* 1, 1988, 3–16.

[8] I. Knonenko, I. Bratko, E. Roskar. Experiments in automatic learning of medical diagnostic rules. *Technical Report, Jozef Stefan Institute, Ljubljana, Yugoslavia*, 1984.

[9] D. Maier. *The Theory of Relational Databases*, Computer Science Press, 1983.

[10] M. V. Manago, Y. Kodratoff. Noise and knowledge acquisition. *Proc. IJCAI* 1987, 348–354.

[11] R. S. Michalski. A theory and methodology of inductive learning. In *Machine Learning. An Artificial Intelligence Approach*, ed. R. S. Michalski, J. G. Carbonell, T. M. Mitchell, Morgan Kauffman, 1983, 83–134.

[12] Z. Pawlak. Rough sets. *Int. J. Computer and Information Sci.* 11, 1982, 341–356.

[13] J. R. Quinlan. Induction of decision trees. *Machine Learning* 1, 1986, 81–106.

[14] J. R. Quinlan. Decision trees as probabilistic classifiers. *Proc. 4th Int. Workshop on Machine Learning* 1987, 31–37.

[15] J. R. Quinlan. Unknown attribute values in induction. *Proc. 6th Int. Workshop on Machine Learning*, 1989, 164–168.

[16] J. R. Quinlan. Probabilistic decision trees. In *Machine Learning. An Artificial Intelligence Approach*, vol III, ed. Y. Kodratoff and R. S. Michalski, 1990, 140–152.

[17] R. Yasdi and W. Ziarko. An expert system for conceptual schema design: A machine learning approach. *Int. J. Man-Machine Studies* 29, 1988, 351–376.