Computational Intelligence 11 (1995), 268–279.

ON THE USEFULNESS OF MACHINE LEARNING APPROACH TO KNOWLEDGE ACQUISITION

DOBROSLAWA M. GRZYMALA-BUSSE AND JERZY W. GRZYMALA-BUSSE

Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045, USA

This paper presents results of experiments showing how machine learning methods are useful for rule induction in the process of knowledge acquisition for expert systems. Four machine learning methods were used: ID3, ID3 with dropping conditions, and two options of the system LERS: LEM1 and LEM2. Also, two knowledge acquisition options of LERS were used as well. All six methods were used for rule induction from six real-life data sets. The main objective was to test how an expert system, supplied with these rule sets, will perform without information on a few attributes. Thus an expert system attempts to classify examples with all missing values of some attributes. As a result of experiments it is clear that all machine learning methods performed much worse than knowledge acquisition options of LERS. Thus, machine learning methods used for knowledge acquisition should be replaced by other methods of rule induction that will generate complete sets of rules. Knowledge acquisition options of LERS are examples of such appropriate ways of inducing rules for building knowledge bases.

Key-words: Knowledge acquisition, learning from examples, rule-based expert systems, incomplete knowledge.

INTRODUCTION

The main module of a rule-based expert system is a rule base, also called knowledge base. The rule base contains specific knowledge about the problem area presented in rules. Rules, in the form **if then** are elementary units of knowledge. The process of constructing the knowledge base is called knowledge acquisition. This is one of the most important and at the same time most difficult steps of building an expert system (Addis *et al*. 1991; Buchanan 1989; Grzymala-Busse 1991; Weiss and Kulikowski 1990). Although books, journals, owner's manuals, data bases, etc. are valuable sources of knowledge, the most popular technique of knowledge acquisition is still an interaction with a human expert. A knowledge engineer, a person acquiring knowledge, interacts with an expert either by observation of the expert in action or by interview. As a result, rules are produced, first in plain English, later on in the coded form accepted by a computer. It is the responsibility of the knowledge engineer to acquire knowledge in such a way that the knowledge base is as complete as possible. The more complete the knowledge base the higher quality performance of the expert system. The quality of performance of an expert system may be measured in many different ways. In this study we will measure this quality by performance of an expert system working with incomplete information, when decision is produced by an expert system even though values of only some attributes are given.

In order to simplify interviewing, which is manual, expensive, potentially incorrect, and time-consuming, a variety of computerized methods have been developed (Boose 1989). Among them are interactive and learning-based techniques. The former is represented by techniques that help the knowledge engineer in building the knowledge base, while the latter is a fully automated method of building the knowledge base by *machine learning* (Addis *et al*. 1991; MacDonald and Witten 1989; Shalin *et al*. 1988). In recent years the mainstream of machine learning has been inductive learning. Inductive learning may be categorized as

learning from examples and *learning from observation.* Learning from examples is practically the only method of machine learning used in expert systems (Buchanan 1991). For the sake of simplicity, in the rest of the paper learning from examples will be called machine learning. Machine learning represents a specific method of building the knowledge base. It provides the knowledge base only with sufficient knowledge, while knowledge acquisition looks for a full spectrum of knowledge. In terms of rule-based systems, it means that machine learning is focused on inducing a sufficient number of rules, while the knowledge acquisition approach to building the rule base should induce all potential rules that can be induced. Hence, we distinguish the machine learning approach and the knowledge acquisition approach in rule induction. There exist machine learning systems that induce some redundant rules (Gams *et al*. 1991), but these systems still represent the machine learning approach.

Although it seems obvious that expert systems using rule sets induced by machine learning methods should not perform well because many potential rules that could be induced from the original data are missing, so far no comparative research has been done to document that machine learning methods used for knowledge acquisition are far from perfect. Research in this direction was the main objective of this work.

In this paper two versions of ID3 and two options of LERS, LEM1 and LEM2, were used as examples of machine learning methods. The choice of a machine learning system is irrelevant to the objective of this research since any machine learning method induces a rule set that is a subset of all potential rules hidden in the original data set. The selection of the rule set by a machine learning method depends on the specific bias of the method. Two other options of LERS were used as examples of knowledge acquisition methods. All six methods were used for rule induction from six real-life data sets. Thus 36 rule sets were induced. The main objective was to test how all six methods will perform when an expert system operates without information on any one, two, or three attributes. Thus entire attributes (one, two, or three) were missing, i.e., their values were missing for all examples. Each of these 36 rule sets were incorporated into an expert system. Then the expert system attempted to classify all examples from all data sets; however, in the first running, for the first data set, the first attribute was missing. In general, some examples were classified correctly, some were not classified, although no example was classified incorrectly. The number of not classified examples was recorded. On that basis, the error rate, i.e., the ratio of all errors (not classified examples) to the total number of examples, was computed. Then the experiment was repeated, for the same rule set and the same data set, only this time the second attribute was missing. Such experiments were done for every single missing attribute and the average error rate was computed. Then, for the same rule set and the same data set, the next sequence of experiments was done, for two arbitrary missing attributes, and the average error rate was computed. In general, for three missing attributes, the length of a sequence of experiments for every rule set and data set was restricted to 600, and the average error rate was also computed. From the analysis of error rate it is clear that all machine learning methods performed much worse than knowledge acquisition options of LERS. Thus, machine learning methods used so far for knowledge acquisition should be replaced by other methods of rule induction that will generate complete sets of rules.

A preliminary and abbreviated version of this paper, restricted only to the system LERS, was presented as (Grzymala-Busse and Grzymala-Busse 1993).

1. PRELIMINARY DEFINITIONS

The fundamental goal of learning from examples lies in inducing rules or decision trees from examples. The set of examples is presented by a *decision table*, exemplified by Table 1. Each example in the decision table is characterized by *attributes* and a *decision*. The following parameters are attributes: Age, #pregnancies, %body-fat, Cholesterol, while Breast-cancer is the decision.

TABLE 1. Decision table

	Age	#pregnancies	%body-fat	Cholesterol	Breast-cancer
1	29..41	1..4	18..28	188..197	no
2	42..56	1..4	18..28	198..320	no
3	42..56	0	29..37	198..320	yes
4	29..41	0	29..37	198..320	yes
5	57..64	1..4	18..28	198..320	no
6	42..56	1..4	18..28	188..197	yes
7	29..41	1..4	18..28	188..197	no
8	42..56	1..4	29..37	198..320	yes
9	57..64	1..4	29..37	198..320	yes
10	57..64	1..4	18..28	188..197	no

Let U be a set of all examples. If d denotes a decision and d_i denotes its value then a concept is a set of $[(d, d_i)]$ of all examples that have value d_i for decision d. For example, decision Breast-cancer from Table 1 has two values: no and yes. Each such value represents a concept. These two concepts are: {1, 2, 5, 7, 10} and {3, 4, 6, 8, 9}. Likewise the *block of an attribute-value pair* $t = [(a, a_j)]$, denoted $[t]$, is the set of all examples that for attribute a have value a_j (Grzymala-Busse 1992).

The examples affiliated with a concept are called *positive examples* while members of the complement of the concept are called *negative examples*. Let us consider the concept [(Breast-cancer, no)] = {1, 2, 5, 7, 10}. The set {1, 2, 5, 7, 10} is the set of positive examples while {3, 4, 6, 8, 9} is the set of all negative examples.

Let Q be the set of all attributes and a decision. In Table 1, Q = {Age, #pregnancies, %body-fat, Cholesterol, Breast-cancer}.

Let P be an arbitrary non empty subset of Q. Let x and y be arbitrary examples (elements of U). Examples x and y are called *indiscernible* by P, denoted $x \underset{P}{\sim} y$ if and only if x and y have the same value on all elements of P. This means that rows of the table labeled by x and y and restricted to columns labeled by elements of P have, pairwise, the same values (Grzymala-Busse 1991, 1992;

Pawlak 1982, 1991). Obviously $\underset{P}{\sim}$ is an equivalence relation and induces a partition of Q, denoted P^*. So, partition P^* is the set of all equivalence classes (*blocks*) of the indiscernibility relation, or simply classes of P^*. For example

$$\{\text{Age}\}^* = \{\{1, 4, 7\}, \{2, 3, 6, 8\}, \{5, 9, 10\}\}.$$

Let B be a subset of the set A of all attributes and let d be a decision. Set $\{d\}$ depends on set B, denoted $B \rightarrow \{d\}$, if and only if $B^* \leq \{d\}^*$ where B^* is the partition on U. Moreover, $B^* \leq \{d\}^*$ means that for each block X of B^* there exists a block Y of $\{d\}^*$ such $X \subseteq Y$.

In practice, the set $\{d\}$ depends on a subset B of the set A of all attributes if and only if the description of examples by attributes from B is sufficient to recognize the concept.

As a result of learning from examples, rules are obtained in the following form:

$$L \rightarrow R.$$

The left side L is a conjunction of conditions

$$C_1 \wedge C_2 \wedge \ldots \wedge C_n.$$

Conditions are attribute-value pairs (a, a_j), while the right side of the rule is the (d, d_i) pair. Rules must be complete and consistent. If all examples in the concept are described by rules then the set of rules is complete. The set of rules is consistent if no example is satisfied by rules describing two different concepts.

2. ID3 ALGORITHM

The ID3 algorithm (Quinlan 1982, 1983, 1986, 1987a, 1987b) is a successor of the CLS algorithm (Hunt *et al*. 1966). The main task of ID3 is constructing a decision tree. Nodes of the tree are labeled by attributes while arcs are labeled by values of an attribute. Our assumption is that the set U of examples is partitioned into at least two concepts. The attribute that is a label of the root is selected on the basis of the maximum of gain ratio criterion. Let a be an attribute with values a_1, a_2,..., a_l and let d be a decision with values d_1, d_2,..., d_k. Then the gain ratio is defined as follows

$$\frac{H(d) - H(d \mid a)}{H(a)},$$

where $H(a)$ is the entropy of attribute a,

$$H(a) = -\sum_{j=1}^{l} p(a_j) \cdot \log p(a_j),$$

where $H(d)$ is the entropy of decision d,

$$H(d) = -\sum_{i=1}^{k} p(d_i) \cdot \log p(d_i),$$

and $H(d \mid a)$ is the conditional entropy of decision d given attribute a,

$$H(d \mid a) = \sum_{j=1}^{l} p(a_j) \cdot H(d \mid a_j)$$

$$= -\sum_{j=1}^{l} p(a_j) \cdot \sum_{i=1}^{k} p(d_i \mid a_j) \cdot \log p(d_i \mid a_j) .$$

At this point, the corresponding decision tree, created by ID3, has the root, labeled by the attribute a and outgoing arches from the root, each such arch corresponds to a value a_j of the attribute a. The set of all examples with the same value a_j of attribute a consists of a new set S of examples. When all members of S are members of the same concept C, they do not need to be further partitioned, so S is a label of the leaf of the tree. However, when S contains examples from at least two different concepts, the node of the tree labeled by S is the root of a new subtree. An attribute to be a label for the root of this new subtree is selected among remaining attributes again on the basis of the maximum of gain ratio criterion. From the decision tree rules may be easily induced. For Table 1, ID3 induced the following rules:

(%body-fat, 29..37) -> (Breast-cancer, yes),

(%body-fat, 18..28) & (Age, 29..41) -> (Breast-cancer, no),

(%body-fat, 18..28) & (Age, 42..56) & (Cholesterol, 198..320) ->

(Breast-cancer, no),

(%body-fat, 18..28) & (Age, 42..56) & (Cholesterol, 188..197) ->

(Breast-cancer, yes),

(%body-fat, 18..28) & (Age, 57..64) -> (Breast-cancer, no).

The rules induced by ID3 may be further simplified using *linear* dropping conditions. For a rule of the form

$$C_1 \wedge C_2 \wedge \cdots \wedge C_l \rightarrow R$$

linear dropping conditions means scanning the list of all conditions, from left to right, with an attempt to drop any l conditions, checking against the decision table whether the simplified rule does not violate consistency of the discriminant description. In the rest of the paper, system ID3 with linear dropping conditions will be denoted by ID3/Drop. For Table 1, the system ID3/Drop induced the following rules

(%body-fat, 29..37) -> (Breast-cancer, yes),
(%body-fat, 18..28) & (Age, 29..41) -> (Breast-cancer, no),
(%body-fat, 18..28) & (Cholesterol, 198..320) -> (Breast-cancer, no),
(Age, 42..56) &(Cholesterol, 188..197) -> (Breast-cancer, yes),

(%body-fat, 18..28) & (Age, 57..64) -> (Breast-cancer, no).

3. THE LERS SYSTEM

The system LERS (Learning from Examples based on Rough Sets) consists of two options of machine learning from examples and two options of knowledge acquisition (Grzymala-Busse 1992).

Machine learning options are LEM1 and LEM2. They produce a sufficient set of rules to cover all examples in the decision table. Knowledge acquisition options are called All Global Coverings and All Rules. Both of them usually produce much bigger sets of rules from the input data given by a decision table.

LERS is able to deal with uncertainty in the input decision table as well. There are many reasons for uncertainty. The main cause of uncertainty is missing values of attributes or inconsistent examples. Missing attribute values are due to lack of information or result from the situation where we do not care what the attribute value is.

Rough set theory was introduced in the early 1980's (Pawlak 1982, 1991). It is especially useful for dealing with inconsistencies. This approach to uncertainty does not need any preliminary or additional information about data.

First LERS tests the input data for consistency. If data are inconsistent then lower and upper approximations of each concept are computed (Grzymala-Busse 1992). Now the user has an option to choose between two machine learning options and two knowledge acquisition options. If a machine learning option is used then the system induces a single minimal discriminant description for each concept. If a knowledge acquisition option is applied a complete set of rules is induced. In both cases local and global approaches may be chosen.

In the case of local options of LERS the system induces certain and possible rules from lower and upper approximation for each concept respectively.

In the case of global options of LERS new partitions on the set U are computed. Say that the original decision table described k concepts, i.e., decision has k values. Then $2k$ new partitions on U, called *substitutional partitions* are, created. Each substitutional partition has exactly two blocks, the first block is either lower or upper approximation of the concept, the second block is the complement of the first block. Substitutional partitions computed from lower approximations are called *lower substitutional partitions*; substitutional partitions computed from upper approximations are called *upper substitutional partitions.* Decisions, corresponding to lower and upper partitions, are called *lower* and *upper substitutional decisions*.

3.1. LEM1 (Single Global Covering Method)

A single global covering method using procedure LEM1 allows the system LERS to compute minimal discriminant description of the concept. This procedure employs the following ideas.

Let A denote the set of all attributes and let d denote a lower or upper substitutional decision. A *global covering* of $\{d\}$ is a subset P of A such that $\{d\}$ depends on P and P is minimal in A. The global covering is called a reduct in (Pawlak 1991). LEM1 computes first a global covering of $\{d\}$ and then computes rules from the global covering.

For the decision table presented in Table 1, induced rules are the same as for the algorithm ID3/Drop. The decision table from Table 1 is a small one and is presented only for illustration. Besides, for this decision table the only global

covering is {Age, %body-fat, Cholesterol}. All attributes from the global covering were selected by ID3 algorithm as well.

3.2. LEM2 (Single Local Covering Method)

The single local covering method, called LEM2, represents the machine learning approach, i.e., it produces minimal rules from examples. The rules generated are said to be minimal because they do not induce unnecessary conditions. The following ideas are employed for LEM2. Let B be a lower or upper approximation of a concept represented by a decision-value pair (d, w). Set B *depends on a set* T of attribute value pairs if and only if

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

Set T is a *minimal complex* of B if and only if B depends on T and no proper subset T' of T exists such that B depends on T'. The minimal complex is called a value reduct in (Pawlak 1991). Let $\mathbb{T}$ be a non empty collection of non empty sets of attribute-value pairs. Then $\mathbb{T}$ is a *local covering of* B if and only if the following conditions are satisfied:

(1) each member T of $\mathbb{T}$ is a minimal complex of B,

(2) $\bigcup_{T \in \mathbb{T}} [T] = B$, and

(3) $\mathbb{T}$ is minimal, i.e., $\mathbb{T}$ has the smallest possible number of members.

The algorithm LEM2 is based on computing a single local covering for each of the concepts from the decision table. The user may select a version of LEM2 with or without taking into account attribute priorities.

Again, since the decision table from Table 1 is so small, rules induced by LEM2 are the same as rules induced by ID3/Drop. However, if the user selects the following priorities for attributes: the first (highest) priority to #pregnancies, the second priority to Cholesterol, the third priority to Age, and the lowest priority to %body-fat, LEM2 induces the following rules:

(#pregnancies, 1..4) & (Age, 29..41) → (Breast-cancer, no),

(Cholesterol, 198..320) & (%body-fat, 18..28) → (Breast-cancer, no),

(Cholesterol, 188..197) & (Age, 57..64) → (Breast-cancer, no),

(%body-fat, 29..37) → (Breast-cancer, yes),

(Cholesterol, 188..197) & (Age, 42..56) → (Breast-cancer, yes).

3.3. All Global Coverings Method

All Global Coverings method represents the knowledge acquisition approach to rule induction. This means that the set $\mathbb{R}$ of all global coverings for every lower and upper substitutional partition of $\{d\}^*$ is discovered and next by using the exponential dropping condition, rules are induced.

In the exponential dropping condition algorithm any subset of the set of all conditions of the rule is checked for dropping conditions. For the decision table presented in Table 1 rules, induced by All Global Coverings option of the system LERS, are

(Age, 29..41) & (%body-fat, 18..28) -> (Breast-cancer, no),
(Age, 57..64) & (%body-fat,18..28) -> (Breast-cancer, no),
(Age, 29..41) & (Cholesterol, 188..197) -> (Breast-cancer, no),
(Age, 57..64) & (Cholesterol, 188..197) -> (Breast-cancer, no),
(%body-fat, 18..28) & (Cholesterol, 198..320) -> (Breast-cancer, no),
(%body-fat, 29..37) -> (Breast-cancer, yes),
(Age, 29..41) & (Cholesterol, 198..320) -> (Breast-cancer, yes),
(Age, 42..56) & (Cholesterol, 188..197) -> (Breast-cancer, yes).

3.4. All Rules Method

All Rules method is another knowledge acquisition approach to rule induction. All rules that can be induced from input data file (decision table) are actually induced. The algorithm is very simple but very demanding from the viewpoint of time complexity. For every concept, represented by a decision-value pair, the lower and upper approximations are computed. Then rules are induced directly from the decision tables, where the original decision is replaced by lower and upper substitutional decisions. Thus, for each rule, the number of conditions is equal to the number of attributes. But final form of induced rules is minimal due to exponential dropping condition process that is used as the last step of the algorithm. For the decision table presented in Table 1, the set of all rules, induced by the All Rules option of the system LERS, is presented below.

(Age, 29..41) & (#pregnancies, 1..4) -> (Breast-cancer, no),
(Age, 29..41) & (%body-fat, 18..28) -> (Breast-cancer, no),
(Age, 57..64) & (%body-fat, 18..28) -> (Breast-cancer, no),
(Age, 29..41) & (Cholesterol, 188..197) -> (Breast-cancer, no),
(Age, 57..64) & (Cholesterol, 188..197) -> (Breast-cancer, no),
(%body-fat, 18..28) & (Cholesterol, 198..320) -> (Breast-cancer, no),
(#pregnancies, 0) -> (Breast-cancer, yes),
(%body-fat,29..37) -> (Breast-cancer, yes),
(Age, 29..41) & (Cholesterol, 198..320) -> (Breast-cancer, yes),
(Age, 42..56) & (Cholesterol, 188..197) -> (Breast-cancer, yes).

4. EXPERIMENTS

Our experiments, done on the computers VAX 9000 under VMS and DEC 5000 under UNIX, were designed to check usefulness of rule sets induced by ID3, ID3/Drop, and all four options of LERS for applications in the expert system area.

Six real-life data sets were used for experiments; see Table 2. They varied widely in the number of examples, attributes, and concepts. There are differences

between original data sets and those used in experiments. First of all, examples with missing values were removed from all original data sets. Inconsistencies were resolved by keeping only dominant examples (i.e., those occurring a greater number of times) and, in the case of a tie, only the first example. Thus, data sets used in our experiments may be of smaller size than original ones.

TABLE 2. Data sets

Data set	Number of examples	Number of attributes	Number of concepts
BANK	66	5	2
BREAST	264	9	2
HOUSE	232	16	2
LYMPHOGRAPHY	148	18	4
SOYA	47	35	4
TUMOR	112	17	18

The data set BANK was created at the New York University School of Business. Data were collected in 1968 and present either bankrupt or non-bankrupt firms.

The data sets BREAST (breast cancer), LYMPHOGRAPHY, and TUMOR (primary tumor) were obtained from the Institute of Oncology of the University Medical Center at Lubljana, Slovenia. The data set LYMPHOGRAPHY originally was consistent and without missing values. The original data set BREAST had 286 examples with inconsistencies and missing attribute values. The original data set TUMOR had 339 examples with inconsistencies and missing attribute values.

The data set HOUSE (1984 United States Congressional Voting Records Database) originally had 435 examples without inconsistencies and with missing attribute values.

The original data set SOYA (also called *small soybean data*) had examples without inconsistencies and without missing attribute values. This data set was donated to the machine learning community by R. M. Michalski (Illinois University).

TABLE 3. Rule sets

Data set	ID3	ID3/drop	LEM1	LEM2 Coverings	All Rules	All
BANK	5	4	4	5	5	9
BREAST	123	106	149	76	573	1447
HOUSE	16	16	77	13	1258	2560
LYMPHOGRAPHY	42	39	77	26	5548	6794
SOYA	4	4	4	4	1078	–
TUMOR	68	65	79	57	2093	3656

Rule sets used in the experiment were induced by machine learning systems ID3, ID3/Drop, two machine learning options LEM1 and LEM2 and two knowledge acquisition options All Global Coverings and All Rules of the system LERS. Computational time complexity of all machine learning options is polynomial. Both knowledge acquisition methods of LERS are extremely time-consuming, since their computational complexity is exponential.

A description of rule sets used in all experiments is presented in Table 3. Each value in the table represents the number of rules induced for every data set. In the case of the SOYA data set, memory of both computers used for experiments was not sufficient to induce all rules.

In practice expert systems are forced to make decisions under uncertainty. We tested how expert systems performance will differ when decisions are made under incomplete information (a case of uncertainty). That means that values of some attributes are not given. It was simulated here by checking rule sets, induced from the same input data, by all six methods, against modified data sets. The modification was accomplished by deleting from the original data set:

1. one single attribute ($m = 1$),
2. two attributes ($m = 2$),
3. three attributes ($m = 3$).

In case 1 every single attribute was deleted from the original data set, one attribute at a time. Deletion of an attribute results in missing values of that attribute for all examples. In this way for the original data set n new modified data sets were created, where n is the number of attributes.

In case 2 every combination of two attributes was deleted (thus, for the original n-attribute data set $n * (n - 1)/ 2$ new modified data sets were created).

In case 3, for practical reasons, the following scheme was used. A sufficient number of triples of attributes, selected randomly, were deleted from the original data set. For each such triple a new modified data set was created. However, for practical reasons, the total number of modified data sets was kept below 600.

Each of the rule sets induced from the input data set was checked against a modified data set. This was done by Rule Checker, which is a special tool of LERS.

TABLE 4. Error rates for experiments

Data set	m	ID3	ID3/Drop	LEM1	LEM2	All Coverings	All Rules
BANK	1	41.21	24.54	24.54	24.54	23.64	7.58
	2	71.21	50.61	50.61	50.61	49.70	25.30
	3	90.60	74.39	74.39	74.39	73.93	53.33
BREAST	1	81.34	47.01	24.12	25.17	9.26	4.25
	2	45.46	48.60	45.46	48.60	23.07	12.58
	3	92.69	65.10	63.10	67.48	40.77	24.83
HOUSE	1	15.46	8.59	10.08	9.51	0.54	0.51
	2	28.77	17.72	20.79	19.39	1.60	1.38
	3	40.32	26.96	32.52	29.43	3.43	2.77
LYMPHO-	1	24.77	11.11	6.83	10.70	0	0
GRAPHY	2	44.14	22.04	17.88	22.13	0.009	0.009
	3	58.99	33.27	26.70	32.30	0.057	0.057
SOYA	1	5.10	3.47	4.07	3.46	0	0
	2	9.65	6.30	7.95	6.79	0	0
	3	14.04	9.88	11.50	10.00	0	0
TUMOR	1	43.86	23.47	21.48	24.68	5.67	3.57
	2	64.77	41.77	38.77	43.38	11.88	7.96
	3	75.71	55.96	52.67	57.59	18.82	13.39

In general, checking of a rule set against a data set may result in two kinds of errors: examples (members of a data set) that are wrongly classified (classified as being members of another concept) and examples that are not classified at all. In our case every single example is either classified correctly or not classified at all, so the first case never occurs. It may be explained as follows. For every modified data set and every option of LERS, during all experiments, the rule set was fixed. Such a rule set correctly classifies all examples from the original data set. Hence, it cannot wrongly classify any example with some missing values of attributes. Examples that are not classified during checking are called errors. In our experiments, the error

rate was computed as the ratio of the average number of errors to the total number of examples, for a modified data set, a given method of rule induction, and number *m* of deleted attributes. Table 4 presents error rates for all experiments.

5. CONCLUSIONS

The objective of this work was to compare the quality of rule sets obtained using the machine learning approach with those obtained by the knowledge acquisition approach, having expert system application in mind.

The more general objective of this work was to study how to improve a knowledge base that is a part of an expert system. Performance of an expert system depends to a great degree on the quality of its knowledge base. Thus it is crucial that the quality of the knowledge base be high. The most important issue is that the knowledge contained in the knowledge base be as complete as possible. Practically this means that even if some attributes are missing, an expert system may still make correct decisions.

Although the main objective of this work seems to be apparent, it should be noted that so far no research has been done to show that the machine learning approach, used for knowledge acquisition, is far from perfect. In current practice, the commercial systems of machine learning used in knowledge acquisition are mostly based on machine learning algorithm ID3.

Table 3 shows that the most complete rule set is induced by the All Rules option. The original hypothesis was that the quality of rules induced by this option should be the highest. Rule sets induced by the All Global Coverings option should follow the lead. Rule sets induced by machine learning systems are the least complete, since they are only sufficient and therefore should be of lower quality.

For a fixed option of LERS, say All Rules, the error rate differs greatly between different data sets. This is caused by the fact that in some data sets there are very few coverings and then deletion of some attributes significantly increases error rate. In data sets with many coverings the effect of deletion of attributes is not significant.

As follows from Table 4, the results of our experiments fully confirmed this hypothesis. It is clear that the smallest error rate is associated with rule sets induced by the All Rules option of LERS. Rule sets induced by the All Global Coverings option of LERS are worse in terms of error rate than those induced by All Rules and better than rule sets induced by any of machine learning methods.

Another conclusion is a result of comparison of four machine learning options, LEM1 and LEM2, from the viewpoint of quality of knowledge bases. The error rate of machine learning options was compared using the Wilcoxon matched-pairs signed rank test (Hamburg 1983). The results of pairwise comparison of machine learning methods are presented in Figure 1. First of all, performances of LEM1 and ID3/Drop do not differ significantly. The null hypothesis of equal performance cannot be rejected even with 5% significance level (two-tailed test). On the other hand, LEM2 performs better than ID3 with 0.5% significance level for one-tailed test, and both LEM1 and ID3/Drop perform better than LEM2 with 2.5% significance level for one-tailed test.

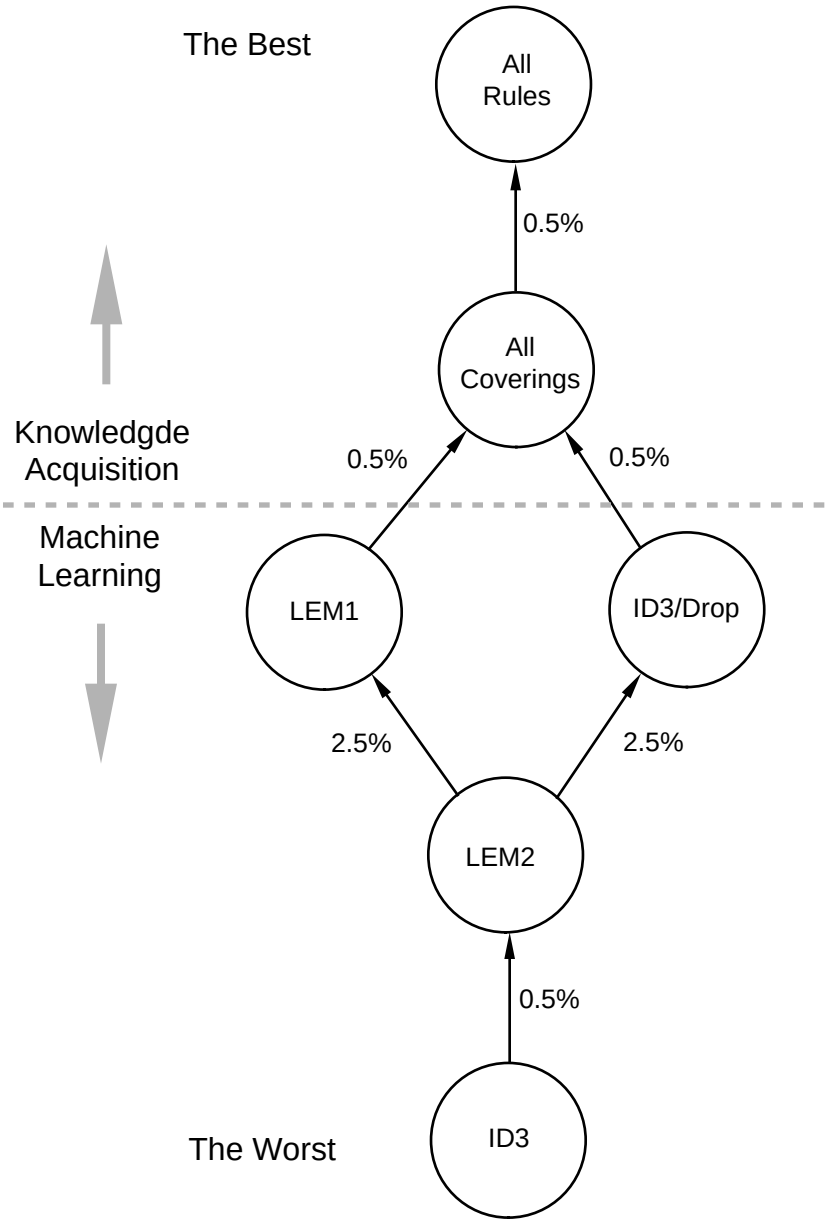

FIG. 1. Comparison of rule induction methods.

An additional experiment was performed to show that the sample of triples of attributes that were deleted from the original data set was of sufficient size. After every 50 data sets with randomly deleted triples of attributes the average error rate for all data sets selected up to that point was computed. Figure 2 presents how the average error changes with increasing number of modified data sets for the original data set TUMOR, $m = 3$ (three randomly deleted attributes) and option LEM2. It is clear that with the increasing number of data sets the error rate is stable and that an additional number of data sets with deleted triples will not change our analysis.

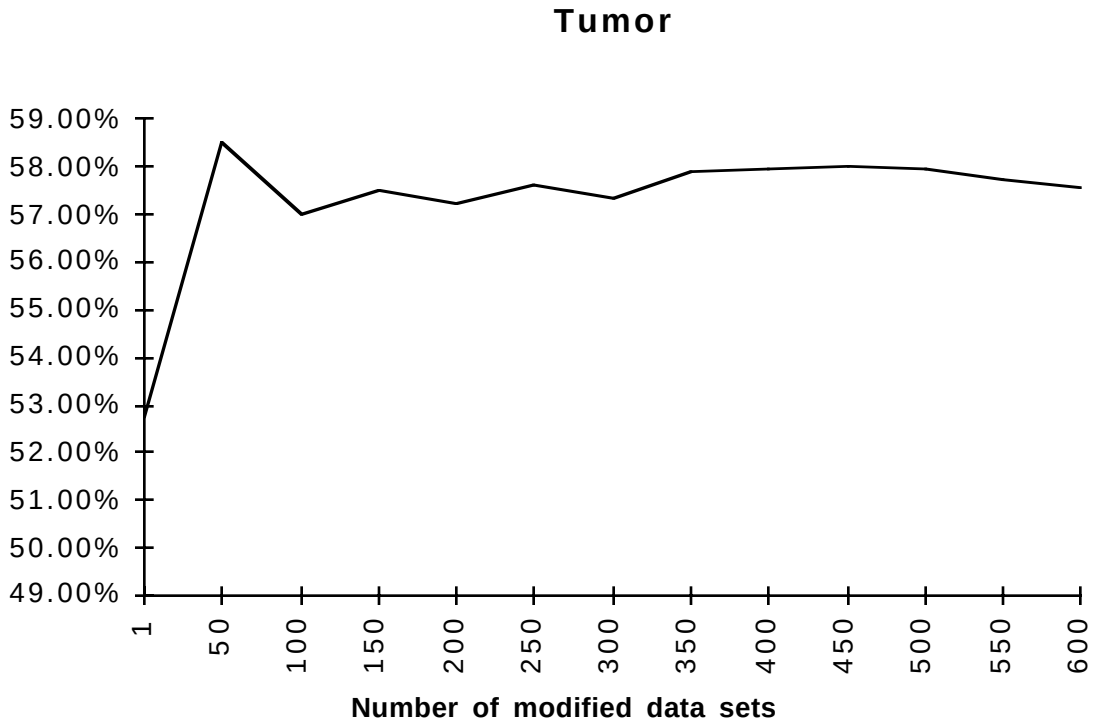

FIG. 2. Average error rate for LEM2 and three missing attributes.

The final conclusion is that machine learning methods used so far for rule induction in knowledge acquisition should be replaced by other methods of rule induction that will generate complete sets of rules. Options All Global Coverings and All Rules are examples of such appropriate ways of inducing rules for building knowledge bases.

REFERENCES

Addis, T. R., Y.Kodratoff, R. Lopez de Mantaras, K. Morik, and E. Plaza. 1991. Panel: Four stances on knowledge acquisition and machine learning. *Proceedings of the European Working Session on Learning—EWSL*-91, Porto, Portugal, pp. 514–533.

Boose, J. H. 1989. A survey of knowledge acquisition techniques and tools. Knowledge Acquisition, **1**: 3–37.

Buchanan, B. G. 1989. Can machine learning offer anything to expert systems? Machine Learning, **4**: 251–254.

Gams, M., M. Drobnic, and M. Petkovsek. 1991. Learning from examples—a uniform view. International Journal of Man-Machine Studies, **34**: 49–68.

Grzymala-Busse, J. W. 1991. Managing Uncertainty in Expert Systems. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Grzymala-Busse, J. W. 1992. 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. *Edited by* R. Slowinski. Kluwer Academic Publishers, Dordrecht, The Netherlands, pp. 3–18.

Grzymala-Busse, D. M., and J. W. Grzymala-Busse. 1993. Comparison of machine learning and knowledge acquisition methods of rule induction based on rough sets. Proceedings of the RSKD–93, International Workshop on Rough Sets and Knowledge Discovery, Banff, Alberta, Canada, pp. 297–306.

Hamburg, M. 1983. Statistical Analysis for Decision Making. 3rd ed. Harcourt Brace Jovanovich, Orlando, FL.

Hunt, E. B., J. Marin, and P. J. Stone. 1966. Experiments in Induction. Academic Press, San Diego, CA.

MacDonald, A. B., .and I. H. Witten. 1989. A framework for knowledge acquisition through techniques of concept learning. IEEE Transactions on Systems, Man, and Cybernetics, **19**: 499–512.

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

Pawlak, Z. 1991. Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Quinlan, J. R. 1982. Semi-autonomous acquisition of pattern-based knowledge. *In* Machine Intelligence 10. *Edited by* J. E. Hayes, D. Michie, and Y.-H. Pao. Ellis Horwood, Chichester, U.K., pp. 159–172.

Quinlan, J. R. 1983. Learning efficient classification procedures and their application to chess end games. *In* Machine Learning. *Edited by* R. S. Michalski, J. G. Carbonell, and T. M. Mitchell. Morgan Kaufmann, San Mateo, CA, pp. 461–482.

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

Quinlan, J. R. 1987a. Decision trees as probabilistic classifiers. Proceedings of the Fourth International Workshop on Machine Learning, Irvine, CA, pp. 31–37.

Quinlan, J. R. 1987b. Generating production rules from decision trees. Proceedings of the Tenth International Joint Conference on AI, Milano, Italy, pp. 304–307.

Shalin, V. L., E. J. Wisniewski, and K. R. Levi. 1988. A formal analysis of machine learning systems for knowledge acquisition. International Journal

of Man-Machine Studies, **29**: 429–446.

Weiss, S. M., and C. A. Kulikowski 1990. Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems. Morgan Kaufmann Publishers, San Mateo, CA.