BULLETIN OF THE POLISH
ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 34, No. 9–10, 1986

COMPUTER AND INFORMATION SCIENCE

Learning from Examples

by

Zdzisław PAWLAK

Presented by Z. PAWLAK on November, 12, 1985

Summary. In this article we propose a new approach to learning and inductive inference. We recommend the use of the rough set concept [9] as the mathematical basis for these areas. The suggested approach enables a precise, mathematical formulation of fundamental concepts of these areas, yields new theoretical results and offers simple learning algorithms.

1. Introduction. This paper is a modified version of [11], where the application of rough sets to learning from examples and induction was suggested.

2. Information system. In this section we introduce basic concepts needed to define precisely the idea of learning from examples, delivered by a teacher, an expert, environment, etc.

2.1. *Definition of information system.* We shall start our considerations from the notion of an information system.

By an information system we understand the 4-tuple

$$S = (\text{Univ}, \text{ Val}, \text{ Att}, f)$$

where:

Univ — is a finite set of objects,

Att — is a finite set of attributes,

$\text{Val} = \bigcup_{a \in \text{Att}} V_a$ — is domain of attribute a,

f: Univ × Att → Val — is a total function such that $f(x, y) \in \text{Val}_a$ for every $a \in \text{Att}$, $x \in \text{Univ}$ called information function.

The function f_x: Att $\rightarrow$ Val such that $f_x(a) = f(x, a)$ for every $x \in$ Univ, $a \in$ Att will be called information (data knowledge, description) about x in S. Any pair (a, v), $a \in$ Att, $v \in \mathrm{Val}_a$ is called descriptor in S.

Thus, an information system may be considered as a finite table in which columns are labelled by attributes, rows are labelled by objects and the entry in the a-th column and x-th row has the value $f(x, a)$.

2.2. *Indiscernibility relation.* Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be an information system and let $A \subseteq$ Att, $x, y \in$ Univ.

By $\tilde{A}$ we mean a binary relation on U (called an indiscernibility relation) — defined as follows: We say that x and y are indiscernible by the set of attributes A in S $(x\tilde{A}y)$ iff $f_x(a) = f_y(a)$ for every $a \in A$.

One can easily check that $\tilde{A}$ is an equivalence relation in Univ for every $A \subseteq$ Att.

The equivalence classes of the relation $\tilde{A}$ are called A-elementary sets in S.

Thus every $A \subseteq$ Att defines a classification (partition) of Univ — denoted A^*, and the equivalence classes of the relation $\tilde{A}$ are classes (blocks) of the classification A^*.

Certainly $A^* = \mathrm{Univ}/\tilde{A}$. We shall use the notation $\mathrm{Univ}/\tilde{A}$ when speaking about relations, and A^* when speaking about classifications.

If $\tilde{A}$ and $\tilde{B}$ are equivalence relations, then $\tilde{C} = \tilde{A} \cap \tilde{B}$ is called an intersection of $\tilde{A}$ and $\tilde{B}$, and is defined as follows:

$$x\tilde{C}y \quad \text{iff} \quad x\tilde{A}y \quad \text{and} \quad x\tilde{B}y.$$

It can easily be seen that

$$\tilde{A} = \bigcap_{a \in A} \tilde{a} \quad \text{for every} \quad A \subseteq \mathrm{Att}.$$

Any finite union of A-elementary sets will be called a A-definable set in S. An empty set is A-definable for every $A \subseteq$ Att in every S.

2.3. *Approximation of sets in an information system.* Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be an information system, $X \subseteq$ Univ and $A \subseteq$ Att.

By the A-lower (A-upper) approximation of $X \subset$ Univ in S, we mean the sets $\underline{A}X$ $(\bar{A}X)$ defined as follows:

$$\underline{A}X = \{x \in \mathrm{Univ}: [x]_{\tilde{A}} \subseteq X\}$$

$$\bar{A}X = \{x \in \mathrm{Univ}: [x]_{\tilde{A}} \cap X \neq \emptyset\}.$$

The set

$$Bn_A(X) = \bar{A}X - \underline{A}X$$

is referred as the A-boundary of X in S.

It is easy to check that each information system $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ and each subset of attributes $A \subseteq$ Att define a topological space $T_S =$

$= (\text{Univ}, \text{Def}_A(S))$, where $\text{Def}_A(S)$ is the family of all A-definable sets in S, and the lower and upper approximations are interior and closure in the topological space T_S. Hence the approximations have the following properties:

1) $\underline{A}X \subseteq X \subseteq \bar{A}X$

2) $\underline{A}\emptyset = \bar{A}\emptyset = \emptyset$; $\underline{A}\ \text{Univ} = \bar{A}\ \text{Univ} = \text{Univ}$

3) $\underline{A}(X \cup Y) \supseteq \underline{A}X \cup \underline{A}Y$

4) $\bar{A}(X \cup Y) = \bar{A}X \cup \bar{A}Y$

5) $\underline{A}(X \cap Y) = \underline{A}X \cap \underline{A}Y$

6) $\bar{A}(X \cap Y) \subseteq \bar{A}X \cap \bar{A}Y$

7) $\underline{A}(-X) = -\bar{A}(X)$

8) $\bar{A}-(-X) = -\underline{A}(X)$,

Moreover, for the topological space T_S we have:

9) $\underline{A}\underline{A}X = \bar{A}\underline{A}X$

10) $\bar{A}\bar{A}X = \underline{A}\bar{A}X$.

$\underline{A}X$ is called the A-positive region of X in S; $Bn_A X$ is called the A-doubtful region of X in S; $U - \bar{A}X$ is called the A-negative region of X in S.

2.4. *Accuracy of approximation.* With every subset $X \subseteq \text{Univ}$ we associate a number $\alpha_A(X)$ called the accuracy of approximation of X by A in S, or, in short, the accuracy of X, where A and S are defined as follows:

$$\alpha_A(X) = \frac{\text{card}\ \underline{A}X}{\text{card}\ \bar{A}X}.$$

Because of properties 3) and 6) (Section 2.3) we are unable to express the accuracy of the union and the intersection of sets X, Y in terms of the accuracies of X and Y.

2.5. *Non-definable sets.* Let $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ be an information system and let $A \subseteq \text{Att}$, $X \subseteq \text{Univ}$. Note that X is A-definable in S if $\underline{A}X = \bar{A}X$. We shall classify non-definable sets into the following classes:

a) X is roughly A-definable in S, iff $\underline{A}X \neq \emptyset$ and $\bar{A}X \neq \text{Univ}$,

b) X is internally A-non-definable in S, iff $\underline{A}X = \emptyset$ and $\bar{A}X \neq \text{Univ}$,

c) X is externally A-non-definable in S, if $\bar{A}X = \text{Univ}$ and $\underline{A}X \neq \emptyset$,

d) X is totally A-non-definable in S, iff $\underline{A}X = \emptyset$ and $\bar{A}X = \text{Univ}$.

Let us remark that if X is definable, roughly definable, or totally non-definable, so is $(-X)$; if X is internally (externally) non-definable, then $(-X)$ is externally (internally) non-definable.

2.6. *Approximation of families of sets.* Let $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ be an information system, $A \subset \text{Att}$, and let $\mathfrak{X} = \{X_1, X_2, \ldots, X_n\}$, where $X_i \subseteq \text{Univ}$, $i \geqslant 2$, be a family of subsets of Univ.

By the A-lower (A-upper) approximation of $\mathfrak{X}$ in S, denoted $\underline{A}\mathfrak{X}$ ($\bar{A}\mathfrak{X}$), we mean sets

$$\underline{A}\mathfrak{X} = \{\underline{A}X_1, \underline{A}X_2, \ldots, \underline{A}X_n\}$$

and

$$\bar{A}\mathfrak{X} = \{\bar{A}X_1, \bar{A}X_2, \ldots, \bar{A}X_n\},$$

respectively.

If $\mathfrak{X}$ is a classification (a partition) of Univ, i.e. $X_i \cap X_j = \emptyset$ for every $i, j \leqslant n$, $i \neq j$ and $\bigcup_{i=1}^{n} X_i = \text{Univ}$, then X_i are called classes (blocks) of $\mathfrak{X}$.

If every class of $\mathfrak{X}$ is A-definable, then the classification $\mathfrak{X}$ will be called A-definable.

$\text{Pos}_A(\mathfrak{X}) = \bigcup_{i=1}^{n} \underline{A}X_1$ will be called the A-positive region of the classification $\mathfrak{X}$ in S.

Since $\text{Univ} = \bigcup_{i=1}^{n} \bar{A}X_i$, there is no A-negative region for any A of the classification $\mathfrak{X}$ in S.

$Bn_A(\mathfrak{X}) = \sum_{i=1}^{n} Bn_A X_i$ will be called the A-doubtful region of the classification $\mathfrak{X}$ in S.

If $\mathfrak{X} = \{X_1, X_2, \ldots, X_n\}$ is a classification of Univ, then

$$\beta_A(\mathfrak{X}) = \frac{\sum_{i=1}^{n} \text{card}(\underline{A}X_i)}{\sum_{i=1}^{n} \text{card}(\bar{A}X_i)}$$

will be called the accuracy of the approximation of $\mathfrak{X}$ by A in S, or simply the accuracy of $\mathfrak{X}$.

$\beta_A(\mathfrak{X})$ expresses the ratio of all positive decisions to all possible decisions, when objects are classified by the set of attributes A.

We can also introduce another coefficient called a quality of approximation of the classification $\mathfrak{X} = \{X_1, X_2, \ldots, X_n\}$ by the set A of attributes, defined as follows:

$$\gamma_A(\mathfrak{X}) = \frac{\sum_{i=1}^{n} \text{card } \underline{A}X_i}{\text{card (Univ)}}.$$

Quality $\gamma_A(\mathfrak{X})$ expresses the ratio of all A-correctly classified objects to all objects in the system.

Obviously, $\beta_A(\mathfrak{X}) \leqslant \gamma_A(\mathfrak{X})$ and $\beta_A(\mathfrak{X}) = \gamma_A(\mathfrak{X})$ iff $\mathfrak{X}$ is A-definable.

2.7. *Dependence of attributes.* Let $S = (\text{Univ, Att, Val}, f)$ be an information system and let $A, B \subseteq \text{Att}$ be subsets of attributes.

We say that set of attributes B depends on the set of attributes A in S, $A \underset{S}{\rightarrow} B$ (or in short $A \rightarrow B$) iff $\tilde{A} \subseteq \tilde{B}$.

One can show by simple computation the following properties:

Fact 2.7.1. The following conditions are equivalent:

1) $A \underset{S}{\rightarrow} B$

2) $\widetilde{A \cup B} = \tilde{A}$

3) B^* is A-definable in S

4) $\underline{A}(B^*) = \bar{A}(B^*)$

5) $\gamma_A(B) = \beta_A(B^*) = 1$.

Fact 2.7.2.

1) if $B \subseteq A$, then $A \underset{S}{\rightarrow} B$

2) if $A \underset{S}{\rightarrow} B$ and $A' \supseteq A$, then $A' \underset{S}{\rightarrow} B$

3) if $A \underset{S}{\rightarrow} B$ and $B' \subseteq B$, then $A \underset{S}{\rightarrow} B'$

4) if $A \underset{S}{\rightarrow} B$, then $A \xrightarrow[S_{P \cup R}]{} B$.

A simple algorithm for checking whether $A \underset{S}{\rightarrow} B$ results or not from properties 1) (Fact 2.7.1) and 4) (Fact 2.7.2).

We say that B roughly depends on A in S, iff $0 < \gamma_A(B^*) < 1$. Then we write $A \underset{k}{\rightarrow} B$, where $k = \gamma_A(B^*)$.

If $A \underset{k}{\rightarrow} B$, then the dependence $A \rightarrow B$ holds for some objects only, namely for all $x \in \text{Pos}_A(B^*)$, i.e. the objects belonging to A-positive region of the classification B^*. The $\gamma_A(B^*)$ indicates the percentage of objects for which the dependence $A \rightarrow B$ holds. In other words, $A \underset{k}{\rightarrow} B$ iff $A \underset{k}{\rightarrow} B$ in $S/\text{Pos}_A(B^*)$.

We say that B is totally independent of A in S iff $\gamma_A(B^*) = 0$. The meaning of this definition is obvious.

2.8. *Reduction of attributes.* Let $S = (\text{Univ, Att, Val}, f)$ be an information system and let $A \subseteq \text{Att}$.

a) $A \subseteq \mathrm{Att}$ is independent in S iff for every $A' \subset A$, $\tilde{A}' \supset \tilde{A}$,

b) $A \subseteq \mathrm{Att}$ is dependent in S iff there exists $A' \subset A$, such that $\tilde{A}' = \tilde{A}$.

c) $A \subseteq \mathrm{Att}$ is reduct of Att in S iff A is the maximal independent set in S.

It can be easily shown that the following properties hold:

Fact 2.8.1.

a) If A is independent in S, then for every $a, b \in A$ neither $a \underset{S}{\rightarrow} b$ nor $b \underset{S}{\rightarrow} a$, i.e. all attributes from A are pairwise independent.

b) If A is dependent in S, then there exists $A' \subset A$, independent in S, such that $A' \underset{S}{\rightarrow} A - A'$.

3. Application to learning. Machine learning from examples can be very easily formulated in our approach leading to new important theoretical and practical results.

In order to avoid confusion with the existing terminology, we introduce new terms for machine learning: static and dynamic learning, discussed in two successive sections of this paper.

3.1. *Static learning.* Suppose we are given a finite set Univ of objects. Elements of Univ are called training examples (instances) and Univ is called training set. Assume further that Univ is classified into disjoint classes $X_1, X_2, \ldots, X_n$ $(n \geqslant 2)$ by a teacher (expert, environment). The classification represents the teacher's knowledge of objects from Univ. Furthermore, let us assume that a student is able to characterize each object from Univ in terms of attributes from set A. Description of objects in terms of attributes from A represents the student's knowledge of objects from Univ.

We can say that the teacher has semantic knowledge and the student — syntactical knowledge of objects from Univ.

The problem we are going to discuss in this section is whether the student's knowledge can be matched with the teacher's knowledge, or, more precisely, whether the teacher's classification can be described in terms of attributes available to the student.

Thus, static learning consists in describing classes $X_1, X_2, \ldots, X_n$ in terms of attributes from A, or more exactly, in finding a classification algorithm which provides the teacher's classification on the basis of properties of objects expressed in terms of attributes from A.

The problem of static learning can be formulated precisely in terms of concepts introduced in the previous sections as follows:

Let $S = (\mathrm{Univ}, \mathrm{Att}, \mathrm{Val}, f)$ be an information system, associated with the student's knowledge of elements of Univ. Note that f_x is student's knowledge about x in S. Let us extend system S by adding a new attribute e representing the classification provided by the teacher, i.e., $e = e^* = \{X_1, X_2, ..., X_n\}$. Thus we obtain a new information system $S' = (\text{Univ}, \text{Att}, \text{Val}, f)$, where $\text{Att} = A \cup \{e\}$, $A \cap \{e\} = \emptyset$, $\text{Val}' = \text{Val} \cup \{1, 2, ..., n\}$, $f'/\text{Univ} \times A = f$, $f'_x(x) = i$ iff $x \in X_i$. The $f'_x(e)$, called teacher's knowledge on x in S', is the number of the class to which x belongs according to the teacher's knowledge.

Thus, the problem of static learning is reduced to the question whether the classification e^* is A-definable. In virtue of property 2.7.1., e^* is A-definable iff $A \to e$, i.e. the problem whether there exists an algorithm to "learn" classification e^* by checking the properties of objects, is reduced to proving whether the attribute e depends on the set of attributes A in S'.

If the dependence $A \to e$ holds, one can formulate an (A, e)-decision algorithm (see [12]), which will be here called a learning or classification algorithm, and which represents the dependence function. In other words, the algorithm can be used directly as a learning algorithm.

Because the algorithm is a set of decision (classification) rules, this means that learning a classification consists in finding the classification rules.

3.2. *Example.* Let us consider for example an information system given in the Table presented below and let us assume that the attribute c in that system represents the classification c^*, provided by the teacher.

Univ	a	b	c
x_1	1	0	2
x_2	0	1	1
x_3	2	0	0
x_4	1	0	0

We ask whether the classification can be expressed by attributes a and b.

Because the dependence $\{a, b\} \to c$ holds, the learning algorithm exists, and it has the form

$$(a: = 1)\ (b: = 0) \Rightarrow (c: = 2)$$

$$(a: = 0) \Rightarrow (c: = 1)$$

$$(a: = 2) + (a: = 1)\ (b: = 1) \Rightarrow (c: = 0).$$

Note that we are not allowed to remove a or b because the set $\{a, b\}$ is independent in S.

It may happen, that the teacher classification c is not A-definable. That is to mean that the learning algorithm does not exist, and it is impossible to classify correctly the objects by examining their features.

In such a case it is possible to classify objects only approximately, i.e. to approximate the classification c^* by the set of attributes A. This is to say, that we are unable to classify every object correctly; only some objects (possibly zero) can be classified properly in this case.

Obviously, there is no deterministic classification algorithm in the case of an approximate classification, but there is non-deterministic one.

Two coefficients — accuracy and quality — of the approximate classification show which part of objects can be classified correctly (quality) and which part of decisions can be correct (accuracy).

It can easily be seen from table above, that the classification c^* is not (a, b)-definable. Hence we can approximate the classification c by set of attributes $\{a, b\}$.

In order to do that, let us first compute classes of the classification c^* (equivalence classes of relation c), which are as follows:

$$Y_1 = \{x_1, x_4, x_{12}\}$$

$$Y_2 = \{x_2, x_6, x_9\}$$

$$Y_3 = \{x_3, x_5, x_7, x_9, x_{10}, x_{11}\}.$$

The equivalence classes of relation $\{a, b\}$ are following

$$X_1 = \{x_1, x_4, x_5, x_8, x_{11}, x_{12}\}$$

$$X_2 = \{x_2, x_6, x_9\}$$

$$X_3 = \{x_3, x_7, x_{10}\}.$$

Let us set $A = \{a, b\}$. Then the following sets are the lower A-approximation of c^*:

$$\underline{A}Y_1 = \emptyset$$

$$\underline{A}Y_2 = X_2$$

$$\underline{A}Y_3 = X_3$$

and the upper A-approximation of c^* is:

$$\bar{A}Y_1 = X_1$$

$$\bar{A}Y_2 = X_2$$

$$\bar{A}Y_3 = X_1 \cup X_2 \cup X_3.$$

Thus the class Y_1 is internally A-non-definable, Y_2 is A-definable, and Y_3 is roughly A-definable.

The corresponding accuracy coefficients are:

$$\alpha_A(Y_1) = 0$$

$$\alpha_A(Y_2) = 1$$

$$\alpha_A(Y_3) = 0.5.$$

Thus it is impossible to learn positive instances of Y_1, but it is possible to learn negative instances of Y_1, (if $x \in Y_2 \cup Y_3$ we know that x is not in Y_1).

In other words, it is impossible to classify correctly x_1, x_4, x_{12} by observing their features expressed by a and b.

Y_2 can be learned fully, i.e. all elements of Y_2 can be classified correctly on the basis of their features expressed by a and b.

Y_3 can be learned only roughly, i.e. only objects x_3, x_7, x_{10} can be recognized on the basis of a and b as elements of Y_3; objects x_2, x_6, x_9 can be excluded from Y_3, and $Y_1 = \{x_1, x_4, x_5, x_8, x_{11}, x_{12}\}$ is the doubtful region of Y_3, i.e. it cannot be decided on the basis of a and b whether the elements of X_1 are, or are not, in Y_3.

The accuracy and quality of learning are:

$$\beta_A(c^*) = \frac{\text{card}(\underline{A}Y_2) + \text{card}(\underline{A}Y_3)}{\text{card}(\bar{A}Y_1) + \text{card}(\bar{A}Y_2) + \text{card}(\bar{A}Y_3)} = 9/18 = 0.5\cdot$$

$$\gamma_A(c^*) = \frac{\text{card}(\underline{A}Y_2) + \text{card}(\underline{A}Y_3)}{\text{card}(\text{Univ})} = 9/12 = 0.75,$$

which means that at most 75 per cent of instances can be classified correctly and at most 50 per cent of decision can be correct.

Note also that $\{a, b\}$ has one reduct, namely a. This means that it is not necessary to have both a and b to learn the classification c^* but it is enough to use a only.

3.3. *Dynamic learning.* Static learning consists in a description of objects by a student classification provided by the teacher, or in other words, in learning classification (decision) rules on the basis of training examples provided by the teacher. The classification rules learned from training examples can be assumed as the background knowledge of the student. The question arises whether the background knowledge can be used to classify correctly the new objects not occurring in training examples.

Classification of new objects on the basis of background knowledge previously acquired from training examples will be called the dynamic learning.

The problem of dynamic learning can also be regarded as a kind of inductive generalization (inference), but we shall not consider this problem here.

Discussion on induction can be found in [2]. In [5] the inductive generalization from the point of view of the rough set theory is discussed, however, in our approach we assume a somewhat different approach.

In this section we shall consider the problem of dynamic learning in terms of concepts introduced in previous sections.

Let $S = (\text{Univ}, \text{Att}, \text{Val}, f)$ be an information system, where Univ is the training set, $\text{Att} = A \cup \{e\}$, A — is the set of attributes associated with student, e — is the teacher attribute providing classification e^* of training examples, and let $\mathfrak{A}$ be the classification algorithm (see [12]) resulting from the set Univ of training examples.

Assume that the student has to classify a new object x (not belonging to the training set Univ) using the classification algorithm $\mathfrak{A}$. Let t_x be an A-elementary term describing object x.

If in the classification algorithm $\mathfrak{A}$ there is a decision rule $t_i \Rightarrow t_i'$ such that $t_i = t_x$, the student will assign object x to the set $f(t_i')$. If there is no such rule, the student is unable to classify the new object by means of algorithm $\mathfrak{A}$.

We assume that the teacher also classifies the new objects according to his knowledge. If both decisions (that of the student and that of the teacher) agree, the student classification is correct — otherwise the classification is incorrect.

Thus by adding a new object x, we face the following possibilities:

1) the student classification of x is correct,

2) the student classification of x is incorrect,

3) the student is unable to classify the new object x.

In order to show how the background knowledge influences the correctness of student decisions we have to investigate how the accuracy and quality of learning can change in all above mentioned three situations.

Because adding a new object x to the set Univ results in a new information system S', our task is to compare the coefficients β_A and $\beta_{A'}$ for S and S', respectively, in the three above mentioned situations (correct, incorrect, classification impossible).

Let us remark that adding a new object x to the set Univ changes also the teacher classification. The new object can match one of existing classes or it can form a completely new single element class.

The accuracy coefficient for these three situations is given below:

a) Correct classification:

$$A'(e) = \frac{\text{card}\, \underline{A}(e^*) + 1}{\text{card}\, \bar{A}(e^*) + k_p(x)}$$

where

$\underline{A}(e^*)$ $(\bar{A}(e^*))$ is the lower (upper) approximation of the classification $e^* = \{X_1, X_2, ..., X_n\}$ in S,

card $\underline{A}(e^*) = \sum_{i=1}^{n}$ card $\underline{A}X_i$,

card $\bar{A}(e^*) = \sum_{i=1}^{n}$ card $\bar{A}X_i$,

$K_A(x)$ — arity of x in S with respect to A.

The arity of x with respect to A in S, $k_A(x)$ is the maximal number of classes $X_{i_1}, X_{i_2}, ..., X_{i_m}$ in the classification e^* such that

$$x \in \bigcap_{j=1}^{m} \bar{A}X_{i_j}.$$

b) Incorrect classification:

$$\beta_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*) - \text{card } [x]}{\text{card } \bar{A}(e^*) + k_A(x)}$$

where $[x]$ is the set of all objects in Univ$\cup x$ having the same description as x.

c) Classification impossible:

$$\beta_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*)}{\text{card } \bar{A}(e^*) + 1}.$$

The quality coefficient has the value:

d) Correct classification:

$$\gamma_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*) + 1}{\text{card Univ} + 1}$$

e) Incorrect classification:

$$\gamma_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*) - \text{card } x}{\text{card Univ} + 1}$$

f) Classification impossible:

$$\gamma_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*)}{\text{card Univ} + 1}$$

Let us discuss briefly the above formulas. Consider firstly the deterministic case, when the classification algorithm is deterministic. In this case both coefficients β' and γ' are the same and have the form:

g) Correct classification:

$$\beta_{A'}(e^*) = \gamma_{A'}(e^*) = \frac{\text{card Univ}+1}{\text{card Univ}+1} = \frac{\text{card Univ}}{\text{card Univ}} = \beta_A(e^*) = \gamma_A(e^*) = 1$$

h) Incorrect classification:

$$\beta_{A'}(e^*) = \gamma_{A'}(e^*) = \frac{\text{card Univ} - \text{card } [x]}{\text{card Univ}+1}$$

i) Classification impossible:

$$\beta_{A'}(e^*) = \gamma_{A'}(e^*) = \frac{\text{card Univ}}{\text{card Univ}+1}.$$

This is to say that:

1. Correct classification does not change the accuracy and quality of learning.

2. Incorrect classification decreases the accuracy and quality of learning "essentially".

3. Impossibility of classification decreases the accuracy and quality of learning "slightly".

Informally, it can be explained as follows:

1. If the training set Univ has all possible types of objects, adding a new object does not improve the background knowledge and this knowledge is sufficient to learn properly how to classify any new object.

2. If the set of attributes A is not large enough then the student may face a situation in which the new object x has the same secription as another object y in the training set Univ, but x and y belong to two different classes according to the teacher knowledge. This is to say that these two objects are different in the teacher opinion, while the student is unable to distinguish them by checking their properties (attributes from the set $\tilde{A}$), which leads to an incorrect classification. Thus, in such a case the background knowledge is not sufficient to classify a new object correctly.

3. If the set of examples Univ is not large enough it may happen that the new object x has a completely new description in terms of attributes from A, and this description does not match any description of objects in the training set Univ. So, the student is unable to classify this object by means of the classification algorithm. Also in this case the background knowledge is not sufficient to classify the new object correctly.

The above discussion could be more precise if we used the concept of a sample of a set (see [9]), but this lays outside the scope of the article.

Let us now discuss the case when the classification algorithm is non-deterministic. The accuracy of learning in this case is the following:

j) Correct classification:

$$\beta_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*)+1}{\text{card } \bar{A}(e^*)+1} > \frac{\text{card } \underline{A}(e^*)}{\text{card } \bar{A}(e^*)} = \beta_A(e^*)$$

k) Incorrect classification:

$$\beta_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*) - \text{card } [x]}{\text{card } \bar{A}(e^*) + k_A(x)}$$

l) Classification impossible:

$$\beta_{A'}(e^*) = \frac{\text{card } \underline{A}(e^*)}{\text{card } \bar{A}(e^*)+1}.$$

It can easily be seen that in the case of correct classification the accuracy is not decreasing with a new experience (new objects). This means that the background knowledge can be improved by proper new examples in contrast to the previous case of deterministic algorithm.

The case of incorrect classification by non-deterministic classification algorithm needs some more explanation. Incorrect classification means that the student is unable to assign the new object to any single class, although he is able to point out several classes to which the object may belong. However, according to our definition, this is not a proper classification. Therefore, the accuracy is decreasing in this case. The last case is obvious.

Similar discussion can be provided for the quality coefficient and is left to the reader.

To sum up, if the student background knowledge is complete in a certain sense (the classification algorithm is deterministic) it provides the highest accuracy and quality, and it is impossible to increase the classification skills of the student by new examples. If the background knowledge is incomplete (the classification algorithm is non-deterministic) the classification skills of the student can be improved by the properly chosen new training examples.

DEPARTMENT OF COMPLEX CONTROL SYSTEMS, POLISH ACADEMY OF SCIENCES, BAŁTYCKA 6, 44-100 GLIWICE

(ZAKŁAD SYSTEMÓW AUTOMATYKI KOMPLEKSOWEJ PAN)

DEPARTMENT OF COMPUTER SCIENCE, UNIVERSITY OF NORTH CAROLINA, NC 28223, USA

REFERENCES

[1] R. B. Banerji, *Artificial intelligence: a theoretical perspective*, Elsevier-North Holland, New York, 1980.

[2] A. Bar, E. Reingenbaum, *The handbook of artificial intelligence*, Harris Tech. Press, Stanford, 1981.

[3] W. Marek, Z. Pawlak, *Information systems: mathematical foundations*, Theor. Comp. Sci., **1** (1978), 331–354.

[4] R. Michalski, J. Carbonell, T. Mitchell, *Machine learning*, Tioga Publishing Company, Palo Alto, 1980.

[5] E. Orłowska, *Semantical analysis of inductive reasoning*, Tech. Note, **210** (1984).

[6] E. Orłowska, *Dependences of attributes in Pawlak's information systems*, Fund. Inform., **6** (1980), 247–256.

[7] E. Orłowska, Z. Pawlak, *Logical foundations of knowledge representation*, ICS PAS Reports, **537** (1984), 1–108.

[8] Z. Pawlak, *Information systems; theoretical foundations*, Inform. Syst., **6** (1981), 205–218.

[9] Z. Pawlak, *Rough sets*, Int. J. Inform. Comp. Sci., **11** (1982), 341–356.

[10] Z. Pawlak, *Rough classification*, Int. J. Man-Machine Studies, **20** (1984), 469–483.

[11] Z. Pawlak, *On learning — a rough set approach*, Lecture Notes, Springer-Verlag (in press).

[12] Z. Pawlak, *On decision tables*, Bull. Pol. Ac. Tech., **34** (1986), 563–571.

З. Павляк, **Обучение по примерам**

В настоящей статье предлагается новый подход к обучению по примерам и к доказательству методом индукции. Предлагается применение приближенных множеств как математической основы этих областей. Вышеназванный подход предоставляет возможность точно математически сформулировать основные понятия этих областей, ведет к новым теоретическим результатам и простым обучающим алгоритмам.