Recognition With Learning As a Problem of Choice

Obraztsov V.
Belarusian State University, 220030 Minsk, Nezavisimosti av. 4, e-mail: obraztsov@bsu.by

Abstract. *The paper highlights a problem of pattern recognition with learning as a part of a wider problem of choice. It is shown that by information basis the problem can be compared with problems of formulae classification in sentence calculus and logical diagnostics. A family of algorithms is constructed that can be used for solving any of the above-mentioned problems. It is shown that an algorithm from the family can be compared with classical ones (method of resolution, etc), and, therefore, is justified at least in the algorithmic sense.*

Keywords. pattern recognition, a problem of choice, justification of algorithms.

1. INTRODUCTION

Mathematical theory of pattern recognition is one of the most difficult leads of the applied mathematics. By now a considerable number of approaches has been developed for solving problems of recognition [1, 2]. In spite of the fact D. Polya's fundamental question "How to solve it ?" [3] as applied to the theory of pattern recognition has not been satisfactorily answered.

In practice one falls to thinking about the rightfulness of using different approaches and algorithms. There arises a typical situation when a problem is "fitted" to the algorithm (more precisely to the approach, if it is considered as a means of problem formalization), and consequently the latter most often dominates over the meaning of the problem, although it should be vice versa. In each case a problem of choosing algorithms is substituted for the problem of choosing the interpretation which is methodological by nature. As a result of such "fitting" the initial problem is somehow transformed to the level of tradition and liking of a particular researcher and this may take far away from the problem under consideration.

At the same time, in spite of diversity of languages used for formalization, all problems have a common statement kernel in respect to the language of the set theory. And this can be used both for their analysis and for making a common algorithmic basis. Moreover, it appears that in this case problems of recognition are found in a class along with such "classical" problems as formulae classification in sentence calculus. For the last-named problems the choice and justification of algorithms have been studying for rather a long time and the procedure of justification is well-known. With a common statement it is natural to make an attempt to determine a unified algorithmic basis for them and to show that it is not worse than previously used algorithms for the problem of formulae classification in sentence calculus and other ones. This approach is under consideration in this paper.

2. ANALYSIS OF A PROBLEM OF CHOICE

Let's consider a problem that on the level of meaning can be formulated in the following manner:
in a set of $\mathcal{X}$ objects of arbitrary nature a certain and possibly an infinite number of subsets (classes) $\mathcal{X}_1,\dots,\mathcal{X}_l$ is specified. It is necessary to determine an algorithm ***A*** *(may be the best one in some sense), that is defined on the whole set $\mathcal{X}$ and the result of which work for each $x \in \mathcal{X}$ can be interpreted in terms of belonging to subsets $\mathcal{X}_i$.*

The level of formalization of the given problem can be judged by such notions as *a set, a subset and an algorithm.* These notions, as is generally known, belong to fundamental ones in mathematics, i.e. they form the foundation of the whole mathematical shell. Thus, the formulated problem by its statement is on the maximum level of generality. In other words the case in point is a class of problems. The best name for this class is **a problem of choice**. This precisely corresponds to the result of solving any problem from the given class, i.e. the **choice** of one of the subsets mentioned above.

For the specified level of formalization there arises a number of typical questions. The main one is as follows. *What can be said about the solvability of the problem of choice and can the solution be justified?* Without a satisfactory answer solution of any applied tasks, being reduced to a particular case of the problem of choice, will always be imperfect. It is not so difficult to answer the last part of the question because numerous works in the field of mathematics foundation are dedicated to this during the last century [4]. Unfortunately, the answer in the general case is the negative even when we speak about a mathematical formalism [5], not to mention applied tasks, inductive by their nature. But there are problems for which the answer is, however, the affirmative. Among these are a problem of formulae classification in sentence calculus with a method of resolution as an algorithm ***A***, and a number of related technical problems of features recognition. These problems, as a matter of fact, constitute the subject of classical mathematics [6].

It is clear that each particular problem greatly depends on peculiarities of the subject area and requirements that are imposed upon algorithms ***A***. Results obtained when analyzing the problem of choice, as well as experts' opinions have shown that the following restrictions are observed:

Restrictions:

1. finite number a of subsets (classes) $\mathcal{X}_1,\dots,\mathcal{X}_l$ $(l \in \mathbf{N})$;
2. initial information about objects $x \in \mathcal{X}$ possessing the property $P_i(x)="x \in \mathcal{X}_i"$ for all $i=1,\dots,l$.
3. an algorithm ***A*** (defined on the whole set $\mathcal{X}$) that calculates the value of properties $P_1(x),\dots,P_l(x)$ for each $x \in \mathcal{X}$.
4. Each class $\mathcal{X}_i$ is well structured in the system of binary characters $\mathbf{B}_2^n(\mathbf{B}_2=\{0,1\})$, i.e. it is possible to indicate subspaces of independent and incoherent

features $\mathbf{B}_2^{n_1},\mathbf{B}_2^{n_2},\ldots,\mathbf{B}_2^{n_k}$ $(n_1+\ldots+n_k=n)$, where $k\in\mathbf{N}$ is determined by a particular localization;

5. Each class $\mathcal{X}_i$ in the subspace $\mathbf{B}_2^{n_j}$ is associated with a set of rules $S_{ji}(j=1,\ldots,k;i=1,\ldots,l)$ where their verity domain determines the initial information about the class. When forming the rules, as it turned out, we may restrict ourselves to statement calculus, i.e. Boolean algebra without quantifiers [7];
6. Verity domains
$\mathcal{X}_i^0=\{x\in\mathbf{B}_2^n: P_i(x)=S_{1i}(x)\wedge\ldots\wedge S_{ki}(x)=\mathbf{truth}\}$
have the property
$$\bigcup_{i=1}^{l}\mathcal{X}_i^0\subset\mathbf{B}_2^n \tag{1}$$

i.e. rules determine some initial information about classes $\mathcal{X}_1,\ldots,\mathcal{X}_l$. Nothing is known about the subset $\mathcal{X}_i\bigcap\mathcal{X}_j$ and classes, in general, may intersect.

7. The sought for algorithm $\boldsymbol{A}$ should be defined on the whole set $\mathbf{B}_2^n$ so that results of its operation can be treated as realization of properties $(P_1,\ldots,P_l)$. And the algorithm should have certain monotony, i.e. it should calculate values of properties on partial and redundant descriptions. It is also desirable that the results of its operation can be interpreted, with the specified degree of detail, i.e. there should be a possibility to justify (or explain) these results.

Using results of the carried out analysis, the above-mentioned problem statement was revised and conditions concerning algorithm $\boldsymbol{A}$ were defined. As for problem statement, actually, a space of the formation of objects $\mathcal{X}\subseteq\mathbf{B}_2^n$ and a way of specifying initial information by rules were revised. Other features of the problem statement remained the same. Conditions for choosing an algorithm can be described in the following manner. Each algorithm $\boldsymbol{A}$ is a representation

$$\boldsymbol{A}:\mathbf{B}_2^n\to[0,1]^l \tag{2}$$

and if we denote $A(x)=(\mu_A^1(x),\ldots,\mu_A^l(x))$, where $\mu_A^i(x)$ is a value of property P_i for an object $x\in\mathcal{X}$ (that can be called **a degree of property confirmation**, because $\mu_A^i(x)\in[0,1]$), then the condition of monotony of algorithm $\boldsymbol{A}$ can be written as

$$\forall x\in\bigcup_{j=1}^{l}\mathcal{X}_j^0\ \forall i=1,\ldots,l\ \begin{cases}\mu_A^i(\bar{x})\le\mu_A^i(x),\\ \mu_A^i(\tilde{x})\le\mu_A^i(x),\end{cases} \tag{3}$$

where $\bar{x},\tilde{x}$ - are objects in $\mathcal{X}$ that are obtained from x by eliminating and adding meaningful features (i.e. that are equal to 1).

Thus, we obtain a problem of decision-making (diagnostics and treatment) that is solved by algorithms of (2) type with the restriction (3) and condition that objects can be described in space $\mathbf{B}_2^n$ and the initial information is specified by rules in the language of statement calculus in such a way that (1) is satisfied.

3. PROBLEM SOLUTION

Notice that any algorithm solving the problem stated above can be realized and used in the system only when it withstands a certain testing, approbation. Methodology of such testing for expert systems is well known and greatly depends on aims of development [6, 7, 8]. As a practical matter it means that solution should be found in a set of algorithms possessing parameters (in the model), because in this case the choice of algorithms becomes considerably easy. Besides, when using parameters we may expect to obtain some additional "good" properties of algorithms, e.g. convergence, etc. It is precisely this reasoning that determined the approach to problem solution described below.

3.1. Canonical algorithm

One can discuss for a long time that algorithms solving the stated problem are inductive both by the nature of information and principles of their construction [6, 7]. But, it appears that the problem has at least two very simple (deductive by techniques) solutions: with the help of resolution methods and exhaustion. In the latter case it is necessary first to transform the initial information from representation in the form of rules to "object" representation in the space $\mathbf{B}_2^n$ or $\mathbf{B}_2^{n_1}\times\ldots\times\mathbf{B}_2^{n_k}$. To carry out this transformation we may use an algorithm for building a disjunctive normal form (DNF). Equivalence between the initial and obtained representations is quite evident [6, 8]. We will not extensively discuss details of such algorithms because they can be found practically in any manual on artificial intelligence or logic's. Results of their operation are equivalent and can be presented in the following manner using designations mentioned above

$$\mu_A^i(x)=\begin{cases}1, if\ S_{1i}(x)\wedge\ldots\wedge S_{ki}(x)=\mathbf{truth}\ or\ x\in\mathcal{X}_i^0,\\ 0, in\ all\ other\ cases.\end{cases} \tag{4}$$

It is easy to notice that these algorithms $\boldsymbol{A}$ solve the problem stated above. At the same time $\mu_A^i(x)\in\{0,1\}$. The algorithms also satisfy the condition (3).

For us such algorithms will be source (**canonical**) ones and that is why we denote them by a special symbol $\boldsymbol{A_0}$. The solution will be sought in a set of algorithms that are as good as $\boldsymbol{A}_0$ from the viewpoint of the following definition

We call algorithm $\boldsymbol{A_0}$ **dominating** $\boldsymbol{A}$ (*let us denote* $\boldsymbol{A}\succ\boldsymbol{A}_0$) *if*
$$\forall x\in X\ \forall i\in\{1,\ldots,l\}\,(\mu_A^i(x)\ge\mu_{A_0}^i(x)).$$

It is easy to see that any algorithm dominating $\boldsymbol{A}_0$

also solves the problem under consideration, but is less categorical outside the sample $\bigcup \boldsymbol{X}_i^0$. Besides, since legitimacy of using $\boldsymbol{A}_0$ for the problem solution is beyond question, then proof of domination also lets us make a similar conclusion about algorithm $\boldsymbol{A}$. Although we realize that such conditions are necessary but insufficient for categorical justification. But, apparently, any categorical justification is possible not earlier then the inductive conclusion as a whole is justified [9].

3.2. Dominating algorithm

Below we will describe one set of algorithms $\boldsymbol{A}$ and show that any of them dominates $\boldsymbol{A}_0$. In the following we will consider some properties of the constructed algorithms. Where it is possible, without sacrifice to understanding, we will leave evident details and restrict our consideration to references.

Let's start from the general scheme of algorithm $\boldsymbol{A}$. For this purpose we will introduce additional designations and assumptions. Let's consider that:

1. when dividing $\mathbf{B}_2^n$ into subsets $\mathbf{B}_2^{n_1},\ldots,\mathbf{B}_2^{n_k}$ each of them has features $I_j \subset \{1,\ldots,n\}$ and $|I_j| = n_j$. Without loss of generality we assume that the set $\{1,\ldots,n\}$ is numbered in the following manner: n_1 features from $\mathbf{B}_2^{n_1}$ (i.e. $I_1 = \{1,\ldots,n_1\}$) come first, followed by n_2 features from $\mathbf{B}_2^{n_2}$ (i.e. $I_2 = \{n_1+1,\ldots,n_1+n_2\}$), etc.;
2. sets $\{1,\ldots,l\} \times I_j$ are associated with:
   - vectors $(a_{im+1},\ldots,a_{im+n_j}) \in \mathbf{R}^{n_j}$ (where $\mathbf{R}$ - are real numbers, $m = \sum_{u=1}^{j-1} n_u$) characterized by $a_{iu} \geq 0$ for all suitable i,u;
   - sets of objects $\boldsymbol{X}_{ij}^0$ that exhaust verity domain of rules from S_{ij} and that are obtained by transformation of the rules to DNF. It is evident that between conjuncts of such DNF and objects from $\boldsymbol{X}_{ij}^0$ there is one-to-one onto function [6].

Now $\boldsymbol{A}$ can be described as a sequence of the following steps.

<u>Algorithm</u>.

Step 1. For the specified object $x \in \mathbf{B}_2^n$ we perform

Step 1.1 We fix the number $u \in \{1,\ldots,k\}$ and pass to the next step.

Step 1.2 For each $i \in \{1,\ldots,l\}$ and for all $x' \in \boldsymbol{X}_{ui}^0$ we calculate

$$\mu_A^{i,u}(x,x') = \mathbf{max}\{0,((\sum_{v\in I_u}(-1)^t \cdot a_{iv}) \cdot (\sum_{v\in I_u} a_{iv})^{-1})\}$$

where

$$t = \begin{cases} 1, if\ x_v \neq x'_v, \\ 2, otherwise. \end{cases}$$

Step 1.3 If all $x' \in \boldsymbol{X}_{ui}^0$ are exhausted, we calculate

$$\mu_A^{i,u}(x) = \mathbf{max}_{x'\in X_{ui}^0}\{\mu_A^{i,u}(x,x')\}$$

Step 1.4 If all numbers i are not exhausted, we return to *step 1.2*. Otherwise, we pass to the next step.

Step 1.5 If all u are not exhausted, we return to *step 1.1*. Otherwise, we pass to *step 2*.

Step 2. For each $i \in \{1,\ldots,l\}$ we calculate

$$\mu_A^i = (\sum_{u=1}^k \gamma_u \cdot \mu_A^{i,u}(x)) \cdot (\sum_{u=1}^k (\gamma_u \cdot \sum_{v\in I_u} a_{iv}))^{-1}$$

where $\gamma_u \in \mathbf{R}(\gamma_u \geq 0)$ is a measure of "importance" of the subset of features I_u.

Step 3. If a set of objects $\boldsymbol{X}$ is not exhausted, we return to *step 1*. Otherwise, the algorithm finishes its work.

It is easy to see that a particular algorithm $\boldsymbol{A}$, in accordance with this scheme, is associated with the choice of parameters a_{iv} and γ_u. In reference to the choice we may say that it is mainly determined by reasons over and above the algorithm: the choice of description space of objects from $\boldsymbol{X}$, the desire to assign some meaningful interpretation to numbers a_{iv} and γ_u, means of formalization of the notion of object likeness (step 1.2), etc. Irrespective of the choice (within permissible limits) it is possible to prove that the following assertion takes place.

<u>Assertion</u>. *Each algorithm* $\boldsymbol{A}$ *solves the problem and* $\boldsymbol{A} \succ \boldsymbol{A}_0$.

Proof. Let's fix an arbitrary algorithm $\boldsymbol{A}$. In other words vectors: $(a_{im+1},\ldots,a_{im+n_u})$ and γ_u, here $u \in \{1,\ldots,k\}$,

$$i \in \{1,\ldots,l\}, m = \sum_{v=1}^{u-1} n_v .$$

To prove the first part it is necessary to make sure that

$$\forall x \in X\ \forall i \in \{1,\ldots,l\}(0 \leq \mu_A^i(x) \leq 1).$$

The left part of the inequality follows from the definition $\mu_A^{i,u}(x,x')$ (step 1.2) and the right one follows from the evident inequalities

$$(\sum_{v\in I_u}(-1)^t \cdot a_{iv}) \cdot (\sum_{v\in I_u} a_{iv})^{-1} \leq 1,\ (\sum_{u=1}^k \gamma_u \cdot \mu_A^{i,u}(x)) \times$$
$$\times (\sum_{u=1}^k (\gamma_u \cdot \sum_{v\in I_u} a_{iv}))^{-1} \leq 1,$$

holding true irrespective of the choice of parameters of algorithm $\boldsymbol{A}$ (within permissible limits) and irrespective of t.

To prove the second part let's assume the rule of contraries, i.e.

$$\exists i\, \exists x \in \boldsymbol{X}\ (\mu_{A_0}^i(x) > \mu_A^i(x)).$$

But, in accordance with the definition of $\boldsymbol{A}_0$ this means, see (4), that $\mu_{A_0}^i(x) = 1$ and, consequently, $x \in \boldsymbol{X}_i^0$. Then, from the construction of $\boldsymbol{A}$ we immediately obtain that for all u the following takes place

$$\mu_A^{i,u}(x,x) = \mathbf{max}\{0,((\sum_{v\in I_u} a_{iv}) \cdot (\sum_{v\in I_u} a_{iv})^{-1})\} = 1 \Rightarrow \mu_A^{i,u}(x) = 1$$

and irrespective of the choice of non-negative γ_u the following holds true

$$\mu_A^i(x)=1.$$

But this conflicts with the assumption and, consequently, $\boldsymbol{A}\succ\boldsymbol{A}_0$ .In view of arbitrary choice of $\boldsymbol{A}$ the assertion can be considered completely proven.■

3.3. Determining parameters and some properties of *A*

As it was stated above, algorithm $\boldsymbol{A}$ and values $\mu_A^1(x),\ldots,\mu_A^l(x)$ calculated by it greatly depend on the choice of parameters. It is clear, that this choice should be such that the results of the operation of the algorithm we may interpret in terms of the problem (item 6 of **restrictions**), analyze the initial information and better understand the problem. Moreover, from the scheme of constructing $\boldsymbol{A}$ it is evident that in the set of algorithms under consideration the question of complexity of realization is of importance because, in accordance with the methodology, $\boldsymbol{A}$ is based on the complete exhaustion of objects from $\boldsymbol{X}_i^0$. We will show below that in the chosen way of determining parameters instead of $\boldsymbol{X}_i^0$ we may restrict ourselves to objects from $\boldsymbol{X}_{ui}^0$ not only in the scheme of constructing $\boldsymbol{A}$.

Let's first describe an algorithm of parameters determination. In so doing we consider that samples $\boldsymbol{X}_{ui}^0\subset\mathbf{B}_2^{n_u}$ for all i,u are constructed by reduction of the corresponding set of rules S_{ui} to DNF.

Algorithm.

Step 1. For the specified I_u and sets of objects $\boldsymbol{X}_{ui}^0$ we perform the following sequence of steps.

Step 1.1 We fix the number of the feature $v\in I_u$ and for each $i\in\{1,\ldots,l\}$ we calculate

$$b_{iv}=(\sum_{x_t\in\boldsymbol{X}_{ui}^0}x_{tv})\cdot(|\boldsymbol{X}_{ui}^0|)^{-1}$$

Step 1.2 If all features are not exhausted, we return to *step 1.1*. Otherwise, we pass to *step 2*.

Step 2. For all $v\in I_u$, $i\in\{1,\ldots,l\}$ we calculate

$$b_v=(\sum_{i=1}^{l}b_{iv})\cdot l^{-1},\, a_{iv}=|b_{iv}-b_v|,\;\; \gamma_u=\min_{i\in\{1,\ldots l\}}\max_{v\in I_u}\{a_{iv}\}.$$

Step 3. End of work algorithm.

Now let's consider some properties of an algorithm for parameters construction and relation of the parameters with properties of dominating algorithms $\boldsymbol{A}$. We restrict our attention only to those algorithms that relate to the question of applicability. Let's first check that the formed parameters can be used. But it is evident because the following property can be observed.

Property 1. *The formed parameters are permissible, i.e.* $a_{iv}\ge 0$ *and* $\gamma_u\ge 0$ *for all* $i\in\{1,\ldots,l\}$, $u\in\{1,\ldots,k\}, v\in I_u$.

Now let's discuss the following aspect. Obviously, when constructing $\boldsymbol{X}_i^0$ by reducing formula $S_{1i}(x)\wedge..\wedge S_{ki}(x)$, see (4), to DNF we will obtain the same set of objects of space $\mathbf{B}_2^n$ as compared with direct summation $\boldsymbol{X}_{1i}^0\oplus..\oplus\boldsymbol{X}_{ki}^0$ of samples $\boldsymbol{X}_{ui}^0$ of the corresponding DNF of rule S_{ui} , i.e. sample $\boldsymbol{X}_i^0$ differs in the set of features I_u in the number of similar members of the sample $\boldsymbol{X}_{ui}^0$. The number of such members can be easily calculated and it is rather large. But there arises a question: is it possible when constructing parameters to restrict ourselves to the sample $\boldsymbol{X}_{ui}^0$ or it is necessary to take into consideration the suitable part of the sample $\boldsymbol{X}_i^0$? In the latter case this would lead to exponential growth of the number of operations, memory capacity and impossibility to use algorithm $\boldsymbol{A}$ (see item 7 of **restrictions**). But, it appears that we can answer in the affirmative concerning the first part of the formulated question because the following property can be observed.

Property 2. *Parameters* a_{iv} *and* γ_u *for samples* $\bigcup_{t=1}^{t_k}\boldsymbol{X}_{uk}^0\,(k=1,\ldots,l)$ *do not depend on* $t_k\in\mathbf{N}$.

Now let's consider some properties of parameters a_{iv} the value of which depends on different nature of rules S_{ui} and, consequently, of samples $\boldsymbol{X}_{ui}^0$. Intuitively it is clear that the greater the difference among the rules the greater is the difference in parameters a_{iv} and the greater is the relative weight of the set of features I_u. We will assign a formal meaning to this property by proving the validity of its negation.

Property 3. *If* $\boldsymbol{X}_{u1}^0=..=\boldsymbol{X}_{ul}^0$ *for some* $u\in\{1,\ldots,k\}$, *then* $\gamma_u=0$ *and* $a_{iv}=0$ *for all* $i\in\{1,\ldots,l\}, v\in I_u$.

It is easy to see that the corresponding features I_u do not in any way influence the formation of the assessment $\mu_A^i(x)$. And this, on the whole, agrees with intuitive knowledge of the operation of $\boldsymbol{A}$. But it should also alert because when calculating $\mu_A^i(x)$ the sum of such values is in the denominator. But it turns out that equality to zero of parameters a_{iv} by the condition $\boldsymbol{X}_{u1}^0=..=\boldsymbol{X}_{ul}^0$ is the only one because the following property can be observed.

Property 4. *From* $a_{iv}=0$ *for all* $v\in I_u$, $\boldsymbol{X}_{u1}^0=..=\boldsymbol{X}_{ul}^0$ *follows*.

The proof of this property can be easily obtained by the rule of contraries if we take into account that

$$\forall v\in I_u\;(\sum_{i=1}^{l}\mathbf{sign}(b_{iv}-b_v)\cdot a_{iv}=0).$$

Then from the condition $a_{iv}=0$ for all $v\in I_u$ we obtain:

$$\boldsymbol{X}_{ui}^0=\bigcup_{j=1}^{l}\boldsymbol{X}_{uj}^0$$

and at least for two different $j_1,j_2{:}\boldsymbol{X}_{uj_1}^0\ne\boldsymbol{X}_{uj_2}^0$ (by the assumption). Now notice that:

$$\forall u\,\forall j_3,j_4\in\{1,\ldots,l\}\,(\boldsymbol{X}_{uj_3}^0\subset\boldsymbol{X}_{uj_4}^0\Rightarrow b_{j_3v}\le b_{j_4v}),$$

and

$$\min_j b_{jv} < b_v < \max_j b_{jv}$$

Thus, we have obtained contradiction with the assumption.■

Let's also note that when rules coincide a decision-making problem does not exist. That is why we can leave a case when $a_{iv} = 0$ for all $v \in I_u$ from consideration.

We can formulate another set of properties of this type. But, we did not aim at fully describing and studying the introduced set of dominating algorithms. We mainly based on the following reasoning. If the behavior of the algorithm is substantiated by the proof of some understandable properties, such algorithms have the right to exist. Eventually, practical application is determined by results of experiments, to have only proofs is obviously insufficient. That is why subsequent study of algorithms we leave for further publications and we now turn to a short description of the system and results of its testing.

4. CONCLUSION

So, we have completely proved that algorithms $\boldsymbol{A}$ solve any problem of choice. As this takes place, each $\boldsymbol{A}$ is equivalent to $\boldsymbol{A}_0$. Hence it follows that on other problems as well these algorithms behave somewhat correctly. At least they satisfy the principle of correctness on $\mathcal{X}^o$ and the are monotonous on $\mathcal{X} \setminus \mathcal{X}^o$. In view of generality of statements of all problems, the algorithm itself, as a scheme for processing the source information into a result, from the point of view of applicability is beyond doubt (possibly not only for the considered problems of choice). And it will suffice in the context of aims sought in the present paper.

But information aspects remain. It appears that in this direction it is also possible to make certain conclusions based on results mentioned above. Main conclusions relate to means of construction and structure of $\mathcal{X}^o$. It is quite obvious that main difficulties when solving problems of different nature are associated with the choice of appropriate means for coding information. If a problem has a solution, there exists a finite coding and a finite sampling for which construction of an algorithm of $\boldsymbol{A}_0$ type is purely of technical nature. It is possible to determine parameters of this coding, the structure of set $\mathcal{X}^o$, etc. This can be done by experiments which are an integral part of problems of pattern recognition.

The algorithms, described in this article, were used when developing decision support systems in orthopedics [10].

5. ACKNOWLEDGMENTS

This work was supported by a grant F10P-097 from Belarusian Republican Foundation for Fundamental Research.

6. REFERENCES

[1] Nilsson, N.J. *Learning Machines*, New York, McGraw-Hill, 1965.

[2] Krasnoproshin, V. and Obraztsov V. Fuzzy Algorithms in Decision-Making and Management Problems. *Proc. of the 2nd Congress SIGEF*, S. De Compostela, 1995, v.2, 68-79.

[3] Polya, G. *Mathematical Methods in Science*, The Mathematical Association of America, Washington, 1977.

[4] Klenne, S.C. *Introduction in the Metamathematics*, Van Nostrand, Princeton and North-Holland, Amsterdam, 1951.

[5] Godel, K. Uber formal unentscheidbare Satze der Principia Mathematica und vervandter System I. Monatshefte Math. Phys. 38, 1931, 173-198.

[6] Barwise, J. (ed) *Handbook of Mathematical Logic*, North-Holland, Amsterdam, 1977.

[7] Thayse, A. and Gribomont P. etc. *Approche logique de l'intelligence artificielle. 1 De la logique classique a la programmation logique*, Bordas, Paris, 1988.

[8] Lauriere, J.-L. *Intelligence Artificielle. Resolution de Problemes par l'Homme et la Machine*, Eyrolles, Paris, 1987.

[9] Pyatnitsyn B.N. *Towards a Problem of Induction-Deduction Relations. "Methods of Logical Analysis"*, Nauka, Moscow, 1977.

[10] Krasnoproshin V., Obraztsov V., Vissia H. *Decision-Making by Precedence: Modeling, Technology and Applications*, Proceedings of International Conference on Modeling and Simulation in Technical and Social Sciences (MS'2002), Girona, Spain, 25-27 June 2002, - p.p. 267-277.