SOVIET CLASSICS
IN PATTERN RECOGNITION AND IMAGE ANALYSIS

An Algebraic Approach to Recognition or Classifications Problems*

Yu. I. Zhuravlev
Computer Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 117967 Russia
e-mail: zhuravlev@ccas.ru

This paper written be the full member of the Russian Academy of Sciences dwells upon the substantiation, developments, and investigation of the mathematical model of pattern recognition algorithm which covers the known classes of pattern recognition algorithms. The article was published in the collection of papers *Problemy Kibernetiki* (Problems of Cybernetics), edited by S.V. Yablonskii. The ideas, problem set-up, and the results of this work provide a basis for evolution of soviet and russian pattern recognition school for next 20 years. The essential fundamental and application results were made on this base in the field of recognition and in several adjacent domains (image analysis and understanding, speech analysis and synthesis, information technologies for data analysis and evaluation).

Unfortunately, this work is not available in English and nowadays the attempts have been made to re-open the results presented in the paper.

The editorial board hopes that this publication will allow the world commonwealth of pattern recognition scientists to get acquainted with the fundamentals of mathematical theory of pattern recognition from the original sourse.

Deputy Editor-in-Chief
Dr.-Eng. Igor B. Gourevitch

INTRODUCTION

The paper discusses the well-known problem of assigning an object to certain classes from a specified list of classes.

The problem can be formulated as follows. We have a set M of objects which are to be classified, and we know that M can be represented by the sum of subsets $K_1, \dots, K_l$ usually called classes.

Given an information I about the classes $K_1, \dots, K_l$, the description of the set M, and the description $I(S)$ of an object S about which it is generally not known to what classes $K_1, \dots, K_l$ it belongs, we need to evaluate for every j the property $S \in K_j$ $(j = 1, 2, \dots, l)$ from the information I and the description $I(S)$.

The problem formulated above is usually called the pattern recognition problem or the classification problem. In what follows, we call it the Z problem or the recognition problem.

At first, problems of the Z type were regarded as purely applied tasks. A special emphasis was laid, for example, on automatic text reading. Each character was assigned the class K of its various images.

It was required to establish to which of the classes K the presented image (which was assumed in advance to be the image of a character) belonged. In other words, it was required to identify a character by its image. Later came similar tasks concerned with speech identification and the like.

Somewhat later, investigators turned to classification tasks involving objects described in a more complicated way. These included medical diagnostics, geological forecasting, assessment of economic and political situations, prediction of properties of chemical compounds, and so on.

There is no need to discuss these problems in detail. There is an extensive literature on each of them and, more important, there is no significant difference between them mathematically. Note only that recognition problems are rather general, a great many practical issues can be reduced to them. For example, with some idealization, any decision-making problem can be reduced to a recognition scheme if the decision-making process is mainly based on an analysis of past experience. To demonstrate, let there be a collection of previously analyzed situations $S_1, \dots, S_m$ and their descriptions $I(S_1), \dots, I(S_m)$ are given in some form. The best decision is known for each situation S_i and this decision, $R(S_i)$ is likewise described. Suppose also that a description of the set $\{R\}$ of all possible decisions may be given. Assume that in the space $\{R\}$ there exists the concept of closeness; i.e., the decisions may be divided into classes $K_1, \dots, K_l$ so that "close" decisions (decisions of the same type) fall in the same class. Decisions in different classes are not close.

Given the description of a new situation, $I(S)$, we need to find the class of the best decision from the collection of descriptions of previously analyzed situations, $I(S_1), \dots, I(S_m)$ and of their respective decisions $R(S_1), \dots, R(S_m)$.

Thus, the recognition problem is, in a special case, a discrete equivalent of the problem of searching for optimal decisions. Recognition problems encompass not only those concerned with the synthesis of best decisions but also other important classes of applied problems.

Problemy Kibernetiki, 1978, no 33, pp. 5–68.

Received October 20, 1997

So, the first reason why a great number of investigators have turned to classification problems in the past decade is abundance of applied issues whose study reduces to solving problems of this type. Another reason, very important to mathematicians, is that the solution of these problems has placed in circulation a great number of ill-defined or what are usually called heuristic algorithms. The point is that the overwhelming majority of the recognition theory applications relates to poorly formalizable sciences and industries, such as medicine, geology, sociology, or chemistry.

It is difficult to develop formal theories and to use standard mathematical methods in these areas. A mathematical format can be given to some intuitive principles and then the resultant empirical formalisms applied to special types of problems at best. This was the reason why a great number of various methods and algorithms emerged at the early stage in the evolution of recognition theory and practice and were applied to practical problems without any serious mathematical basis. As is customary in all experimental sciences, the methods were verified by a direct test—success or failure in tackling real problems. Many of them have stood this test and are used despite the lack of mathematical justification.

It must be admitted that ill-defined algorithms has long become a fact of life. Apparently, any algorithm of this type can be regarded as an experiment and the total set of these experiments and their findings should be treated as a set of objects new to mathematics. So, having recognized the existence and practical usefulness of ill-defined procedures of solving poorly formalized problems as reality, we face the task of studying the very set of these procedures using rigorous mathematical methods. An antithesis to this ideology might be to build formal models in areas that now defy formalization. However, this path can hardly lead to success in, say, descriptive biology or geology.

Consider the collection of ill-defined algorithms intended to handle recognition problems. As they were accumulated, not only individual algorithms but their underlying principles were described. These principles, now operating on subsets of algorithms and, likewise, poorly formalized at first, were later given (or might have been given) exact mathematical descriptions. The choice of a principle only was heuristic at that stage but the algorithms produced by this principle could be formulated in a standard manner. Thus, the formalization of different principles led to models of recognition algorithms.

Consider a simple example of a principle formulated intuitively. In many problems where object descriptions are defined by sets of values of numerical features (the objects are points in an n-dimensional space), the descriptions belonging to different classes can be separated by surfaces of a fairly simple form. We call this approach the separation principle.

One of the possible formalizations is this. Consider hyperplanes, the simplest class of separating surfaces:

$$\sum_{i=1}^{n} a_i x_i + a_{n+1} = 0.$$

Let the set of admissible objects be separated into two classes: K_1, K_2, $K_1 \cap K_2 = \varnothing$. Let it also be known that the objects $S_1, \ldots, S_m$ belong to K_1 and the objects $S_{m+1}, \ldots, S_q$ to K_2. Generally, these objects are not equivalent. Therefore, we introduce their numerical characteristics $\gamma(S_i) = \gamma_i$ which is the weight of the object S_i, $i = 1, 2, \ldots, m, m+1, \ldots, q$. So, the set of algorithms is characterized by specifying the parameters $a_1, \ldots, a_{n+1}$ which are coefficients in the hyperplane equation, and $\gamma_1, \ldots, \gamma_q$, which are the weights of the objects classified above. The recognition process for $I(S) = (\alpha_1, \ldots, \alpha_n)$ proceeds as follows.

Suppose

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i + a_{n+1}.$$

We separate the objects $S_1, \ldots, S_m$ into sets K_1^+ and K_1^- such that $S_i \in K_1^+$ if $f(I(S_i)) \geq 0$ and $S_i \in K_1^-$ if $f(I(S_i)) < 0$. Similarly, we separate the objects $S_{m+1}, \ldots, S_q$ into sets K_2^+ and K_2^-. Consider the quantities

$$\gamma(K_1^+) = \sum_{S_i \in K_1^+} \gamma(S_i), \quad \gamma(K_1^-) = \sum_{S_j \in K_1^-} \gamma(S_j)$$

and similar quantities $\gamma(K_2^+)$ and $\gamma(K_2^-)$.

Evaluate $f(I(S))$. We compare to S two numbers, $\Gamma_1(S)$ and $\Gamma_2(S)$, which are, respectively, the values of the membership function of S in K_1 or K_2. If $f(I(S)) \geq 0$, then

$$\Gamma_1(S) = \frac{\gamma(K_1^+) + \gamma(K_2^-)}{\gamma(K_1^-) + \gamma(K_2^+)}, \quad \Gamma_2(S) = \frac{\gamma(K_2^+) + \gamma(K_1^-)}{\gamma(K_1^+) + \gamma(K_2^-)}.$$

For $f(I(S)) < 0$, $\Gamma_1(S) = \dfrac{\gamma(K_1^-) + \gamma(K_2^+)}{\gamma(K_1^+) + \gamma(K_2^-)}$, and, similarly, $\Gamma_2(S)$.

By the numbers $\Gamma_1(S)$ and $\Gamma_2(S)$, the decision is made to include S in K_1 or K_2. This procedure is specified by a decision rule. Consider the class of decision rules defined by the parameter $\delta \geq 0$:

if $\Gamma_1(S) - \Gamma_2(S) > \delta$, then $S \in K_1$,

if $\Gamma_2(S) - \Gamma_1(S) > \delta$, then $S \in K_2$,

if $|\Gamma_1(S) - \Gamma_2(S)| \leq \delta$, then no decision is made because the algorithm has refused to classify S.

We have built one of the possible models based on the separation principle. This model draws upon two hypotheses: (a) the elements (or at least a significant part of the elements whose classification is of interest) in the classes K_1 and K_2 are separated by a hyperplane, and (b) the elements of the classes are not equal in importance, and this importance can be expressed numerically.

The hypotheses were embodied in the model

$$\mathfrak{M}(a_1, \ldots, a_{n+1}, \gamma_1, \ldots, \gamma_q, \delta),$$
$$-\infty < \gamma_i, \quad a_i < +\infty, \quad \delta \geq 0.$$

Values of all the parameters of a model are defined by its element—a specific recognition algorithm.

The second stage in the evolution of recognition theory was marked above all by changeover from individual algorithms to models, i.e., families of classification algorithms, used to solve classification problems. Several types of models have been developed and tested by now. We focus on several models that are the most common in various applications.

1. Models based on the separation principle (*R* models) [55, 56, 61, 76]. Such a model was examined in the above example. The *R* models mainly differ in the specification of the class of surfaces from which we select a surface (or a set of surfaces) that separates elements of different classes.

2. Statistical models. They are formed on the basis of mathematical statistics. These are used mainly in the cases where probabilistic characteristics, such as distribution functions, of classes $K_1, \ldots, K_l$ are known (or are simple to find). Since the initial information in most problems does not make it possible to determine these characteristics reliably enough, it is natural to regard these models likewise as based on heuristic principles.

3. Models based on the principle of potentials (*P* models [4]). They draw upon the analogy with a well-known physical principle: the force of attraction between any two masses is directly proportional to the product of the masses and inversely proportional to the distance between them.

Concerning the objects whose membership in a class K_j has been established previously, it is possible in one way or another to introduce the concept of the mass of that set and of the distance from the set to the object S to be recognized and to choose as the value of the membership function of S in K_j the quantity which is a monotonically increasing function of mass or a monotonically decreasing function of distance. *P* models can be built in more than one way.

4. Estimation (voting) models (*V* models) [27, 28, 31]. They are based on the principle of partial precedence. Corresponding parts of the descriptions of previously classified objects and of the object to be recognized are analyzed for closeness. The existence of closeness is a partial precedence and is estimated by a defined rule.

The overall estimate of the object with respect to a class is formed from the set of estimates and it is a value of the membership function of the object in the class.

A model based on one of the above principles can use ideas of other principles. For example, the *R* model set up above also uses the principle of potentials. The latter [4] directly combines the principle of potentials and the separation principle.

The committee method [55, 56, 61, 76] uses ideas of the separation principle and of the voting principle.

At the time when the focus was on specific dedicated algorithms emphasis was laid on the construction of efficient computational schemes and experiments; that is, on applied problems. Since the switch to models, a multitude of new problems of interest to mathematicians have arisen.

These include, first of all, the synthesis of algorithms extremal in terms of recognition quality within a given model. The quality functional of an algorithm can be defined in several ways. Its definition is usually based on the following principle. The manner is specified in which objects from each class are built. A fixed algorithm from the given model is estimated in terms of the percentage of the objects it classifies correctly, i.e., assigns them to a given class. The quantity thus obtained is averaged over the classes and referred to as the quality functional of the algorithm. The objective is to find in that model an algorithm that has a maximum quality functional. For example, the following law of the generation of the classes K_1, K_2 can be specified. Let the descriptions $I(S)$ of the objects S be collections $(a_1(S), \ldots, a_n(S))$ of numerical features such that $-\infty < a_i(S) < +\infty$ $(i = 1, 2, \ldots, n)$.

Two normal distributions are defined in the n-dimensional space with means m_1 and m_2 and with variances σ_1 and σ_2. Points (object descriptions) are picked at random and the class to which they are assigned is drawn according to the specified laws. Then, if it is assigned to, say, K_1 with probability p, the object S is included in the learning sample and, with probability $1 - p$, it is included in the control sample. The same procedure is applied to objects in K_2. We have thus formed a learning and a control samples. The former includes objects $S_{11}, \ldots, S_{1m}$ from K_1 and objects $S_{21}, \ldots, S_{2t}$ from K_2 and the latter objects $S_{31}, \ldots, S_{3v}$ from K_1, and objects $S_{41}, \ldots, S_{4u}$ from K_2. An algorithm A is developed in the model, which, given the descriptions $I(S_{11}), \ldots, I(S_{1m})$ and $I(S_{21}), \ldots, I(S_{2t})$, returns the maximum value of the quality functional $\varphi(A) = \frac{q'}{q''}$, where q' is the number of the control sample's objects correctly classified by the algorithm A, and $q'' = v + u$ is the total number of objects in the control sample.

The quantity $\varphi(A)$ is a random variable, and its characteristics (instants) give an idea about the accuracy of the model for a certain type of recognition problems. It is not at all trivial to calculate these characteristics. The results in such problems can be only obtained for relatively simple models and class formation laws (see, for example, [67, 69]).

A more standard approach is that where, given an initial fixed information I_0 and a model, it is required to find an algorithm in the model to classify as accurately as possible the given collection S_i $(i = 1, 2, \ldots, m)$ of control objects whose membership in $K_1, \ldots, K_l$ is unknown.

Naturally, the information of the type $S_i \,\overline{\in}\, K_j$ and $S_i \in K_j$ is not included in an algorithm. Extremum algorithms constructed in a model on a specified control sample produce new types of extremum problems to be solved and studied. These studies were the subject of a great number of papers, especially on R and V models. Here is one example. Let there be given the descriptions $I(S_1), \ldots, I(S_m)$ of objects in K_1 and $I(S_{m+1}), \ldots, I(S_q)$ of objects in K_2, such that $I(S_i) = (\alpha_{i1}, \ldots, \alpha_{il})$. There is no initial information available. An R model is built, and the partition is done by the hyperplane $f(\vec{x}) = \sum_{i=1}^{n} a_i x_i + a_{n+1}$. The model parameters are the coefficients $a_1, \ldots, a_{n+1}$ of the hyperplane.

Decision rule. If $f(I(S_i)) \geq 0$, then $S_i \in K_1$. For $f(I(S_i)) < 0$, the object S_i $(i = 1, 2, \ldots, q)$ is included in K_2.

It is easy to write for every S_i the condition of correct classification. Having written these conditions sequentially for $S_1, \ldots, S_m, S_{m+1}, \ldots, S_q$ we get a system of linear inequalities in the unknowns $a_1, \ldots, a_{n+1}$

$$\begin{gathered} a_1\alpha_{11} + \ldots + a_n\alpha_{1n} + a_{n+1} \geq 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_1\alpha_{m1} + \ldots + a_n\alpha_{mn} + a_{n+1} \geq 0, \\ a_1\alpha_{m+1,1} + \ldots + a_n\alpha_{m+1,n} + a_{n+1} < 0, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_1\alpha_{q1} + \ldots + a_n\alpha_{qn} + a_{n+1} < 0. \end{gathered} \tag{1}$$

The system in (1) is inconsistent, generally speaking. To set up the desired algorithm, it is required to find a maximum consistent subsystem in (1). By solving this problem, we can obtain the values of $a_1, \ldots, a_{n+1}$ and, thus, an algorithm which is extremal on the sample $S_1, \ldots, S_q$.

The selection of a maximum consistent subsystem is a difficult problem even for linear systems, and special methods are needed to solve it. In more sophisticated models, the development of algorithms extremal on a given sample likewise involves searching for maximum consistent subsystems, but inequalities in analogues to (1) are not linear.

Finally, it is interesting by itself in many cases to construct a convenient model description and to solve problems inherent in the model and not related directly to the synthesis and analysis of extremum algorithms. Thus, in models of estimates $\Gamma_j(S)$ calculation, it is often unfeasible to try to find these quantities by direct exhaustive search of object descriptions for close parts. The need, therefore, arose to develop a special "analytical" tool whereby the quantities $\Gamma_j(S)$, could be evaluated efficiently.

Another branch of studies evolved in parallel to the changeover from individual algorithms to models. It was primarily intended to facilitate the preparation of learning information for recognition problems. Successful attempts were made in this area to extend the class of learning informations suitable for further processing by recognition algorithms. This branch came to be known as structural recognition [78].

Previously, it was required for every class K_j $(j = 1, 2, \ldots, l)$ to draw up a list of objects known in advance to belong or not to belong to K_j, to select a set of characteristics (features), and to give descriptions of the objects thus prepared as collections of features. It took a good deal of time and effort to construct a sufficient number of template objects and their descriptions. It was required in fact to define classes by a partial enumeration of their elements. On the other hand, discrete mathematics (algebra, mathematical logic, theory of algorithms, etc.) had long been using another method to define sets. Select a system of basis elements $\{B\}$ and a system of operations $\{O\}$. Apply sequentially operations in $\{O\}$ to basis elements in $\{B\}$, i.e., use a standard procedure to generate a set in succession. The class K_j is defined for recognition problems by the collection $\{B\}_j$ of basis elements and by the collection $\{O\}$ of generation rules (operations).

The collection of features (properties) is constructed in a similar way. A set of basis properties (elementary predicates), not necessarily binary, is defined. A set of predicates is developed from them inductively, e.g. using logical operations, quantifiers, etc., and it is these predicates that are the features used to describe the objects. By analogy with formulas of formal theory, these predicates are denoted by $\{\Phi\}_j$. The quantities $\Phi(S)$ are evaluated on the constructions of S and on the generated elements in K_j (actually, a finite subclass is generated in K_j in a short time).

The collection of elementary properties describing the relation of S to K_j is selected. For example, $(S \in K_j)$; it is more probable for S to belong to K_u than to K_t; the membership function (according to Zadeh [79]) of S in K_j is not greater than ε, etc. The class of formulas, $\{\Psi\}_j$, is constructed over the collection of elementary properties by analogy with $\{\Phi\}_j$.

Finally, we consider the formulas $\Phi \longrightarrow \Psi$, $\Psi \in \{\Psi\}_j$, and $\Phi \in \{\Phi\}_j$, i.e., conclusions in terms of $\{\Psi\}_j$ from the satisfied Φ (it is possible to consider necessary conditions: $\Psi \longrightarrow \Phi$).

The collection of the formulas $\Phi \longrightarrow \Psi$ with the satisfied Φ is estimated numerically and is used as a basis for the numerical quantity $\Gamma_j(S)$ (explicitly or implicitly), which is the membership function of S in K_j $(j = 1, 2, \ldots, l)$. The decision is made on the basis of the vector $(\Gamma_1(S), \ldots, \Gamma_l(S))$ or the matrix $\{\Gamma_j(S_i)\}$ $(i = 1, 2, \ldots, q)$ as to the class to which S or each of the objects $S_1, \ldots, S_q$ belongs.

The structural approach has inherent problems and problems related to applications.

We do not dwell on these problems in this paper. Here are only some particular results to demonstrate that the structural approach in recognition does not differ fundamentally from the approach whereby classes are defined by enumerating descriptions of a finite number of objects.

To sum up the foregoing, we should note that the theory and practice of recognition have accumulated a wealth of experimental material (various algorithms) and a certain experience in the construction and analysis of models, i.e., descriptions of classes of recognition algorithms. Taking the above history as the basis, it is possible to set the task of developing a general theory of recognition algorithms. Mention should be made of an interesting paper by Grenander [77] in this connection. The present paper develops a different approach to the formulation of this general theory.

An alternative definition of the recognition algorithm is proposed, which can accommodate all existing types of algorithms. The terms in which the definitions are given make it possible to use the latter both in theoretical studies and in applied problems.

Furthermore, an approach is proposed, similar to the one set forth in connection with structural methods, to provide an efficient analysis of the class of recognition algorithms and of their constructive description. Basic algorithms and models are selected and operations on them are introduced to generate subsequently new algorithms and models. This approach can be used not only to describe efficiently many new classes of recognition algorithms but also to identify some of their important properties. For example, conditions were derived under which a given family of algorithms is basic with respect to the operations thus introduced. Two models were taken as examples to demonstrate the test of these conditions. It was found what properties a model should have to contain an algorithm that can classify correctly all objects of this finite sample. Methods were given whereby these algorithms can be developed for various models.

Here is a brief outline of the paper by chapters.

The first chapter introduces basic definitions and the statement of problems with an emphasis laid on the concept of initial learning information.

The second chapter briefly describes the R models and P models and, in more detail, the V (estimates calculation) model.

Taking the latter as an example, the transition from specific algorithms to an intuitively formulated principle and finally to a formal description of a model is shown in detail.

Efficient analytical tools to calculate estimates $\Gamma_j(S)$ of the classified objects S are described. This chapter also discusses models based on the structural approach.

The third chapter analyzes specific models and proposes a definition of the recognition algorithm. This algorithm A is regarded as the consecutive application of two operators, R_A and r_A, such that $A = R_A \cdot r_A$. In what follows, R_A is called the recognition operator and is applied to the initial (learning) information I_0 and to the descriptions $I(S_1^{\prime})$, …, $I(S_q^{\prime})$ of the objects to be recognized. It transforms them into a numerical matrix, $\{a_{ij}\}_{q \times l}$, where a_{ij} is the value of the membership function of S_i in class K_j ($j = 1, 2, \ldots, l$, $i = 1, 2, \ldots, q$). The term "membership function" was taken from the theory of fuzzy sets [79].

We use a shorter term, "estimate," instead, in many cases. We believe that it reflects better the meaning of a_{ij}.

The operator r_A—the decision rule of A—transforms the matrix $\{\alpha_{ij}\}_{q \times l}$ into the information matrix $\{\alpha_{ij}\}_{q \times l}$, $\alpha_{ij} \in \{0, 1, \Delta\}$. For $\alpha_{ij} \in \{0, 1\}$, the algorithm A states that $P_j(S_i^{\prime}) = \alpha_{ij}$, $\alpha_{ij} = \Delta$ mean that A failed to evaluate the property $P_j(S_i^{\prime})$.

Operations of addition and multiplication by a scalar are introduced for the recognition operators R_A (and, hence, for algorithms with a fixed r_A). The latter form a linear vector space in these operations. Given a control sample whose size is bounded from above, this space has a finite basis under fairly broad assumptions about the initial information and possible descriptions of control objects. In addition, the product (consecutive application of recognition operators) is introduced. The same chapter proves a theorem that establishes the properties of models, which enable any finite control sample to be classified without error. The theorems thus proved are used to analyze linear closures of the R models and V models.

The fourth chapter reviews operations on algorithms at the level of information matrices. Subgroups in these operations are described. More general types of operations are also introduced, which are interpreted as multiplace functions of ternary logic. The application of these operations to the synthesis of new recognition-algorithm models is demonstrated.

The fifth chapter describes basic types of problems that arise in the synthesis of extremum algorithms and the principles are set forth for their solution.

CHAPTER I
STATEMENT OF THE PROBLEM. BASIC DEFINITIONS. LEARNING INFORMATION. QUALITY FUNCTIONAL

§ 1. Statement of the Problem

Let there be given a set M of what is called below admissible objects. The set of admissible objects is covered by a finite number of subsets, K_1, …, K_l: $M = \bigcup_{i=1}^{l} K_i$. The subsets K_j ($j = 1, 2, \ldots, l$) are called classes. The partition of M is not defined completely. Only a certain information, I_0, about K_1, …, K_l, is given. Similarly, an admissible object S is defined by values of certain characteristics. The collection of the specified values defines the description $I(S)$ of S.

The prime problem (or the Z problem) is to use the information I_0 about the classes K_1, …, K_l, $I_0(K_1, \ldots, K_l)$ and the description $I(S)$ of the admissible object S in order to evaluate the predicates $P_j(S)$ implying "$S \in K_j$" ($j = 1, 2, \ldots, l$). It is customary to call I_0 learning information and $P_j(S)$ elementary predicates.

The restatement of Z is as follows. Let there be given two sets, $\{I_0\}$ and $\{I(S)\}$. The former is the set of admissible learning informations, $I_0(K_1, \ldots, K_l)$, and the latter is the set of descriptions $I(S)$ of the admissible $S \in M$.

It is required to construct an algorithm A such that

$$A(I_0(K_1, \ldots, K_l), I(S)) = (\alpha_1^A(S) \ldots \alpha_l^A(S)), \quad (2)$$

where

$$\alpha_i^A(S) = P_j(S), \quad j = 1, 2, \ldots, l,$$

or, as applied to the collection of admissible objects,

$$A(I_0(K_1, \ldots, K_l), I(S'_1), \ldots, I(S'_q)) = \left\{\alpha_j^A(S'_i)\right\}_{q \times l}, \quad (3)$$

$$\alpha_j^A(S'_i) = P_j(S'_i), \quad i = 1, 2, \ldots, q, \quad j = 1, 2, \ldots, l.$$

The solution to the Z problem also involves ill-defined algorithms A such that $\alpha_j^A(S'_i) \in \{0, 1\}$ but not necessarily $\alpha_j^A(S'_i) = P_j(S'_i)$. Finally, algorithms are considered which refuse to evaluate $P_j(S'_i)$, $1 \le j \le l$ for some admissible S'_i. This fact is written in symbols as $\alpha_j^A(S'_i) = \Delta$.

When the Z problem is solved in the class of ill-defined algorithms, it is usual to impose additional constraints on the algorithm A.

Let the set $\{A\}$ of algorithms be specified, such that

$$A(I_0(K_1, \ldots, K_l), I(S)) = (\beta_1^A(S), \ldots, \beta_l^A(S)), \quad (4)$$
$$\beta_j^A(S) \in \{0, 1, \Delta\}.$$

The numerical functional $\varphi(A)$ is defined on the set $\{A\}$ of these algorithms. It is called the quality functional of the algorithm A.

The refined prime problem ($\tilde{Z}$ problem) is as follows: Among the algorithms defined by (4), find an algorithm A^* such that

$$\varphi(A^*) = \sup_{A \in \{A\}} \varphi(A). \quad (5)$$

For a complete formulation of the problem, it is necessary to refine the concepts of:

1. The learning information $I_0(K_1, \ldots, K_l)$;
2. The description $I(S)$ of the admissible object S;
3. The quality functional $\varphi(A)$ of the algorithm A.

§ 2. The Learning Information $I_0(K_1, \ldots, K_l)$. Descriptions of Admissible Objects

I. The Standard Types of Learning Information and Descriptions

1. Let a collection of features $\{1, 2, \ldots, n\}$ be set. Each feature i has a set M_i of admissible values ($i = 1, 2, \ldots, n$). It is usual to consider the features that have the following sets of values:

1°. $M_i^2 = \{0, 1\}$: the feature is or is not defined on the object.

2°. $M_i^k = \{0, 1, 2, \ldots, k-1\}$: the feature has several gradations, ($k > 2$).

3°. $\tilde{M}_i^k = \{a_1, \ldots, a_k\}$: the feature takes the finite number of values. The elements in $\tilde{M}_i^k$ are generally not numbers ($k > 2$).

4°. $M_i = [a, b]$, $(a, b]$, $[a, b)$, (a, b), and a, b: arbitrary numbers or the symbols $-\infty$ and $+\infty$.

5°. M_i: a more complicated subset of the set of real numbers.

6°. M_i^f : the values of the feature i are functions in a class of functions.

7°. M_i^μ: the values of the feature i are distribution functions of a random variable.

Clearly, the sets 1° through 7° listed above do not exhaust the diversity of the features that can be found in recognition problems.

In what follows, we assume that the sets M_i of feature values can be supplemented with the element Δ which means that the value of the ith feature is not known. The set $M_i \cup \Delta$ is denoted by $M_i(\Delta)$.

Let there be given the collections $(b_1, \ldots, b_n) = \tilde{b}$ and $(c_1, \ldots, c_n) = \tilde{c}$ such that $b_i, c_i \in M_i(\Delta)$. The collections $\tilde{b}$ and $\tilde{c}$ are called distinct if there is at least one number i such that $b_i \ne \Delta$, $c_i \ne \Delta$, and $b_i \ne c_i$ ($1 \le i \le n$).

Definition 1. The description $I(S) = (a_1(S), \ldots, a_n(S))$, $a_i(S) \in M_i(\Delta)$, of the admissible object S, $S \in M$, is called the standard description of S. The description $I(S)$ is called complete if $a_i(S) \ne \Delta$, $i = 1, 2, \ldots, n$.

There exist traditionally several specifically selected classes of features and names for them. Thus, the features i with the value set M_i^2 are called binary features; those with (M_i, 4°) are simple numerical features; those with (M_i, 5°) are complex numerical features; those with M_i^f are functional features, and those with M_i^μ are probabilistic features.

Features with special additional constraints imposed on the set M_i are particularly important in recognition problems.

Definition 2. The feature i ($1 \le i \le n$) such that its M_i is a metric space is called a metric feature.

If the metric in M_i is denoted by ρ_i, then i can be denoted subsequently by (M_i, ρ_i). In some cases, the function ρ_i satisfies all axioms of distance except the

Table 1

	1	2	…	i	…	$n-1$	n	
S_1	1	a_{12}	…	a_{1i}	…	a_{1n-1}	a_{1n}	Class K_1
⋮	…	…	…	…	…	…	…	
S_{m_1}	$m_1 1$	$a_{m_1 2}$	…	$a_{m_1 i}$	…	$a_{m_1 n-1}$	$m_1 n$	
.	…	…	…	…	…	…	…	
$S_{m_{l-1}+1}$	$a_{m_{l-1}+1,1}$		…	$a_{m_{l-1}+1,i}$	…	$a_{m_{l-1}+1,n}$		Class K_l
⋮	…	…	…	…	…	…	…	
S_m	a_{m1}	…		a_{mi}	…	a_{mn}		

axiom of a triangle. If so, ρ_i is a semimetric and the feature (M_i, ρ_i) is a semimetric feature.

In actual problems, the standard descriptions of admissible objects usually include different types of features.

2. Information vectors. The recognition (classification) problem consists of evaluating the elementary properties $P_j(S)$: "S belongs to K_j" for the given S and the set of classes $K_1, \ldots, K_l$.

Consider three types of information about $P_j(S)$: it was established that $S \in K_j$; it was established that $S \bar{\in} K_j$; and it is not known whether or not S belongs to K_j. These facts are coded by symbols 1, 0, and Δ, respectively. The information that S belongs to $K_1, \ldots, K_l$ is coded by the vector $(\alpha_1\alpha_2 \ldots \alpha_l)$, $a_i \in \{0, 1, \Delta\}$, $(i = 1, \ldots, l)$. The information can be true or false, complete or incomplete. The situations that may arise here give grounds for introducing the following definitions.

Definition 3. The vector $\tilde{\alpha} = (\alpha_1 \ldots \alpha_l)$ is called the information vector if $\alpha_t \in \{0, 1, \Delta\}$. The information vector $\tilde{\alpha}$ assigned to the admissible object S is denoted by $\tilde{\alpha}(S) = (\alpha_1(S), \ldots, \alpha_l(S))$. The vector $\tilde{\alpha}(S)$ is complete if $\alpha_i(S) \neq \Delta$ $(i = 1, 2, \ldots, l)$. The vector $\tilde{\alpha}(S)$ is called well-defined for S if the condition $\alpha_j(S) \neq \Delta$ implies that $P_j(S) = \alpha_j(S)$, $(j = 1, 2, \ldots, l)$. The complete vector $\tilde{\alpha}(S)$, well-defined for S is called a true vector for S [32].

3. Let there be given the collection $S_1, \ldots, S_m$ of admissible objects and their standard descriptions $I(S_1), \ldots, I(S_m)$. Let also every object in S_t be assigned a well-defined information vector $\tilde{\alpha}(S_t)$ different from $(\Delta\Delta \ldots \Delta\Delta)$.

Definition 4. The standard learning information $I_0(K_1, \ldots, K_l)$ is the collection of sets $\mathfrak{M}_1 = (I(S_1), \ldots, I(S_m))$ and $\mathfrak{M}_2 = (\tilde{\alpha}(S_1), \ldots, \tilde{\alpha}(S_m))$. The standard information $I_0(K_1, \ldots, K_l)$ is written below as

$$I_0(K_1, \ldots, K_l) = \mathfrak{M}_1 \cup \mathfrak{M}_2 \tag{6}$$

or

$$I_0(K_1, \ldots, K_l) \\ = (I(S_1), \tilde{\alpha}(S_1), \ldots, I(S_m), \tilde{\alpha}(S_m)). \tag{7}$$

The standard learning information is called well-defined (true) if the vectors $\tilde{\alpha}(S_i)$ are well-defined (true) for S_i.

We consider, henceforth, true standard informations. As well-defined standard informations are discussed further, it is assumed that the elements of $\mathfrak{M}_2$ do not include the vector $(\Delta\Delta \ldots \Delta\Delta)$.

The standard information that contains the descriptions of the objects $S_1, \ldots, S_m$ in terms of features $1, 2, \ldots, n$ and the information vectors of length l are denoted below by

$$I(S_1, \ldots, S_m, K_1, \ldots, K_l).$$

If it is clear a priori what objects, features, or classes are meant or an exact enumeration of the appropriate components is not needed, we use the notation

$$I_0(l), I_0(m, l), \text{etc.}$$

4. Learning information in a problem with disjoint classes $K_1, \ldots, K_l$ is often given in a special form, i.e., a learning table T^0_{nm}. The rows in the table are standard descriptions of the objects $S_1, \ldots, S_m$ (see Table 1).

In what follows, we assume that the objects $S_1, \ldots, S_{m_1}$ belong to $K_1, \ldots$, the objects $S_{m_{j-1}+1}, \ldots, S_{m_j}$ to $K_j, \ldots$, and the objects $S_{m_{l-1}+1}, \ldots, S_m$ to K_l.

5. Parts of standard learning information. Let us select a subset of objects $S_{i_1}, \ldots, S_{i_k}$ in the set $S_1, \ldots, S_m$ and a subset of features $u_1, \ldots, u_t$ in the system of features $1, 2, \ldots, n$.

We leave only the descriptions $I(S_{i_1}), \ldots, I(S_{i_k})$ in $I_0(K_1, \ldots, K_l)$ and only the coordinates numbered $u_1, \ldots, u_t$ in these descriptions. The information vectors are left unchanged. The information thus obtained is called a part of standard learning information.

Clearly, a part of standard information is standard information.

II. The Structural Learning Information and Structural Descriptions

Given finite alphabets $\mathfrak{A} = \{a_1, \ldots, a_n\}$ and $\Lambda = \{\Lambda_1, \ldots, \Lambda_q\}$, we form the language Σ as follows:

1°. All finite words with characters in $\mathfrak{A}$ belong to Σ;

2°. If $S', S'' \in \Sigma$, then $S'\Lambda_i S'' \in \Sigma$; if S' and S'' are nonempty words.

In what follows, words in Σ are called phrases.

Definition 5. The description $I(S)$ of an admissible object is called structural if it is an element in Σ (a phrase).

Clearly, any standard description is structural. The part of a phrase enclosed between the consecutive occurrences of characters in Λ from the first character inclusive to the first character in Λ exclusive is called the correct part of a phrase or a word from the last occurrence of the character in Λ (exclusive) to the last character of the phrase (inclusive).

The remaining parts of the phrase are called subphrases.

Let for every class K_j there be specified a collection of phrases $B_1^j, \ldots, B_{k(j)}^j$ and operations $O_1^j, \ldots, O_{q(j)}^j$, whose application to the phrases forms a phrase. Let further K_j consist solely of the phrases that are derived from the phrases B_u^j $(u = 1, 2, \ldots, k(j))$ by applying the operations O_t^j $(t = 1, 2, \ldots, q(j))$.

Definition 6. The description $I(K_j) = \{B_1^j, \ldots, B_{k(j)}^j, O_1^j, \ldots, O_{q(j)}^j\}$ of K_j is called structural or algebraic and the collection $B^j = \{B_1^j, \ldots, B_{k(j)}^j\}$ is the generating set (the basis) in $I(K_j)$.

The collection $I(K_1), \ldots, I(K_l)$ is a generalization of the collection of the descriptions $I(S_1), \ldots, I(S_m)$ in standard learning information.

Let there be specified a collection of properties $P_1^j(\Phi), \ldots, P_{n(j)}^j(\Phi)$ defined on the phrases Φ in Σ, which are the descriptions of admissible objects in K_j. We construct a class of formulas $\{\varphi\}_j$ over $P_1^j, \ldots, P_n^j$ using, for example, the logical operations $\wedge$, $\vee$, $-$, and $\longrightarrow$ and the quantifiers $\exists$ and $\forall$. Note that the predicates $P_1^j, \ldots, P_n^j$ are generally multiplace and Φ is solely one of the arguments in P_i^j $(i = 1, 2, \ldots, n)$.

For example, $P(\Phi, \Phi')$ implies "Φ' is a word in Φ." Select a collection of elementary relations that describe the membership of S in the classes K_j. Note particularly among them: $Q_j(S)$ $(Q_j(\Phi))$ implying that an object S belongs to K_j $(j = 1, 2, \ldots, l)$, $Q_{ut}^{p1}(S)$ $(Q_{ut}^{p1}(\Phi))$ implying that "$S \in K_u$ with a greater probability than $S \in K_t$," $Q_{ut}^{p0}(S)$ $(Q_{ut}^{p0}(\Phi))$ implying that the events $S \in K_u$ and $S \in K_t$ are equiprobable, and $Q_j^{p,a,b}(S)$ $(Q_j^{p,a,b}(\Phi))$ implying that the event $S \in K_j$ has probability p enclosed in the interval $[a, b]$, that is $a \le p \le b$.

Similarly, we introduce $Q_{ut}^{f,1}$, $Q_{ut}^{f,0}$, and $Q_j^{f,a,b}$, where, instead of probability p, we consider the membership function f of an object S in K_u, K_t, and K_j.

The collection of elementary relations (predicates) may include other elementary relations as well.

We construct a system of formulas $\{\psi\}_j$ on the elementary relation $Q^1, \ldots, Q^r$ using the logical operations $\wedge$, $\vee$, $-$, and $\longrightarrow$.

Finally, we specify a collection of formulas, $\{\chi\}_j$, where the formulas take the form: (a) $\varphi \longrightarrow \psi$ and (b) $\psi \longrightarrow \varphi$, such that $\varphi \in \{\varphi\}_j$ and $\psi \in \{\psi\}_j$.

The implication symbol in $A \longrightarrow B$ is defined only for the case where $A = 1$ in a standard fashion: if $A = 1$, then $B = 1$.

Definition 7. The collection of formulas $\{\chi\}_j$ is called the information set K_j in the structural learning information.

Definition 8. The set $\tilde{I}_0(K_1, \ldots, K_l) = \{I(K_1), \{\chi(K_1)\}, \ldots, I(K_l), \{\chi(K_l)\}\}$ is called the structural learning information over the classes $K_1, \ldots, K_l$.

Example 1. Let $\Phi(S)$ be the description of an object S and a collection of words $\Phi^1, \ldots, \Phi^{r(j)}$ be specified for the class K_j.

The elementary predicates $P_i(\Phi(S), \Phi^i)$ imply that the word Φ^i is a correct part of the phrase $\Phi(S)$, $i = 1, 2, \ldots, r(j)$.

Apply the operations $\wedge$, $\vee$, and $-$ to $\{P_i\}$ construct $\{\varphi\}_j$. The set $\{\varphi\}_j$ consists of all possible DNFs $\vee\, P_{i_1}^{\sigma_{i_1}} \cdot \ldots \cdot P_{i_k}^{\sigma_{i_k}}$.

The elementary predicates for the system $\{\psi\}_j$: $Q_{ut}^{p,1}(\Phi)$ are $S \in K_u$. They imply that S belongs to K_u with a greater probability than S belonging to K_t $(u = 1, 2, \ldots, l, t = 1, 2, \ldots, u-1, u+1, \ldots, l)$. The predicate $Q_{ut}^{p,1}(\Phi)$ in this example means $K_u(\Phi) > K_t(\Phi)$.

We form the system $\{\psi\}_j$ by applying the operations $-$, $\wedge$, and $\vee$ to the elementary predicates.

The collection of formulas $\{\chi\}_j$ does not contain formulas of the $\psi \longrightarrow \varphi$ type. We assign the only formula in ψ of the type $\prod_t (K_j(\Phi) > K_t(\Phi))$ to each elementary predicate $P_i(\Phi(S), \Phi^i)$, and the product is taken over the indices occurring in the subset $\{1, 2, \dots, l\}\backslash j$. We include the following formula in $\{\chi\}_j$:

$$P_i(\Phi(S), \Phi^i) \longrightarrow \prod_t (K_j(\Phi) > K_t(\Phi)),$$

$$i = 1, 2, \dots, r(j).$$

We assign no formulas from $\{\psi\}_j$ to the formulas in $\{\varphi\}_j$ which contain at least one $\bar{P}_i$. If $A, B \in \{\varphi\}_j$ and $\{\chi\}_j$ includes the formulas

$$A \longrightarrow \prod_u (K_j(\Phi) > K_u(\Phi)),$$

$$B \longrightarrow \prod_v (K_j(\Phi) > K_v(\Phi)),$$

with the indices u and v running through the subsets M_u and M_v, respectively, then $\{\chi\}_j$ includes the formulas

$$A \cdot B \longrightarrow \prod_w (K_j(\Phi) > K_w(\Phi)), \quad w \in M_u \cup M_v,$$

$$A \vee B \longrightarrow \prod_{w \in (M_u \cap M_v)} (K_j(\Phi) > K_w(\Phi)),$$

if $M_u \cap M_v$ is nonempty.

Thus, we built the system of formulas $\{\chi\}_j$ $(j = 1, 2, \dots, l)$.

Clearly, we can likewise consider the concept of a part of the structural learning information:

$$\tilde{I}_a(K_1, \dots, K_l) \text{ is the part } \tilde{I}(K_1, \dots, K_l),$$

if

$$\tilde{I}_a(K_1, \dots, K_l) = \{I^\alpha(K_1), \{\chi_1\}^\alpha, \dots, I^\alpha(K_l), \{\chi_l\}^\alpha\},$$

$$\tilde{I}(K_1, \dots, K_l) = \{I(K_1), \{\chi\}_1, \dots, I(K_l), \{\chi\}_l\}$$

$$\{\chi_j\}^\alpha \subseteq \{\chi\}_j, \quad I^\alpha(K_j) \subseteq I(K_j).$$

Finally, we can combine an arbitrary part of the standard information $I(K_1, \dots, K_l)$ and an arbitrary part of the structural information $\tilde{I}(K_1, \dots, K_l)$.

We do not dwell on other forms of learning information.

§ 3. The Quality Functional $\varphi(A)$ of the Algorithm A

We consider the following types of the functionals $\varphi(A)$. Let $S'_1, \dots, S'_q$ be a collection of admissible objects called further a control set; $\tilde{\alpha}_i = (\alpha_{i1} \dots \alpha_{il})$ be true information vectors of objects S'_i; and $\tilde{\alpha}_i^A = (\alpha_{i1}^A \dots \alpha_{il}^A)$ be information vectors constructed for S'_i by the algorithm A $(i = 1, 2, \dots, q)$. Let $\rho(\tilde{\alpha}, \tilde{\alpha}')$ denote an arbitrary semimetric in the space of information vectors.

Table 2

$P_j(S'_i)$ \ α_{ij}^A	0	1	Δ
0	φ_{ij}^{00}	φ_{ij}^{01}	$\varphi_{ij}^{0\Delta}$
1	φ_{ij}^{10}	φ_{ij}^{11}	$\varphi_{ij}^{1\Delta}$

Consider the sequence of functions $f_1(x), f_2(x_1, x_2), \dots, f_t(x_1, \dots, x_t)$, which satisfy the following conditions:

1°. All f_t are defined for $x_i \geq 0$ $(i = 1, 2, \dots, t)$.

2°. All f_t are not increasing in any variable.

3°. The function f_t reaches the absolute maximum at the point $(0, 0, \dots, 0)$, and this maximum is 1.

The latter condition can always be satisfied through a proper choice of the normalizing factor.

Definition 9. The quality function $\varphi(A)$ over the control set $S'_1, \dots, S'_q$ is the quantity

$$f_q(\rho(\tilde{\alpha}_1, \tilde{\alpha}_1^A), \dots, \rho(\tilde{\alpha}_q, \tilde{\alpha}_q^A)).$$

Subsequently, we sometimes omit the words "over the control set."

Definition 9 is very general. A special concept of quality functional is used when optimum algorithms are formulated constructively.

Let there be a pair (S'_i, K_j). The former is an object from the control sample; the latter is a class from $K_1, \dots, K_l$, and Table 2 is given for each pair.

Here, α_{ij}^A is the value of the predicate $P_j(S'_i)$, "$S'_i \in K_j$" evaluated in the algorithm A, $P_j(S'_i)$ is the true value of this predicate, $\varphi_{ij}^{\alpha, \beta}$ is a numerical estimate (a reward or a penalty) of the event: $P_j(S'_i) = \alpha$, $\alpha_{ij}^A = \beta$, $\alpha \in \{0, 1\}$, and $\beta \in \{0, 1, \Delta\}$. For the control sample $S'_1, \dots, S'_q$, a matrix $\{\gamma_{ij}\}_{q \times l}$, where $\gamma_{ij} = \varphi_{ij}^{\alpha_{ij}, \alpha_{ij}^A}$, is constructed from the set of true information vectors $\tilde{\alpha}_i$ and vectors $\tilde{\alpha}_i^A$.

Definition 10. The functional $\varphi(A) = \dfrac{1}{ql}\sum_{i=1}^{q}\sum_{j=1}^{l}\gamma_{ij}$ [32] is called the linear quality functional.

An important special case of the linear functional is the functional $\varphi^*(A)$ such that Tables 2 are the same for

Table 3

$P_j(S_i')$ \ α_{ij}^A	0	1	Δ
0	1	0	0
1	0	1	0

all pairs (i, j) $(i = 1, 2, \dots, q, j = 1, 2, \dots, l)$ and take the form given in Table 3.

The functional $\varphi^*(A)$ is called the fraction of correct predictions.

CHAPTER II
SOME MODELS IN THE THEORY OF RECOGNITION ALGORITHMS

SECTION I. MODELS OF THE ALGORITHMS OF ESTIMATES CALCULATION

§ 1. General Remarks

Algorithms based on the estimates calculation principle are a formalization of the idea of precedence or partial precedence. Suppose that standard descriptions of objects $\{\tilde{S}\}$, such that $\tilde{S} \in K_j$ and $\{S'\}$, $S' \overline{\in} K_j$ are specified. An object S, $j = 1, 2, \dots, l$ is presented for recognition. Assume that a method is given whereby we can determine the degree of closeness between some parts of the description of S and the corresponding parts of the descriptions $\{I(\tilde{S})\}$ and $\{I(S')\}$. By evaluating the closeness between the parts of $I(S)$ and $I(\tilde{S})$ and, accordingly, between $I(S)$ and $I(S')$, we can deduce a generalized closeness between S and the object sets $\{\tilde{S}\}$ and $\{S'\}$. In the simplest case, the generalized closeness is equal to the sum of partial closenesses.

With the generalized closenesses denoted by Γ_j^+ and Γ_j^-, we construct a characteristic of the type $\Gamma_j(S) = \Gamma_j^+ - \Gamma_j^-$, which can naturally be taken as the value of the membership function of the object S in the class K_j. If $S = S_i'$, then $\Gamma_j(S_i')$ is later denoted by Γ_{ij}. The quantity Γ_{ij} is called the estimate of S_i' in the class K_j. Clearly, the estimates calculation algorithm transforms the descriptions of objects S_i' presented for recognition into a numerical matrix $\{\Gamma_{ij}\}_{q\times l}$ further called an estimation matrix. We apply a decision rule to this matrix to construct the matrix $\{\alpha_{ij}\}_{q\times l}$ of information vectors for the objects $S_1', \dots, S_q'$.

§ 2. Some Algorithms of the Estimates Calculation Type

The first estimation algorithms appeared in the 1960s [18]. They proved their worth in the solution of practical problems such as geological forecasting, medical diagnosis, etc. Later [28], they provided the basis for models of estimates calculation algorithms. These models were analyzed in detail in a large number of studies and were also used in various applications.

It was the test algorithm that played a special role in the evolution of the estimates calculation model. This algorithm draws upon the concept of test introduced in 1956 by S.V. Yablonskii. A great number of modifications of the test algorithm, their machine implementations and applications were reported [19, 22, 45–50, 66, 75].

This paper describes a modification of the test algorithm, which is convenient to illustrate the evolution of estimates calculation models. The test algorithm is described for problems with disjoint classes $K_1, \dots, K_l$. The initial information I_0 is presented as the learning table T_{nm}^0 (see Table 1, § 2, Chapter I), and the descriptions of the objects $S_1, \dots, S_m$ are binary vectors.

Definition 11. The test of T_{nm}^0 is the collection of columns, $i_1, \dots, i_k$, such that deletion of all columns, except those numbered $i_1, \dots, i_k$, from T_{nm}^0 produces a new table, $T_{n-k,m}$, where all pairs of rows belonging to different classes are distinct. The test $\{i_1 \dots i_k\}$ is called the irreducible test if none of its true parts is a test [71].

Consider the set $\{T\}$ of irreducible tests in the learning table T_{nm}^0. Suppose $T = \{i, \dots, i_k\} \in \{T\}$. We select in the description $I(S_i')$ of the object S_i' being recognized a subdescription $(a_{i_1}, \dots, a_{i_k})$ in terms of features $i_1, \dots, i_k$. Compare $(a_{i_1}, \dots, a_{i_k})$ with all the subdescriptions $(a_{ui_1}, \dots, a_{ui_k})$ of the objects $S_1, \dots, S_m$ in the learning table T_{nm}^0.

Let $\Gamma_{ij}(T)$ denote the number of times that $(a_{i_1}, \dots, a_{i_k})$ coincides with $(a_{ui_1}, \dots, a_{ui_k})$, $u = m_{j-1} + 1, \dots, m_j$; $j = 1, 2, \dots, l$; $m_0 = 0$; $m_l = m$ (according to the descriptions of the jth class). The quantity

$$\Gamma_{ij} = \Gamma_j(S_i') = \frac{1}{m_j - m_{j-1}} \sum_{T \in \{T\}} \Gamma_{ij}(T)$$

is called the estimate of S_i' in the class K_j.

Once the estimates $\Gamma_1(S_i'), \dots, \Gamma_l(S_i')$ are available, it is easy to classify S_i' using simple decision rules. For example, if the set of estimates has at least two maxima, the object is not classifiable. With one maximum, the object is assigned to the class where it has the highest estimate.

Somewhat later algorithms were used, where various nonempty subsets of the set $\{1, 2, \dots, n)$ or all possible subsets of the same cardinality k $(1 \le k \le n)$ were taken as the set $\{(i_1 i_2 \dots i_k)\}$.

Geological forecasting problems were solved with algorithms of the KORA type [11]. These used a subclass of subsets of length 3 specifically selected from T^0_{nm} as the subsets $(i_1 \dots i_k)$.

Yablonskii and colleagues [75] came up with the idea that not all the templates in T^0_{nm} should be regarded as being equal. They should be assigned numerical weights, and these should be considered in evaluating $\Gamma_{ij}(T)$ and $\Gamma_j(S'_i)$. Earlier, the weights $p(i)$ of features i $(i = 1, 2, \dots, n)$ were introduced on the basis of the analysis numerous irreducible tests [18], and these were likewise taken into account in evaluating $\Gamma_{ij}(T)$ and $\Gamma_j(S'_i)$.

In a test algorithm, $p(i)$ is evaluated as follows. Let $\tau(n, m)$ be the number of irreducible tests in T^0_{nm}, and $\tau_i(n, m)$ be the number of such tests containing the column i. Then

$$p(i) = \frac{\tau_i(n, m)}{\tau(n, m)}.$$

The greater is the weight $p(i)$, the greater is the importance of i in describing admissible objects in M. The latter assertion is justified by the following plausible arguments. The table T^0_{nm} is a description of the set M. It is interesting in that it delivers certain information about the partition of M into classes $K_1, \dots, K_l$.

The irreducible test description cannot be compressed any further but still contains all information about the division of M into classes. The greater is the number of these irredundant descriptions that contain i, the greater is the importance of that feature.

If the weights $p(1), \dots, p(n)$ of features and the weights $\gamma(S_1), \dots, \gamma(S_m)$ of the objects whose descriptions make up the learning table are determined, should $(a_{i_1}, \dots, a_{i_k})$ in S' being recognized coincide with the subrow $(a_{ui_1}, \dots, a_{ui_k})$ of the object S_u, $S_u \in K_j$, this coincidence is rewarded with $\gamma(S_u)(p(i_1) + \dots + p(i_k)) = \Gamma_T(S_u, S')$.

Then,

$$\Gamma_j(S') = \frac{1}{m_j - m_{j-1}} \sum_{T \in \{T\}} \sum_{u = m_{j-1}+1}^{m_j} \Gamma_T(S_u, S'). \quad (8)$$

The properties of irreducible tests for binary learning tables were the subject of many publications [58, 63–65].

§ 3. Models of Estimates Calculation Algorithms for Standard Learning Information

Here, we review only some estimation models. In what follows, we assume that the features $1, 2, \dots, n$ are metric features (Definition 2), i.e., the sets M_i are metric spaces with metrics ρ_i $(i = 1, 2, \dots, n)$.

I. The first step in developing the algorithm A in a model is to select the system Ω_A of subsets in the set $\{1, 2, \dots, n\}$. The elements of Ω_A are called the *reference sets* of the algorithm and the system Ω_A is the system of reference subsets of A. Examples of Ω_A are sets of irreducible tests (in the problem with disjoint classes and the binary learning table T^0_{nm}), the system of all subsets of cardinality k (in this case Ω_A is defined by the parameter k, $1 \le k \le n - 1$), and the system of all nonempty subsets of the set $\{1, 2, \dots, n\}$.

Clearly, each subset $\Omega = \{i_1, i_2, \dots, i_k\}$ may be placed in one-to-one correspondence with a *characteristic Boolean vector* $\tilde{\omega} = (\alpha_1 \dots \alpha_n)$, where $\alpha_{i_1} = \dots = \alpha_{i_k} = 1$ and the remaining coordinates are 0.

Definition 12. The characteristic function $f^A_\Omega(x_1, \dots, x_n)$ of the system Ω_A (algorithm A) is a Boolean function defined by the relation

$$f^A_\Omega(\tilde{\omega}) = \begin{cases} 1, & \text{if } \tilde{\omega} \text{ is the characteristic vector} \\ & \text{of an element in } \Omega_A \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

In many situations, it is convenient to specify Ω_A in terms of $f^A_\Omega(x_1, \dots, x_n)$.

Example 2. If Ω_A is the system of all nonempty subsets of the set $\{1, 2, \dots, n\}$, then $f^A_\Omega = 0$ solely on the collection (00 ... 00). Then $f^A_\Omega = x_1 \vee \dots \vee x_n$. If Ω_A is the system of all subsets of cardinality k, then f^A_Ω is a symmetric Boolean function equal to 1 solely on the kth layer of vertices in an n-dimensional unit cube.

II. The second step in specifying the algorithm A is to define the proximity function $B_{\tilde{\omega}}(S_u, S_t)$, where $\tilde{\omega}$ corresponds to $\Omega \in \Omega_A$ and S_u, S_t are arbitrary admissible objects.

We introduce the concept of the $\tilde{\omega}$ part of the description $I(S)$ (of an object S). Let in $\tilde{\omega}$ the coordinates numbered $i_1, \dots, i_k$ in $\tilde{\omega}$ be 1 and the remaining coordinates be 0.

Definition 13. The description $(a_{i_1}, \dots, a_{i_k})$ is called the $\tilde{\omega}$ part of $I(S)$, which is denoted by $\tilde{\omega} I(S)$. The object S is identified sometimes with its description for brevity. We also use the symbol $\tilde{\omega} S$ where this does not cause confusion.

Accordingly, the $\tilde{\omega}$ part of the set of descriptions $I(S_1), \dots, I(S_m)$ (of the matrix T_{nm}) is the set $\{\tilde{\omega} I(S_1), \dots, \tilde{\omega} I(S_m)\}$ (the matrix $T_{km} = \tilde{\omega} T_{nm}$ from which all columns were deleted except the columns numbered $i_1, \dots, i_k$).

In what follows, the proximity function $B_{\tilde\omega}(S_u, S_t)$ is also denoted by $B(\tilde\omega S_u, \tilde\omega S_t)$.

The following proximity functions were considered in various models [7–9, 12, 13, 24, 25, 29, 31–37, 44, 53, 67–69, 72, 73, 80].

1. $B_\varepsilon^{\vec\varepsilon}(\tilde\omega S_u, \tilde\omega S_t)$. Let $\varepsilon_i \ge 0$ $(i = 1, 2, \ldots, n)$, $\vec\varepsilon = (\varepsilon_1 \ldots \varepsilon_n)$, where ε is an integer, $\varepsilon \ge 0$, $\tilde\omega S_u = (a_{u_1}, \ldots, a_{u_k})$ and $\tilde\omega S_t = (a_{t_1}, \ldots, a_{t_k})$. Write the system of inequalities

$$\rho_i(a_{u_i}, b_{t_i}) \le \varepsilon_i, \quad i = 1, 2, \ldots, k. \tag{10}$$

Here, ρ_i is the metric in the set M_i of the values of the ith feature, $i = 1, 2, \ldots, n$. Then

$$B_\varepsilon^{\vec\varepsilon}(\tilde\omega S_u, \tilde\omega S_t) = \begin{cases} 1, & \text{if the number of unsatisfied inequalities in (10) is not greater than } \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

Important special forms of the function $B_\varepsilon^{\vec\varepsilon}$ are $B_\varepsilon^{(0\ldots0)}$. The latter is equal to 1 if and only if not more than ε coordinates are not equal in $\tilde\omega S_u$ and $\tilde\omega S_t$; $B_0^{\vec\varepsilon} = 1$ if all inequalities in (10) are satisfied, and $B_0^{(0\ldots0)} = 1$ if $\tilde\omega S_u = \tilde\omega S_t$.

$$2.\ \ B_\varepsilon(\tilde\omega S_u, \tilde\omega S_t) = \begin{cases} 1, & \text{if } \sum\limits_{j=1}^{k} \rho_j(a_{u_j}, b_{t_j}) \le \varepsilon \\ 0 & \text{otherwise.} \end{cases}$$

For the case of numerical features such that $\rho_i(x, y) = |x - y|$, the function $B_\varepsilon(\tilde\omega S_u, \tilde\omega S_t)$ was analyzed in detail in [57].

In what follows, we mainly review models with the function $B_\varepsilon^{\vec\varepsilon}$ and with functions B that are dependent significantly on the inequalities in (10) but of a more general form than $B_\varepsilon^{\vec\varepsilon}$.

III. Estimates $\Gamma_\omega(I(S), I(S_i))$, and $\Gamma_\omega(S, S_i)$. Let S_i be part of the standard learning information $I_0(K_1, \ldots, K_l)$, S be an admissible object, and $\tilde\omega$ be the characteristic vector of Ω, $\Omega \in \Omega_A$.

Let also the object S_i be assigned numerical parameters $(\gamma_1(S_i), \ldots, \gamma_q(S_i)) = \tilde\gamma(S_i)$ and the set Ω (the vector $\tilde\omega$) be assigned numerical parameters $(p_1(\tilde\omega), \ldots, p_r(\tilde\omega)) = \tilde p(\tilde\omega)$. Then

$$\Gamma_{\tilde\omega}(S, S_i) = f(B_\omega(S, S_i), \tilde\gamma(S_i), \tilde p(\tilde\omega)) \tag{11}$$

is called the estimate of S with respect to the object S_i and the set Ω.

The following form of the function $\Gamma_{\tilde\omega}$ was considered in applications:

$$\Gamma_{\tilde\omega}(S, S_i) = (\tilde\gamma(S_i)\tilde p(\tilde\omega))B_{\tilde\omega}(S, S_i). \tag{12}$$

In many cases, if the set Ω associated with the vector $\tilde\omega$ consists of elements $i_1, \ldots, i_k$, then $p(\tilde\omega) = p_{i_1} + \ldots + p_{i_k}$, $q = 1$, and then

$$\Gamma_{\tilde\omega}(S, S_i) = \gamma_1(S_i)(p_{i_1} + \ldots + p_{i_k})B_{\tilde\omega}(S, S_i). \tag{13}$$

In the general case, if the function $B_{\tilde\omega}(S, S_i)$ is taken to be the function $B_\varepsilon^{\vec\varepsilon}(\tilde\omega S_i, \tilde\omega S)$, then the quantities $\Gamma_{\tilde\omega}$ in (11) are specified by collections of values of the parameters $\varepsilon_1, \ldots, \varepsilon_n$, $\gamma_1(S_1) = \gamma_1, \ldots, \gamma_1(S_m) = \gamma_m$, and $p_1, \ldots, p_n$. If the reference subsets are collections of all the subsets of the set $\{1, 2, \ldots, n)$ of k elements, then the steps in (I) through (III) are defined by the collection of values of the parameters $(k, \varepsilon, \vec\varepsilon, \vec p, \vec\gamma)$; that is, by the collection of values of $2n + m + 2$ parameters. The parameters ε_i are usually called accuracies of feature measurements, the parameters p_i are called weights of features, and the parameters $\gamma(S_u) = \gamma_u$ are called weights of objects. Thus, they are given a simple physical interpretation. The set of parameters $(k, \varepsilon, \vec\varepsilon, \vec p, \vec\gamma)$ is crucial in the definition of the most commonly used model containing the subclass of estimation algorithms.

IV. The estimate $\Gamma^j_{\tilde\omega}(S)$ of an object S by its reference set Ω (or its characteristic vector $\tilde\omega$) in the class K_j:

$$\Gamma^j_{\tilde\omega}(S) = \varphi(\Gamma_{\tilde\omega}(S, S^j_{u_1}), \ldots, \Gamma_{\tilde\omega}(S, S^j_{u_t})), \tag{14}$$

where $\{S^j_{u_i}\}$ is a subset of objects from $S_1, \ldots, S_m$ that belong to K_j $(i = 1, 2, \ldots, t)$. The function $\Gamma^j_{\tilde\omega}(S)$ in (14) is defined most often as

$$\Gamma^j_{\tilde\omega}(S) = \frac{1}{\mu(W_j^1)} \sum_{S^j_{u_i} \in W_j^1} \Gamma_{\tilde\omega}(S, S^j_{u_i}),\ \mu(W_j^1), \tag{15}$$

where $\mu(W_j^1)$ is the number of elements in W_j^1 and $W_j^1 = K_j \cap \{S_1, \ldots, S_m\}$.

Then in the model (*k*, ε, , ,):

$$\Gamma^j_{\tilde{\omega}}(S) = \frac{1}{\mu(W^1_j)} \times \sum_{S^j_{u_t} \in W^1_j} \gamma(S^j_{u_t})(p_{i_1} + \ldots + p_{i_{k(\tilde{\omega})}}) B_{\tilde{\omega}}(S, S^j_{u_t}). \tag{16}$$

In problems with disjoint classes

$$\Gamma^j_{\tilde{\omega}}(S) = \frac{1}{m_j - m_{j-1}} \times \sum_{i = m_{j-1}+1}^{m_j} \gamma(S_i)(p_{i_1} + \ldots + p_{i_{k(\tilde{\omega})}}) B^{\vec{\varepsilon}}_{\varepsilon}(S, S_i). \tag{17}$$

V. The estimate $\Gamma_j(S)$ with respect to the class K_j. It is defined as a function of the estimates $\Gamma^j_{\tilde{\omega}}(S)$ over various subsets Ω, $\tilde{\omega} \longleftrightarrow \Omega$. Ordinarily,

$$\Gamma_j(S) = \sum_{\tilde{\omega} \leftrightarrow \Omega \in \Omega_A} \Gamma^j_{\tilde{\omega}}(S)$$

or (18)

$$\Gamma_j(S) = \frac{1}{N} \sum_{\tilde{\omega} \leftrightarrow \Omega \in \Omega_A} \Gamma^j_{\tilde{\omega}}(S),$$

where *N* is a normalizing factor. In view of (15),

$$\Gamma_j(S) = \frac{1}{N} \sum_{\tilde{\omega} \leftrightarrow \Omega \in \Omega_A} \sum_{S^j_{u_t} \in W^1_j} \Gamma_{\tilde{\omega}}(S, S^j_{u_t}). \tag{19}$$

In regards to (16) and (19), in the model $(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma})$, we generally have

$$\Gamma_j(S) = \frac{1}{N} \cdot \frac{1}{\mu(W^1_j)} \times \sum_{\{\tilde{\omega}^k\}} \sum_{S^j_{u_t}} (p_{i_1} + \ldots + p_{i_k}) B^{\vec{\varepsilon}}_{\varepsilon}(\tilde{\omega}S, \tilde{\omega}S^j_{u_t}). \tag{20}$$

Here, the first summation is taken over the vectors $\tilde{\omega}^k$ that have exactly *k* ones among their coordinates and the second summation over the elements $S^j_{u_t}$ that are members of $W^1_j = K_j \cap \{S_1, \ldots, S_m\}$.

In the problem with disjoint classes,

$$\Gamma_j(S) = \frac{1}{N} \cdot \frac{1}{m_j - m_{j-1}} \times \sum_{\{\tilde{\omega}^k\}} \sum_{i = m_{j-1}+1}^{m_j} \gamma(S_i)(p_{i_1} + \ldots + p_{i_k}) B^{\vec{\varepsilon}}_{\varepsilon}(\tilde{\omega}S, \tilde{\omega}S_i). \tag{21}$$

If we consider in succession objects of the control sample $S'_1, \ldots, S'_q$ as *S*, then the initial information

$$(I(S_1), \tilde{\alpha}(S_1), \ldots, I(S_m), \tilde{\alpha}(S_m), I(S'_1), \ldots, I(S'_q)) = (I_0(l), I_S(q))$$

is transformed by computations set out in I through V to a matrix $\{\Gamma_{ij}\}_{q \times l}$. Here, $\Gamma_{ij} = \Gamma_j(S'_i)$ $(i = 1, 2, \ldots, q;\ j = 1, 2, \ldots, l)$. The matrix $\{\Gamma_{ij}\}$ is called an estimation matrix. The operator R_A such that

$$R_A(I_0(l), I_S(q)) = \{\Gamma_{ij}\}_{q \times l} \tag{22}$$

is called an estimation operator. In the cases where Γ_{ij} are determined by the choice of values for the parameters $\pi_1, \ldots, \pi_t$, the operator R_A is sometimes denoted by

$$R_A\begin{pmatrix} I_0(l), I_S(q) \\ \pi_1 \ldots \pi_t \end{pmatrix} = \{\Gamma_{ij}\}_{q \times l} \tag{23}$$

or

$$R_A(\pi_1, \ldots, \pi_t).$$

Thus, for the model with the parameters $(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma})$

$$R_A = R_A(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma}).$$

VI. Decision rules in estimates calculations algorithms. This rule *r* for the row $(\Gamma_{i1}, \ldots, \Gamma_{il})$ in the estimation matrix for each class K_j $(j = 1, 2, \ldots, l)$ evaluates the predicate $P_j(S'_i) =$ "$S'_i \in K_j$" or refuses to do so. It is written in the latter case $P_j(S'_i) = \Delta$.

Thus, $r(\Gamma_{i1}, \ldots, \Gamma_{il}) = (\alpha^A_{i1} \ldots \alpha^A_{il}) = \tilde{\alpha}_A(S'_i)$, where $\tilde{\alpha}_A(S'_i)$ is the information vector of S'_i in the algorithm *A* defined by the operator R_A and by the decision rule *r*. Clearly,

$$r(\{\Gamma_{ij}\}_{q \times l}) = \{\alpha^A_{ij}\}_{q \times l}.$$

Here, $\{\alpha^A_{ij}\}_{q \times l}$ is the matrix of the information vectors of the elements $S'_1, \ldots, S'_q$ constructed by the algorithm *A*.

Different estimation models use different decision rules. In problems with disjoint classes, for example, the following decision rule was used:

The elements of the row $(\Gamma_{i1}, \ldots, \Gamma_{il})$ include the maximum element Γ_{ij}. If $\Gamma_{ij} - \Gamma_{iu} > \delta_1$ $(u \neq j)$ and $\frac{\Gamma_{ij}}{\sum^{l}_{u \neq j} \Gamma_{iu}} > \delta_2$, then $S'_i \in K_j$ and, consequently, $S'_i \overline{\in} K_u$, $u \neq j$. If at least one of the conditions listed above is not satisfied, then all the elements in the row $(\alpha^A_{i1}, \ldots, \alpha^A_{il})$ of the matrix $\{\Gamma_{ij}\}$ are equal to Δ. The model defined by the operator $R(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma})$ and by this decision rule with the parameters δ_1 and δ_2 is denoted by

$$\mathfrak{M}(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma}, \delta_1, \delta_2). \tag{24}$$

This model and its submodels were used most often in solving practical problems by estimation algorithms [7, 13, 34, 36, 72].

More general forms of decision rules are as follows:

1°. For each class K_j, we specify the linear form

$$b_1^j x_1 + \ldots + b_l^j x_l + b_{l+1}^j = L_j(\tilde{b}^J, \tilde{x})$$

and the threshold constants c_{1j}, c_{2j} ($j = 1, 2, \ldots, l$). Let $\tilde{\Gamma}_\iota = (\Gamma_{\iota 1}, \ldots, \Gamma_{\iota l})$.

The linear decision rule:

$$\text{if } L_j(\tilde{b}^J, \tilde{\Gamma}_\iota) > c_{2j}, \text{ then } S_\iota' \in K_j \text{ or } \alpha_{\iota j}^A = 1,$$

$$\text{if } L_j(\tilde{b}^J, \tilde{\Gamma}_\iota) < c_{1j}, \text{ then } S_\iota' \,\bar{\in}\, K_j \text{ or } \alpha_{\iota j}^A = 0,$$

for $c_{2j} \geq L_j(\tilde{b}^J, \tilde{\Gamma}_\iota) \geq c_{1j}$, the algorithm A does not establish the membership of S_ι' in K_j and $\alpha_{\iota j}^A = \Delta$.

2°. Similarly to 1° above, it is possible to define a nonlinear decision rule. In this case, instead of the linear forms $L_j(\tilde{b}^J, \tilde{x})$, we specify a system of continuous functions $f_j(x_1, \ldots, x_l)$ ($j = 1, 2, \ldots, l$).

Important special cases of the rules defined in 1° and 2° above are

$$L_j(\tilde{b}^J, \tilde{x}) \equiv L(\tilde{b}, \tilde{x}), \quad j = 1, 2, \ldots, l, \tag{25}$$

$$f_j(x_1, \ldots, x_l) \equiv f(x_1, \ldots, x_l), \quad j = 1, 2, \ldots, l. \tag{26}$$

By varying rules I through VI for the determination of characteristics in estimation models, we can obtain various models for recognition algorithms of the estimation type.

§ 4. Formulas for $\Gamma_j(S)$

The elements of a matrix of estimates are usually impossible to evaluate directly from their definitions in § 3.

Indeed, the number of reference subsets in the model $\mathfrak{M}(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma})$ (24) is C_n^k and the number of terms in (19) for $\Gamma_j(S)$ is $\mu(W_j^1)C_n^k$, where n is the number of features in the description of objects and $\mu(W_j^1)$ is the number of descriptions of objects in K_j in the learning information.

In many cases, however, it is possible to develop practically efficient formulas for $\Gamma_j(S)$.

It is not the aim of this paper to enumerate all such cases. We only describe two fairly general methods whereby simple formulas can be developed for $\Gamma_j(S)$ and demonstrate several of their applications.

I. The first method draws upon the special property of estimates obtained by (19) when

$$\Gamma_{\tilde{\omega}}^J(S) = \frac{1}{\mu(W_j^1)} \times \sum_{S_u^\iota \in W_j^1} \gamma(S_u^\iota)(p_{\iota_1} + \ldots + p_{\iota_{k(\omega)}}) B_{\tilde{\omega}}(S, S_\iota), \tag{27}$$

where $i_1, \ldots, i_{k(\tilde{\omega})}$ is the collection of all coordinates of value 1 of the vector $\tilde{\omega}$, the proximity function $B_{\tilde{\omega}}$ takes only 0 or 1.

The quantity $\Gamma_j(S)$ is produced from $\Gamma_{\tilde{\omega}}^J(S)$, $\tilde{\omega} \longleftrightarrow \Omega \in \Omega_A$ by (18). Fix the feature t ($1 \leq t \leq n$). Denote by $V_t(S, S_\iota)$ the number of subsets Ω in Ω_A (the number of corresponding characteristic vectors $\tilde{\omega}$) containing the feature t (the coordinate numbered t has value 1) and such that

$$B(\tilde{\omega}S, \tilde{\omega}S_\iota) = 1. \tag{28}$$

The set $K_j \cap \{S_1, \ldots, S_m\}$ is denoted by W_j^1.

Theorem 1.

$$\Gamma_j(S) = \frac{1}{\mu(W_j^1)} \frac{1}{N} \sum_{S_\iota \in W_j^1} \gamma(S_\iota) \sum_{t=1}^{n} p_t V_t(S, S_\iota).$$

Proof. Indeed,

$$\sum_{\tilde{\omega} \leftrightarrow \Omega \in \Omega_A} \Gamma_{\tilde{\omega}}^J(S, S_\iota) = \gamma(S_\iota) \sum_{\Omega \in \Omega_A} B_{\tilde{\omega}}(S, S_\iota) \sum_{t \in \Omega} p_t$$

$$= \gamma(S_\iota) \sum_{\Omega \;\; B_\omega(S, S_\iota) = 1} \sum_{t \in \Omega} p_t = \gamma(S_\iota) \sum_{t=1}^{n} p_t V_t(S, S_\iota).$$

Substituting the above expression in (18) proves the theorem.

Corollary. Let $\Gamma(S, S_\iota)$ denote $\sum_{\Omega \in \Omega_A} \Gamma(\tilde{\omega}S, \tilde{\omega}S_\iota)$

Then

$$\Gamma(S, S_\iota) = \gamma(S_\iota) \sum_{t=1}^{n} p_t V_t(S, S_\iota) \tag{29}$$

and

$$\Gamma_j(S) = \frac{1}{N} \cdot \frac{1}{\mu(W_j^1)} \sum_{S_\iota \in W_j^1} \Gamma(S, S_\iota). \tag{30}$$

Note that, if the number of distinct values of $V_t(S, S_\iota)$ is small in (29), then the formula for $\Gamma(S, S_\iota)$ is simplified and so is the formula for $\Gamma_j(S)$. In this case, the summation over the system of reference sets almost disappears for $\Gamma_j(S)$ in (19), and only the summation over the elements S_t from W_j^1 remains.

Let us describe two forms of the proximity function for which the number of distinct values of $V_t(S, S_t)$ is small for all S_i in W_j^1.

Consider the model $\mathfrak{M}$ where the proximity function for $(\tilde{\omega} I(S_i), \tilde{\omega} I(S))$, $I(S) = (a_1, \dots, a_n)$, $I(S_i) = (b_1 \dots b_n)$ is defined by the system of inequalities $\rho_1(a_1, b_1) \le \varepsilon_1, \dots, \rho_n(a_n, b_n) \le \varepsilon_n$ as follows. If the identically numbered inequalities in the pairs $(\tilde{\omega} S_i, \tilde{\omega} S)$ and $(\tilde{\omega} S_u, \tilde{\omega} S')$ are either both satisfied or both not satisfied, then

$$B(\tilde{\omega} I(S_i), \tilde{\omega} I(S)) = B(\tilde{\omega} I(S_u), \tilde{\omega} I(S')). \tag{31}$$

Definition 14. The functions B that satisfy (31) and assume 0 or 1 are called threshold functions.

The set $\{B\}$ of threshold functions is denoted below by $\{B^{\vec{\varepsilon}}\}$.

Let the pair $(I(S) = (a_1, \dots, a_n)$ and $I(S') = (b_1, \dots, b_n))$ of admissible objects be assigned the characteristic vector $\tilde{\delta} = (\delta_1, \dots, \delta_n) = \tilde{\delta}(S, S')$:

$$\text{if } \rho_i(a_i, b_i) \le \varepsilon_i, \text{ then } \delta_i = 1,$$
$$\text{if } \rho(a_i, b_i) > \varepsilon_i, \text{ then } \delta_i = 0, i = 1, \dots, n.$$

Clearly, if $\tilde{\delta}(S_i, S) = \tilde{\delta}(S_u, S')$, then for any $\tilde{\omega}$ the threshold function $B^{\vec{\varepsilon}}$ satisfies the equality

$$B(\tilde{\omega} S_i, \tilde{\omega} S) = B(\tilde{\omega} S_u, \tilde{\omega} S'). \tag{32}$$

Let there be given vectors $\tilde{\delta}(S, S_i)$ and the characteristic vectors $\tilde{\omega}^1$ and $\tilde{\omega}^2$ of the reference subsets Ω_1 and Ω_2.

Definition 15. The threshold function $B^{\vec{\varepsilon}}$ is called symmetric if the condition that the vectors $\tilde{\delta}(S, S_i)\tilde{\omega}^1$, $\tilde{\delta}(S, S_i)\tilde{\omega}^2$ (here, the multiplication of vectors is taken to mean their coordinate-wise multiplication) have the same number of coordinates of value 1 and the same number of coordinates of value 0 implies that

$$B^{\vec{\varepsilon}}(\tilde{\omega}^1 I(S), \tilde{\omega}^1(I(S_t))) = B^{\vec{\varepsilon}}(\tilde{\omega}^2 I(S), \tilde{\omega}^2 I(S_i)).$$

If in $\tilde{\omega}$ the vectors of value 1 are $i_1, \dots, i_k$, then any threshold function $B^{\vec{\varepsilon}}$ is symmetric if its value on $(\tilde{\omega} I(S), \tilde{\omega} I(S_i))$ is determined only by the number of inequalities satisfied or not satisfied in (10).

Definition 17. The system of reference subsets, Ω_A, is called correct if the condition that $\Omega \in \Omega_A$ and Ω has cardinality k, $1 \le k \le n - 1$, implies that all subsets of cardinality k belong to Ω_A.

Theorem 2. *If the proximity function $B_{\tilde{\omega}}^{\vec{\varepsilon}}$ is symmetric and the system of reference subsets is correct, then $V_t(S, S_i)$ $(t = 1, 2, \dots, n)$ take at most two distinct values for each pair (S, S_i).*

Proof. Combine in one group the features $j_1, \dots, j_q$ such that the corresponding coordinates of the vector $\tilde{\delta}(S, S_i)$ have value 1. Take an arbitrary pair of features, j_u and j_v, from this group. Consider an arbitrary $\tilde{\omega}$ such that

$$j_u \in \Omega(\tilde{\omega} \longleftrightarrow \Omega), \quad j_v \mathrel{\overline{\in}} \Omega \text{ and } B_{\tilde{\omega}}^{\vec{\varepsilon}}(S, S_i) = 1.$$

Then, due to the symmetry of the proximity function, for $\tilde{\omega}^1$ corresponding to the set $(\Omega \cup j_v)\backslash j_u$ we have

$$B^{\vec{\varepsilon}}(\tilde{\omega}^1 I(S), \tilde{\omega}^1 I(S_i)) = 1. \tag{33}$$

Due to (33) and that $\Omega \longleftrightarrow (\Omega \cup j_v)\backslash j_u$ is one-to-one correspondence, we have for the collections M_{10} and M_{01} whose elements are reference sets containing j_u and not containing j_v (in the case of M_{10}) or not containing j_u but containing j_v (in the case of M_{01}): $V_{j_u}(S, S_i) = V_{j_v}(S, S_i)$. We also made use of the fact that the system of reference subsets is correct.

The same equality is proved in a similar way for features associated with 0 coordinates in the vector $\tilde{\delta}(S, S_i)$.

The theorem is proved.

We proved that all features which correspond in $\tilde{\delta}(S, S_i)$ coordinates of value 0 or 1 occur in the same number V^0 or V^1 of reference Ω with respect to which the proximity function between $I(S)$ and $I(S_i)$ is 1.

The theorem can be formulated as follows.

If $(\tilde{v} \cdot \tilde{w})$ is the scalar product of the vectors $\tilde{v}$ and $\tilde{w}$, then

$$\Gamma(S, S_t) = (\tilde{\delta}(S, S_i) \cdot \vec{p}) \cdot V^1 + (\overset{=}{\tilde{\delta}}(S, S_t) \cdot \vec{p}) \cdot V^0, \tag{34}$$

where $\vec{p} = (p_1 \dots p_n)$.

Here are several examples of applications for the theorem. Suppose $B^{\vec{\varepsilon}}(\tilde{\omega} S, \tilde{\omega} S_t) = B_{\varepsilon}^{\vec{\varepsilon}}$ (see 1°, § 3, Chapter II),

$$\Gamma(\tilde{\omega} S, \tilde{\omega} S_i) = \gamma(S_i) B_{\varepsilon}^{\vec{\varepsilon}}(\tilde{\omega} S, \tilde{\omega} S_i)(p_{i_1} + \dots + p_{i_k}).$$

Then (Theorem 1)

$$\Gamma_j(S) = \frac{1}{N\mu(W_j^1)} \sum_{S_t \in W_j^1} \Gamma(S, S_i)$$

and

$$\Gamma(S, S_i) = V_1 \sum_u p_u + V_0 \sum_v p_v,$$

with the summations taken over the coordinates of value 1 (u) and of value 0 (v) in the vector $\tilde{\delta}(S, S_i)$. Consider the system Ω_A of reference sets consisting of all subsets of cardinality k in the set $\{1, 2, \ldots, n\}$.

Let t be the feature which is assigned a coordinate of value 1 in the vector $\tilde{\delta}(S, S_i)$. Then $V_t = V^1$. The quantity V^1 is equal to the number of subsets of cardinality k containing t and not more than ε features which correspond to zero coordinates in $\tilde{\delta}(S, S_i)$. Let $q(S, S_i)$ be the number of one coordinates in $\tilde{\delta}(S, S_i)$.

It is easy to calculate that

$$V^1(S, S_i) = \sum_{u=0}^{\varepsilon} C^u_{n-q(S,S_i)} C^{k-u-1}_{q(S,S_i)-1}.$$

Similarly,

$$V^0(S, S_i) = \sum_{u=1}^{\varepsilon} C^{u-1}_{n-q(S,S_i)-1} C^{k-u}_{q(S,S_i)}.$$

Substitution in (34) yields

$$\Gamma_j(S) = \frac{1}{N\mu(W_j^1)} \sum_{S_i \in W_j^1} \gamma(S_i)$$

$$\times \left\{ (\tilde{\delta}(S, S_i) \cdot \vec{p}) \sum_{u=0}^{\varepsilon} C^u_{n-q} C_q^{k-u-1} \right. \qquad (35)$$

$$\left. + (\bar{\tilde{\delta}}(S, S_i) \cdot \vec{p}) \sum_{u=1}^{\varepsilon} C^{u-1}_{n-q} C_q^{k-u} \right\}.$$

Let now the collection of all subsets in the set $\{1, 2, \ldots, n)$ be chosen as the system Ω_A. The function $B(\tilde{\omega} S, \tilde{\omega} S_i)$ depends on the parameters $\vec{\tilde{\varepsilon}}$, $\tilde{\varepsilon}_1$, and $\tilde{\varepsilon}_2$.

$$B(\tilde{\omega}S, \tilde{\omega}S_i) = \begin{cases} 1, & \text{if the coordinates in } \tilde{\delta}(S, S_i) \\ & \text{numbered in } \Omega(\tilde{\omega} \longleftrightarrow \Omega) \text{ include} \\ & \text{at least } \tilde{\varepsilon}_1 \text{ coordinates of value 1} \\ & \text{and not greater than } \tilde{\varepsilon}_2 \\ & \text{coordinates of value 0} \\ 0 & \text{in all other cases.} \end{cases} \qquad (36)$$

The remaining steps in evaluating $\Gamma_j(S)$ are defined as in the previous case.

It is easy to see that the function $B^{\vec{\tilde{\varepsilon}}}$ defined in (36) is symmetric. If, again, the number of one coordinates in $\tilde{\delta}(S, S_i)$ is $q(S, S_i)$, then V^1 and V^0 are defined as in the first example.

Then

$$V^1 = \sum_{u=0}^{q(S,S_i)-\tilde{\varepsilon}_1} C^{\tilde{\varepsilon}_1+u-1}_{q(S,S_i)-1} \sum_{u=0}^{\tilde{\varepsilon}_2} C^u_{n-q(S,S_i)}, \qquad (37)$$

$$V^0 = \sum_{u=0}^{q(S,S_i)-\tilde{\varepsilon}_1} C^{u+\tilde{\varepsilon}_1}_{q(S,S_i)} \sum_{u=0}^{\varepsilon_2} C^{u-1}_{n-q(S,S_i)-1}. \qquad (38)$$

Having (37) and (38) and using the theorem, we can easily derive a formula for $\Gamma_j(S)$ ($j = 1, 2, \ldots, l$).

To demonstrate, we prove (37).

In addition to one fixed feature t, the reference subset should include either $\tilde{\varepsilon}_1 - 1$ or $\tilde{\varepsilon}_1, \ldots$, or $q(S, S_i)$ elements that correspond to coordinates of value 1 in the vector $\tilde{\delta}(S, S_i)$.

The choice is made from $q(S, S_i) - 1$ elements, the element t is fixed and the number of 1 coordinates in $\tilde{\delta}(S, S_i)$ is $q(S, S_i)$. The total number of possible choices is

$$\sum_{u=0}^{q(S,S_i)-\tilde{\varepsilon}_1} C^{\tilde{\varepsilon}_1+u-1}_{q(S,S_i)-1}. \qquad (39)$$

Each of the selected subsets can be supplemented with at most $\tilde{\varepsilon}_2$ elements from the number $n - q(S, S_i)$ of the elements that correspond to 0 coordinates in $\tilde{\delta}(S, S_i)$. The total number of possible adjunctions is

$$\sum_{u=0}^{\tilde{\varepsilon}_2} C^u_{n-q(S,S_i)}. \qquad (40)$$

Equations (39) and (40) readily lead to (37). Equation (38) is proved in the same way.

Having formulas (37) and (38) for V^1 and V^0, we can easily derive a formula for $\Gamma_j(S)$.

The generalization of the concept of the symmetric threshold function is the concept of a threshold function that is symmetrically with respect to partition.

Let $\{1, 2, \ldots, n\} = \bigcup_{i=1}^{v} N_i$ be the partition R of a set of features into subsets $N_1, \ldots, N_v$. Let there be given two subsets, Ω^1 and Ω^2, with characteristic vectors $\tilde{\omega}^1$ and $\tilde{\omega}^2$. The characteristic vectors of the subsets N_i are denoted by $\tilde{\omega}_i$ ($i = 1, 2, \ldots, v$). The subsets Ω_1 and Ω_2 are called *equivalent with respect to R and the descriptions I(S), I(S_u)* if the vectors $(\tilde{\omega}^1 \cdot \tilde{\omega}_i)$ and $(\tilde{\omega}^2 \cdot \tilde{\omega}_i)$ have the same number of 1 coordinates ($i = 1, 2, \ldots, v$).

Definition 18. The proximity function $B(\tilde{\omega} S, \tilde{\omega} S')$ is called symmetric with respect to the partition R if

$$B(\tilde{\omega}^1 S, \tilde{\omega}^1 S_u) = B(\tilde{\omega}^2 S, \tilde{\omega}^2 S_u).$$

Divide the feature set $\{1, 2, \ldots, n\}$ into equivalence classes with respect to the partition R: the features t' and t'' are R-equivalent if they both occur (do not occur) in each subset N_i ($i = 1, 2, \ldots, v$). The number of classes of R-equivalent features is denoted by $r(R)$.

Theorem 3. *If B is a threshold function symmetric with respect to the partition R and the system of reference subsets is correct, then the number of distinct values of $V_u(S, S_i)$ is not greater than $2r(R)$.*

Proof. It is only slightly more elaborate than that of Theorem 2. Let the features t' and t'' be R-equivalent and their corresponding coordinates in $\tilde{\delta}(S, S_i)$ be equal.

Consider an arbitrary reference set Ω', $t' \in \Omega'$, $t'' \bar{\in} \Omega'$. Since the system of reference sets is correct, $\Omega'' = (\Omega' \cup t'')\backslash t'$ is likewise a reference set. Establish one-to-one correspondence between the reference sets containing at least one of the elements t' or t''. If $\{t', t''\} \in \Omega$, then $\Omega \longleftrightarrow \Omega$. If $t' \in \Omega'$ and $t'' \bar{\in} \Omega'$, then $\Omega' \longleftrightarrow \Omega''$. The correspondence symbol signifies that t' and t'' occur in the same number of reference subsets. Since t' and t'' are R equivalent, they either both occur or both do not occur in each of N_i ($i = 1, 2, \ldots, v$).

Since the coordinates in $\tilde{\delta}(S, S_i)$ corresponding to t' and t'' are equal, the same number of 1 and 0 coordinates in $\tilde{\delta}(S, S_i)$ correspond to the elements of the sets $\Omega' \cap N_i$, $\Omega'' \cap N_i$. If $\Omega' \longleftrightarrow \tilde{\omega}'$ and $\Omega'' \longleftrightarrow \tilde{\omega}''$, then, because the proximity function B is symmetric with respect to R, we get

$$B(\tilde{\omega}'S, \hat{\omega}'S') = B(\tilde{\omega}''S, \tilde{\omega}''S'). \tag{41}$$

It readily follows from (41) that $V_{t'}(S, S_i) = V_{t''}(S, S_i)$. The latter equality, the condition that the number of R-equivalent features is $r(R)$, and the fact that the coordinates in $\tilde{\delta}(S, S_i)$ take at most two values easily prove the assertion of the theorem.

Corollary. *If the partition R consists of v disjoint subsets, then the number of distinct values of $V_t(S, S_i)$ is at most $2v$.*

The evaluation of $V_t(S, S_i)$ for correct subsets and threshold functions symmetric for $N_i, \ldots, N_v$ usually involves only technical difficulties. Let us calculate these values for two cases.

Let the system of reference subsets Ω_A^k be the collection of all the subsets of k elements and let the function $B(\tilde{\omega}S, \tilde{\omega}S_i)$ be symmetric with respect to the system of disjoint subsets and defined by the parameters $\varepsilon_1, \ldots, \varepsilon_n$ (they define the vector $\tilde{\delta}(S, S_i)$) and by the parameters ε_j^1, $j = 1, 2, \ldots, v$ as follows:

$$B(\tilde{\omega}S, \tilde{\omega}S_i) = \begin{cases} 1, & \text{if in } \tilde{\delta}(S, S_i) \text{ the coordinates} \\ & \text{numbered from } N_j \cap \Omega \text{ include} \\ & \text{at most } \varepsilon_j^1 \text{ coordinates of value 0,} \\ & j = 1, 2, \ldots, v \\ 0 & \text{in all other cases.} \end{cases} \tag{42}$$

The corollary of Theorem 3 implies that $V_t(S, S_i)$ takes at most $2v$ values. All the features u from N_j with the same coordinate α of the vector $\tilde{\delta}(S, S_i)$ have the same value of V_u. We denoted it by V_j^α ($\alpha = 0, 1$; $j = 1, 2, \ldots, v$).

Let us now derive formulas for V_j^0 and V_j^1.

Let N_j contain Δ_j elements of which Δ_j^1, Δ_j^0, and $\Delta_j^1 + \Delta_j^0 = \Delta_j$ are such that the coordinate in $\tilde{\delta}(S, S_i)$ numbered the same as an element in Δ_j^1 or Δ_j^0, takes value 1 or 0, respectively. Similarly, the number k_j of elements in $N_j \cap \Omega$ can be represented as $k_j^0 + k_j^1 = k_j$ (according to the values of the respective coordinates in $\tilde{\delta}$ ($j = 1, 2, \ldots, v$).

We say that the set $N_j \cap \Omega$ satisfies the condition Q_j, if the number of 0 coordinates in $\tilde{\delta}(S, S_i)$, numbered from $N_j \cap \Omega$, is not greater than ε_j^1 ($1 \le j \le v$).

Let us calculate the number $B(k_j, \Delta_j^1, \Delta_j^0, \varepsilon_j^1)$ of such sets. It can be done similarly to the calculations given in the example of an application for Theorem 2. Note that, if $\Delta_j^0 \le \varepsilon_j^1$, then

$$B(k_j, \Delta_1^j, \Delta_0^j, \varepsilon_i^1) = C_{\Delta_j}^{k_j}. \tag{43}$$

For $\Delta_j^0 > \varepsilon_j^1$,

$$B(k_j, \Delta_1^j, \Delta_0^j, \varepsilon_1^j) = \sum_{t=0}^{\varepsilon_j^1} C_{\Delta_j^1}^{k_j - t} C_{\Delta_j^0}^{t}. \tag{44}$$

Let now $B_\alpha(k_j, \Delta_j^0, \Delta_j^1, \varepsilon_j^1)$ be the number of the sets $N_j \cap \Omega$ that satisfy the condition Q_j and contain a feature u such that the uth coordinate in $\tilde{\delta}(S, S_i)$ is equal to α ($\alpha = 0, 1$).

It is easy to prove that

$$B_0(k_j, \Delta_j^0, \Delta_j^1, \varepsilon_j^1) = \sum_{t=0}^{\varepsilon_j^1} C_{\Delta_j^1}^{k_j - t} C_{\Delta_j^0 - 1}^{t-1}, \qquad (45)$$

$$B_1(k_j, \Delta_j^0, \Delta_j^1, \varepsilon_j^1) = \sum_{t=0}^{\varepsilon_j^1} C_{\Delta_j^1 - 1}^{k_j - t - 1} C_{\Delta_j^0}^{t}. \qquad (46)$$

It is equally obvious that $k = k_1 + \ldots + k_v$.

In view of (43) through (46), we get

$$V_j^\alpha = \sum_{\substack{(k_1, \ldots, k_v): \\ k_1 + \ldots + k_v = k \\ k_u \le \Delta_u \\ \Delta_u^0 + \Delta_u^1 = \Delta_u}} \prod_{\substack{i \ne j \\ 1 \le i \le v}} B(k_i, \Delta_i^0, \Delta_i^1, \varepsilon_i^1) \times B_\alpha(k_j, \Delta_j^0, \Delta_j^1, \varepsilon_j^1).$$

Similarly, we can derive formulas for V_j^α when the system Ω_A of reference subsets in the algorithm A consists of nonempty subsets in the set $\{1, 2, \ldots, n)$, and the function $B(\tilde{\omega} S, \tilde{\omega} S_i)$ is defined by the parameters $\varepsilon_1, \ldots, \varepsilon_n, \varepsilon_1^1, \varepsilon_1^2, \ldots, \varepsilon_1^v, \varepsilon_2^v$ and is symmetric with respect to the system of disjoint subsets $N, \ldots, N_v$. Suppose

$$B(\tilde{\omega} S, \tilde{\omega} S_i) = \begin{cases} 1, & \text{if, in the vector } \tilde{\delta}(S, S_i) \text{ the coordinates numbered in } \Omega_A \cap N_j \text{ include at least } \varepsilon_j^1 \text{ coordinates of value 1 and not more than } \varepsilon_j^2,\ j = 1, 2, \ldots, v \\ 0 & \text{in all other cases.} \end{cases} \qquad (47)$$

As before, we use Δ_j, Δ_j^0, and Δ_j^1 to denote the number of elements in N_j and the number of elements in N_j with 0 and 1 coordinates, respectively, in the vector $\tilde{\delta}(S, S_i)$ $(j = 1, 2, \ldots, v)$.

Theorem 4.

$$V_j^0 = N \frac{\sum_{k=\varepsilon_j^1}^{\Delta_j^1} C_{\Delta_j^1}^k \sum_{k=0}^{\varepsilon_j^2 - 1} C_{\Delta_j^0 - 1}^k}{\sum_{k=\varepsilon_j^1}^{\Delta_j^1} C_{\Delta j 1}^k \sum_{k=0}^{\varepsilon_j^2} C_{\Delta_j^0}^k},$$

$$V_j^1 = N \frac{\sum_{k=\varepsilon_j^1 - 1}^{\Delta_j^1 - 1} C_{\Delta_j^1 - 1}^k \sum_{k=0}^{-\varepsilon_j^2} C_{\Delta_j^0}^k}{\sum_{k=\varepsilon_j^1}^{\Delta_j^1} C_{\Delta_j^1}^k \sum_{k=0}^{\varepsilon_j^2} C_{\Delta_j^0}^k},$$

$$N = \prod_{l=1}^{v} \left(\sum_{k=\varepsilon_l^1}^{\Delta_l^1} C_{\Delta_l^1}^k \sum_{k=0}^{\varepsilon_l^2} C_{\Delta_l^0}^k \right).$$

Proof. Note that, if $\Delta_j^1 < \varepsilon_j^1$ for at least one j $(j = 1, 2, \ldots, v)$, then there is no reference Ω such that $B(\tilde{\omega} S, \tilde{\omega} S_i) \ne 0$. In this case $V_j^\alpha = 0$, $\alpha = 0, 1$ $(j = 1, 2, \ldots, v)$.

Therefore, in what follows, we assume that $\Delta_j^1 \ge \varepsilon_j^1$ $(j = 1, 2, \ldots, v)$. We say that $\Omega \cap N_j$ satisfies the condition P_j if the vector $\tilde{\delta}(S, S_i)$ has at least ε_j^1 coordinates from $\Omega \cap N_j$ that take value 1 and not more than ε_j^2 such coordinates that take value 0. The number of the sets that satisfy P_j is denoted by $D(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^1, \Delta_j^0)$.

Here, Δ_j^1 and Δ_j^0 denote the coordinates in $\tilde{\delta}(S, S_i)$ that take value 1 and 0, respectively, numbered from N_j $(j = 1, 2, \ldots, v)$.

Fix a feature u in $\Omega \cap N_j$, form the set $(\Omega \cap N_j) \backslash u = \Omega(j, u)$ and introduce the properties P_j^0 and P_j^1 on it.

$P_j^0(\Omega(j, u))$ is the number of coordinates in $\tilde{\delta}(S, S_i)$, at least ε_j^1 of value 1, numbered in $\Omega(j, u)$ and not greater than $\varepsilon_j^2 - 1$ such coordinates of value 0.

The property $P_j^1(\Omega(j, u))$ implies that there are in $\Omega(j, u)$ at least $\varepsilon_j^1 - 1$ coordinates in $\tilde{\delta}(S, S_i)$ of value 1 and not greater than ε_j^2 coordinates of value 0.

The number of the sets Ω that satisfy the property P_j and of the sets $\Omega(j, u)$ that satisfy the properties P_j^0 or P_j^1 is denoted by $D(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^0, \Delta_j^1)$, $D_j^0(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^0, \Delta_j^1)$, and $D_j^1(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^0, \Delta_j^1)$, respectively. Similarly, D_j^{0t} or D_j^{1t} denote the number of the sets $\Omega(j, u)$ that satisfy the property P_j^0 or P_j^1 and an additional condition that the uth coordinate in $\tilde{\delta}(S, S_i)$ is t, $t \in \{0, 1\}$.

It is easy to show that all D_j^{0t} (and, similarly, D_j^{1t}) are equal for all these features u.

The equality

$$\begin{aligned} V_j^t &= \prod_{l \in \{1, 2, \dots, v\} \backslash j} D(\varepsilon_l^1, \varepsilon_l^2, \Delta_l^1, \Delta_l^0) D_l^{tt}(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^1, \Delta_j^0) \\ &= \prod_{l=1}^{v} D(\varepsilon_l^1, \varepsilon_l^2, \Delta_l^1, \Delta_l^2) \frac{D_j^{tt}(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^1, \Delta_j^0)}{D(\varepsilon_j^1, \varepsilon_j^2, \Delta_j^1, \Delta_j^0)} \end{aligned} \qquad (48)$$

holds for all features u in N_j such that the uth coordinate of $\tilde{\delta}(S, S_i)$ is t ($j = 1, 2, \dots, v$, $t = 0, 1$).

It is easy to calculate directly that

$$\begin{aligned} D(\varepsilon_l^1, \varepsilon_l^2, \Delta_l^1, \Delta_l^2) &= \left(C_{\Delta_j^1}^{\varepsilon_j^1} + C_{\Delta_j^1}^{\varepsilon_j^1 + 1} + \dots + C_{\Delta_j^1}^{\Delta_j^1}\right) \\ \times \left(C_{\Delta_j^0}^{\varepsilon_j^2} + C_{\Delta_j^0}^{\varepsilon_j^2 - 1} + \dots + C_{\Delta_{j_0}}^{\Delta_{j_0}}\right) &= \sum_{k = \varepsilon_j^1}^{\Delta_j^1} C_{\Delta_j^1}^k \sum_{k=0}^{\varepsilon_j^2} C_{\Delta_j^0}^k. \end{aligned} \qquad (49)$$

Similarly,

$$D_j^{00} = \sum_{k = \varepsilon_j^1}^{\Delta_j^1} C_{\Delta_j^1}^k \sum_{k=0}^{\varepsilon_j^2 - 1} C_{\Delta_j^0 - 1}^k. \qquad (50)$$

We can calculate D_j^{11} in the same way.

The substitution of D_j^{00}, D_j^{11} and D in (48) proves the theorem.

In simple cases, the formulas for $\Gamma_j(S)$ are not complex. Thus, if $\|\tilde{\delta}(S, S_i)\|$ denotes the number of unities in $\tilde{\delta}(S, S_i)$, and we consider the function $B(\tilde{\omega} S, \tilde{\omega} S_i)$ that is equal to 1 if and only if the coordinates ω that take value 1 in $\tilde{\omega}(S, S_i)$ are included among the coordinates that take value 1 in $\tilde{\delta}(S, S_i)$ (all inequalities $\rho_\omega \leq \varepsilon_\omega$ are satisfied), then:

1°. If the system of reference subsets consists of all k-element subsets in the set $\{1, 2, \dots, n\}$,

$$\Gamma_j(S) = \frac{1}{\mu(W_j^1)} \sum_{S_i \in W_j^1} \gamma(S_i) C_{\|\tilde{\delta}(S, S_i)\|}^k,$$

2°. If the system of reference subsets consists of all nonempty subsets in the set $\{1, 2, \dots, n\}$, then

$$\Gamma_j(S) = \frac{1}{\mu(W_j^1)} \sum_{S_i \in W_j^1} \gamma(S_i)(2^{\|\tilde{\delta}(S, S_i)\|} - 1).$$

Here, $W_j^1 = K_j \cap \{S_1, \dots, S_m\}$ and $I(S_u) \in I_0$ ($u = 1, 2, \dots, m$).

II. The second principle underlying formulas for $\Gamma_j(S)$ draws upon the fact that the characteristic function $f_\Omega^A(\tilde{\omega})$ of the system Ω_A of the reference sets of A can be given a special representation. It is well-known [74] that any Boolean function $f, f \neq 0$ (and so, $f_\Omega^A(\tilde{\omega})$) can be represented as a DNF composed of orthogonal elementary conjunctions, that is

$$f_\Omega^A(\tilde{\omega}) = K_1 \vee \dots \vee K_r, \quad K_i K_j \equiv 0 \text{ for } i \neq j, \qquad (51)$$

$$K_\iota = x_{i_1}^{\sigma_{\iota_1}} \cdot \dots \cdot x_{i_q}^{\sigma_{\iota_q}}, \quad x^\sigma = \begin{cases} x, & \sigma = 1 \\ \bar{x}, & \sigma = 0. \end{cases}$$

The function $f_\Omega^A(\tilde{\omega})$ is denoted below by $f(x_1 \dots x_n)$ or simply by f. It is also known that any representation (51) corresponds to the partition of the set N_f of unities in the function into a system of disjoint intervals $N_{K_1}, \dots, N_{K_r}$ (where N_{K_ι} is the set of ones in the conjunction K_i); that is,

$$N_f = N_{K_\iota} \cup \dots \cup N_{K_r}, \; N_{K_\iota} \cap N_{K_j} \qquad (52)$$

is empty for $i \neq j$.

Let $\tilde{\omega}_A$ be the collection of characteristic vectors for the system Ω_A of reference subsets of the algorithm A. A partition is specified such that $\tilde{\omega}_A = \bigcup_{i=1}^r \{\tilde{\omega}\}_\iota$ (consequently, $\Omega_A = \bigcup_{i=1}^r \{\Omega\}_i$) into a collection of disjoint subsets $\{\tilde{\omega}\}_i$ (and, respectively, $\{\Omega\}_i$ ($i = 1, 2, \dots, r$)).

Consider a collection of estimates calculation algorithms $A_1, \dots, A_r$ with systems of reference sets $\{\Omega\}_1 = \Omega_{A_1}, \dots, \{\Omega\}_r = \Omega_{A_r}$, in which the remaining steps, II through VI, are defined identically, such that

$$\begin{aligned} \Gamma_j^{A_u}(S) &= \frac{1}{N} \cdot \frac{1}{\mu(W_j^1)} \\ &\times \sum_{S \in W_j^1} \sum_{\tilde{\omega} \in \{\tilde{\omega}\}_\iota} \varphi(B(\tilde{\omega} S, \tilde{\omega} S_i, \pi_1, \dots, \pi_t)), \end{aligned} \qquad (53)$$

here $\varphi(B, \pi_1, \dots, \pi_t)$ is an arbitrary numerical function defining the estimates $\Gamma(\tilde{\omega} S, \tilde{\omega} S_i)$ from the proximity function and the fixed values of numerical parameters $\pi_1, \dots, \pi_t$.

Thus, we previously considered $(\vec{p}, \vec{\omega})$, the scalar product of the numerical vector $(p_1 \dots p_n)$ by $\tilde{\omega}$ as φ: $\gamma(S_i)(\vec{p}, \vec{\omega}) B(\tilde{\omega} S, \tilde{\omega} S_i)$, B being a threshold proximity function. We have

Theorem 5 [14]

$$\Gamma_j^A(S) = \sum_{u=1}^{r} \Gamma_j^{A_u}(S).$$

Proof. It is easy to see that

$$\Gamma_j^A(S) = \frac{1}{N}\cdot\frac{1}{\mu(W_j^1)}$$

$$\times\sum_{S_i\in W_j^1}\sum_{\Omega\in\Omega_A}\varphi(B(\tilde\omega S, \tilde\omega S_i), \pi_1, \ldots, \pi_t)$$

$$=\sum_{u=1}^{r}\left(\frac{1}{N\mu(W_j^1)}\sum_{S_i\in W_j^1}\sum_{\Omega\in\{\Omega\}_i}\varphi(B(\tilde\omega S, \tilde\omega S_i), \pi_1, \ldots, \pi_t)\right)$$

$$=\sum_{u=1}^{r}\Gamma_j^{A_u}(S).$$

The theorem is proved.

Let the algorithms A_1, A_2, A, and B be defined by the systems of reference subsets Ω_1, Ω_2, $\Omega_1\cup\Omega_2$, $\Omega_{12} = \Omega_1\cap\Omega_2$, and $\Gamma_j(S)$ are estimated by formula (53) where $\{\tilde\omega\}_i$ is replaced by Ω_1, Ω_2, $\Omega_1\cup\Omega_2$, and Ω_{12}, respectively. It is easy to prove (the proof is omitted)

Theorem 6 [14]

$$\Gamma_j^A(S) = \Gamma_j^{A_1}(S)+\Gamma_j^{A_2}(S)-\Gamma_j^B(S).$$

It follows from Theorems 5 and 6 that, if for some basis subsets of characteristic vectors (and the corresponding systems of reference sets) the estimates $\Gamma_j(S)$ are evaluated by the formulas derived above, it is possible to obtain formulas by representing arbitrary systems of reference sets as the sums of basis subsets.

It turns out [74] that, if the collection $\{\tilde\omega\}_A$ of characteristic vectors forms the interval N_K $(K = x_{i_1}^{\sigma_1}\cdot\ldots\cdot x_{i_t}^{\sigma_t})$ and proximity function B is a threshold function, then it is likewise possible to formulate a general method for the synthesis of formulas for $\Gamma_j(S)$. Then, using the representations in (51) and (52) and Theorems 5 and 6, we can develop formulas for algorithms with fairly diverse systems of reference sets. The development of a formula for $\{\tilde\omega\}_A = N_K$ $(K = x_{i_1}^{\sigma_1}\cdot\ldots\cdot x_{i_t}^{\sigma_t})$ follows.

The conjunction K turns to unity on any $\tilde\omega = (\omega_1\ldots\omega_n)$ such that $\omega_{i_1} = \sigma_1, \ldots, \omega_{i_t} = \sigma_t$ and the remaining coordinates are arbitrary. To simplify the notation and without loss of generality, we can assume that $K = \bar{x}_1\cdot\ldots\cdot\bar{x}_r\cdot x_{r+1}\cdot\ldots\cdot x_t$ (this form can always be achieved by rearranging the variables). Clearly, the system $\Omega_A = N_K$ includes all the subsets that do not contain $\{1, 2, \ldots, r)$ and include $\{r+1, \ldots, t\}$. Let us evaluate separately

$$\Gamma_j^0(S) = \frac{1}{\mu(W_j^1)}\sum_{S_i\in W_j^1}\Gamma(\tilde\omega S, \tilde\omega S_i),$$

$$\Omega = \{r+1, \ldots, t\}.$$

Consider the function $B^{\vec{\tilde\varepsilon}}(\tilde\varepsilon_1, \tilde\varepsilon_2)$ in (36) analyzed above. Formulas of V^1 and V^0 were obtained for it in (37) and (38). Using the formulas and Theorem 1, we can easily write a formula for $\Gamma_j(S)$.

We denote by Δ_{t-r}^1 and Δ_{t-r}^0 the coordinates of value 1 and 0, respectively, of the vector $\tilde\delta(S, S_i)$, which are numbered from $\{r+1, \ldots, t\}$. Then $B^{\vec{\tilde\varepsilon}}(\tilde\varepsilon_1, \tilde\varepsilon_2) = 1$ on $(\tilde\omega S, \tilde\omega S_i)$ if and only if: (1) the number of unities among the coordinates of $\tilde\delta(S, S_i)$, numbered from $\{t+1, \ldots, n\}$, is at least $\tilde\varepsilon_1-\Delta_{t-r}^1$ and (2) the number of zeroes among these coordinates does not exceed $\tilde\varepsilon_2-\Delta_{t-r}^0$.

So, we reduced the problem to the one solved previously by the substitution: $n \longrightarrow n-t$, $\tilde\varepsilon_1 \longrightarrow \tilde\varepsilon_1-\Delta_{t-r}^1$, $\tilde\varepsilon_2 \longrightarrow \varepsilon_2-\Delta_{t-r}^0$. If we substitute them in (37) and (38), include the term $\Gamma_j^0(S)$ and apply Theorem 1, we get

Theorem 7.

$$\Gamma_j(S) = \frac{1}{N\mu(W_j^1)}\left(\Gamma_j^0(S)+\sum_{S_i\in W_j^1}\gamma(S_i)((\tilde\delta(S, S_i)\cdot\vec{p})\right.$$

$$\times\sum_{u=0}^{\tilde q(S, S_i)-\tilde\varepsilon_1+\Delta_{t-r}^1}C_{\tilde q(S, S_i)}^{\tilde\varepsilon_1-\Delta_{t-r+u-1}^1}\sum_{u=0}^{\tilde\varepsilon_2-\Delta_{t-r}^0}C_{n-\tilde q(S, S_i)}^{u}$$

$$\left.+(\bar{\tilde\delta}(S, S_i)\cdot\vec{p})\sum_{u=0}^{\tilde q(S, S_i)-\tilde\varepsilon_1+\Delta_{t-r}^1}C_{q(S, S_i)}^{\tilde\varepsilon_1-\Delta_{t-r}^1+u}\sum_{u=0}^{\tilde\varepsilon_2-\Delta_{t-r}^0}C_{n-\tilde q(S, S_i)}^{u-1})\right),$$

where $\tilde q(S, S_i)$ is *the number of the unities among the coordinates of* $\tilde\delta(S, S_i)$, *which are numbered from* $\{t+1, \ldots, n)$.

Generally, once a formula was obtained for $\Gamma_j(S)$ in an algorithm with the system Ω_A of reference subsets identical to all the subsets $\{1, 2, \ldots, n)$, we can use a method similar to the one described in proving Theorem 1 to derive a formula for the case $\{\tilde\omega\}_A = N_K$.

SECTION II. OTHER MODELS OF RECOGNITION ALGORITHMS

§ 5. Algorithm Models Based on the Separation Principle (*R* Models) [10]

Algorithms based on construction of the class separating surfaces appeared before any other algorithms. They are mainly applied to problems with the learning information $I_0(K_1, \ldots, K_l)$ defined in standard form with numerical features.

Consider first a problem with two disjoint classes, K_1 and K_2. It is postulated explicitly or implicitly that there is a surface R of a fairly simple form, which separates the elements of K_1 and K_2, so, for example, that

$$\text{if } S \in K_1 \text{ and } I(S) = (a_1, \ldots, a_n) = \tilde{a}, \text{ then } R(\tilde{a}) > 0;$$

$$\text{if } S' \in K_2 \text{ and } I(S') = (b_1, \ldots, b_n) = \tilde{b}, \text{ then } R(\tilde{b}) \le 0.$$

Algorithms differ in the set $\{R\}$ of surfaces, from which the separating surface is selected, and in how the surface is selected. Most often, the set $\{R\}$ is the class of hyperplanes, $\alpha_1 x_1 + \ldots + \alpha_n x_n + \alpha_{n+1} = 0$, the class of piecewise linear surfaces, and the class of second-order surfaces.

If all features in the descriptions of S are binary, then the collection of elements in the first and second class is identified with the set of unities and zeroes, respectively, of a not everywhere defined Boolean function $F(x_1, \ldots, x_n)$ (its domain coincides with the set of descriptions $I(S_1), \ldots, I(S_m)$ in standard information) [5, 6]. The arbitrary logical function $\Phi(x_1, \ldots, x_n)$ that implements F is a separating function.

The object S in this case is assigned to K_1 or to K_2 if $F(I(S)) = 1$ or $F(I(S)) = 0$, respectively.

A similar procedure is followed in constructing the class of separating surfaces if the features 1, 2, ..., n have each not more than k gradations and if the set of admissible objects is separated into no more than k disjoint classes. Then, similar to the Boolean case, a not everywhere defined function $F(x_1, \ldots, x_n)$ of k-valent logic is developed, which takes $j - 1$ values on the descriptions $I(S)$ of objects which are members of K_j ($j = 1, 2, \ldots, l$; $l \le k$). Then the separator is an arbitrary formula of k-valent logic, which implements $F(x_1, \ldots, x_n)$. It is natural to select a disjunctive normal form (in the case of two classes) or its equivalent in k-valent logic to act as the corresponding formula.

The idea of constructing the separating surface by the successive approximation method is the basis of the method of potential functions [4]. It embraces a fairly broad class of surfaces as the class of possible separating surfaces.

Once the class $\{R\}$ of separating surfaces is fixed, one of the surfaces is generally selected by solving an extremum problem. For example, we can select a surface that produces the best recognition accuracy on a specified control set.

The principal conclusion that can be drawn from the analysis of various R models is this. Any of these algorithms takes two stages to execute. Let a recognition problem be solved for admissible objects $S'_1, \ldots, S'_q$.

1. Using learning information, set up a numerical matrix $\{R_{ij}\}_{q \times l}$. The element R_{ij} can be interpreted as a value of the membership function of S'_i in K_j ($i = 1, 2, \ldots, q$; $j = 1, 2, \ldots, l$).

2. Using the matrix $\{R_{ij}\}_{q \times l}$ and the decision rule of the algorithm, set up the matrix $\{\alpha_{ij}\}_{q \times l}$ of information vectors for the elements $S'_1, \ldots, S'_q$, such that $\alpha_{ij} \in \{0, 1, \Delta\}$. If the transformation of I_0 into $\{R_{ij}\}_{q \times l}$ is regarded as the application of the operator R_A and the transformation of $\{R_{ij}\}_{q \times l}$ into $\{\alpha_{ij}\}_{q \times l}$ as the application of the other operator r_A, then an arbitrary algorithm in the R model is representable as $A = (R_A(I_0(l), I_S(q))r_A\{R_{ij}\}_{q \times l})$. Here, the product of the operators is taken to mean their successive application.

The basic R models are as follows.

I. $\mathfrak{M}(a_1, \ldots, a_n, a_{n+1})$. The separation is made by the hyperplane $a_1 x_1 + \ldots + a_n x_n + a_{n+1} = 0$. The model is defined by giving the parameters $a_1, \ldots, a_{n+1}$ and a decision rule.

II. $\mathfrak{M}_L^k$. The separation is made by a piecewise-linear surface. Any separating surface R divides the space of descriptions $I(S)$ into two subsets: $M^+(R) = \{I(S): R(I(S)) > 0\}$ and $M^-(R) = \{I(S): R(I(S)) < 0\}$. The notation $R(I(S)) \lesseqgtr 0$ is conventional.

III. The models $\mathfrak{M}^1(l)$ differ from the models in **I** and **II** above in that the separation is made by surfaces $R_1, \ldots, R_l$ such that the surface R_j separates the objects that are members of K_j from the objects that are not members of K_j ($j = 1, 2, \ldots, l$). The operators R_A are introduced in the same way as in **I** and **II**.

IV. $\mathfrak{M}^1(\tilde{\gamma})$.

Let there be given a standard learning information

$$I_0 = (I(S_1), \tilde{\alpha}(S_1), \ldots, I(S_m), \tilde{\alpha}(S_m))$$

and descriptions $I(S'_1), \ldots, I(S'_q)$ of admissible objects. Each S_i is assigned a parameter $\gamma(S_i) = \gamma_i > 0$. The separating surface R is picked from the class $\{R\}$.

Consider the predicate $P_R(S_i)$:

$$P_R(S_i) = \begin{cases} 1, & \text{if } R(I(S_i)) > 0 \\ 0, & \text{if } R(I(S_i)) \le 0. \end{cases}$$

Each S_i is assigned a pair (α, β), where $\alpha = P_R(S_i)$ and $\beta = \alpha_{ij}$ is the jth coordinate in the information vector $\tilde{\alpha}(S_i)$.

The set $S_1, \ldots, S_m$ is partitioned into four disjoint subsets S_j^{00}, S_j^{01}, S_j^{10}, and S_j^{11} of the elements S_t that are assigned the pairs (α, β) equal to (0, 0), (0, 1), (1, 0) and (1, 1), respectively. Suppose

$$Q_{11}^j = \sum_{S_t \in S_j^{11}} \gamma_t, \quad Q_{10}^j = \sum_{S_t \in S_j^{10}} \gamma_t,$$

$$Q_{01}^j = \sum_{S_t \in S_j^{01}} \gamma_t, \quad Q_{00}^j = \sum_{S_t \in S_j^{00}} \gamma_t.$$

Assign S_i' a numerical vector $(r_{i1}, \ldots, r_{il})$ as follows.

$$\begin{aligned} &\text{If } R(I(S_i')) > 0, \text{ then } R_{ij} = \frac{Q_{11}^j + Q_{00}^j}{Q_{10}^j + Q_{01}^j + 1}, \\ &\text{if } R(I(S_i')) \le 0, \text{ then } R_{ij} = \frac{Q_{01}^j + Q_{10}^j}{Q_{11}^j + Q_{00}^j + 1}. \end{aligned} \qquad (54)$$

The conditions in (54) define the operator R_A that transforms I_0 into a matrix $\{R_{ij}\}_{q \times l}$.

V. The model $\mathfrak{M}^1(\tilde{\gamma}, l)$ is similar to the model in IV above and differs in that the surface R_{ij} for each class K_j is found relative to the surface R_j $(j = 1, 2, \ldots, l)$.

§ 6. Statistical and Potential-Type Models

I. Statistical models. We do not dwell on these models in detail. What is important for the purposes of this paper is that their algorithms transform the learning information I_0 and the descriptions of admissible objects, $I(S_1' \ \ldots \ S_q')$; that is, the pair $(I_0(l), I_S(q))$, into a numerical matrix, $\{p_{ij}\}_{q \times l}$.

Then from the elements of this matrix we develop information vectors $\tilde{\alpha}_i^A(S) = (\alpha_{i1} \ldots \alpha_{il})$ for objects S_i'. The elements p_{ij} can be, for example, the probabilities of S_i' being members of K_j $(j = 1, 2, \ldots, l, i = 1, 2, \ldots, q)$ or some other statistical measures. We need only to state the fact that, as with other models, their statistical counterparts involve two stages:

1°. The operator R_A is applied to transform $(I_0(l), I_S(q))$ into the numerical matrix $\{p_{ij}\}_{q \times l}$.

2°. The decision rule r_A is applied to transform $\{p_{ij}\}_{q \times l}$ into the information vector matrix $\{ \alpha_{ij}^A \}$ $(i = 1, 2, \ldots, q, j = 1, 2, \ldots, l)$.

II. Algorithms of the potential function type. Consider the case of numerical features and assume that the original information is specified in standard form.

Among $S_1, \ldots, S_m$, we pick the set W_j^1 of objects which are members of K_j. The algorithms are specified as follows.

Consider a collection of functions, $\{F\}$. The potential of S_i' with respect to K_j is the quantity $F(I(W_j^1), I(S_i'))$, where $I(W_j^1)$ is the collection of descriptions for objects out of W_j^1.

The following set of functions can be considered as $\{F\}$. Let $\rho(I(S), I(S_i'))$ be the distance in the set of descriptions of admissible objects and $\gamma(S_t) = \gamma_t$ be the mass of an object S_t. Put

$$\rho(I(S_i'), I(W_j^1)) = \sum_{S_t \in W_j^1} \gamma_t \rho(I(S_t), I(S_i')).$$

Then $\{F\} = \{e^{-\alpha\rho(x, y)}\}$ or $\{F\} = \left\{\frac{c_1}{c_2\rho(x, y) + c_3}\right\}$. (It is possible to specify the class $\{F\}$ in other ways.) The potential of S_i' relative to K_j, respectively, is

$$e^{-\alpha \sum\limits_{S_t \in W_j^1} \nu_t \rho(I(S_t), I(S_i'))}, \qquad \frac{c_1}{c_2 \sum\limits_{S_t \in W_j^1} \rho(I(S_t), I(S_i')) + c_3}.$$

In the general case, the potential of S_i' relative to K_j (the set W_j^1) is denoted by Π_{ij}.

Potential-type algorithms are executed likewise in two stages.

1. At the first stage, the potential matrix $\{\Pi_{ij}\}_{q \times l}$ is set up using the given control sample $S_1', \ldots, S_q'$. I.e., the operator R_A acts to transform $(I_0(l), I_S(q))$ into the numerical matrix $\{\Pi_{ij}\}_{q \times l}$.

2. The collection of potentials $\Pi_{i1}, \ldots, \Pi_{il}$ (or the entire matrix $\{\Pi_{ij}\}$) is inspected and the decision is made as to the membership of S_i' in K_j $(j = 1, \ldots, l)$. In other words, the information vectors matrix $\{ \alpha_{ij}^A \}_{q \times l}$ is derived from the potential matrix $\{\Pi_{ij}\}_{q \times l}$, i.e., the decision rule r_A is applied and $r_A\{\Pi_{ij}\} = \{ \alpha_{ij}^A \}_{q \times l}$.

The statistical and potential-type algorithms and their subclasses can be described using various models. The underlying principles of the models are similar to those used in describing V (estimates calculation) models or R models.

§ 7. Structural Recognition Algorithms [33, 78]

The basic scheme of structural recognition methods was described in the Introduction. We confine ourselves, therefore, to an analysis of one example.

Suppose we are given a finite alphabet $\mathfrak{A} = \{\Lambda, a_1, \ldots, a_n\}$ and descriptions of objects to be classified, which are phrases in Σ (Definition 5).

The class $\{\psi\}_j$ is composed of elementary predicates, $Q_{jt}^{f, 1}(S)$: the membership function (estimate) of an object in the class K_j is greater than the membership function (estimate) of S in K_t.

In addition to $Q_{jt}^{f,1}(S)$, the class $\{\psi\}_j$ includes the predicates $\overline{Q}_{jt}^{f,1}(S)$: the estimate of S for K_j is not greater than that for K_t.

In addition to the above predicates $Q_{jt}^{f,1}$ and $\overline{Q}_{jt}^{f,1}$, all their logical products are also included.

The set of formulas, $\{\Phi\}_j$, is composed of elementary predicates $P(r, a_i, S)$, which implies that the rth character in the phrase S is a_i ($r = 1, 2, \ldots, t, \ldots, i = 1, 2, \ldots, n$).

Let it also be known that the classes K_j ($j = 1, 2, \ldots, l$) are disjoint.

The basis elements $\{B\}^j$ were used in operations $\{O\}^j$ to construct the set M_j of descriptions of objects in K_j. The numerical parameters δ_{ut}^1 and δ_{ut}^2 ($u = 1, 2, \ldots, l$, $t \neq u$) were introduced.

Consider the description S of an object to be recognized, $S \overline{\in} M_j$ ($j = 1, 2, \ldots, l$).

Suppose the rth character in S is a_i. We denote the number of words in M_j, which contain the rth character a_i by $Q_j(r, a_i)$ and that for M_t by $Q_t(r, a_i)$.

If $Q_j(r, a_i) - Q_t(r, a_i) > \delta_{jt}^1$, we include the formula $P(r, a_i, S) \longrightarrow Q_{jt}^{f,1}$ in $\{\chi\}_j$.

If $Q_t(r, a_i) - Q_j(r, a_i) > \delta_{jt}^2$, we include the formula $P(r, a_i, S) \longrightarrow \overline{Q}_{jt}^{f,1}$ in $\{\chi\}_j$.

If $\{\chi\}_j$ includes the formulas $\Phi_1 = (P(k, a_u, S) \longrightarrow \tilde{Q}_{jt}^{f,1})$ and $\Phi_2 = (P(k, a_u, S) \longrightarrow \tilde{Q}_{j\nu}^{f,1})$, then $\{\chi\}_j$ also includes the formula $P(k, a_u, S) \longrightarrow \tilde{Q}_{jt}^{f,1}, \tilde{Q}_{ju}^{f,1}$.

The symbol $\tilde{Q}$ signifies that $\tilde{Q} = Q$ or $\tilde{Q} = \overline{Q}$.

We introduce the parameters $x_{tj}^1(r, a_i)$ and $x_{tj}^2(r, a_i)$ ($j = 1, 2, \ldots, l$, $t = 1, 2, \ldots, j-1, j+1, \ldots, l$).

Suppose $S = (b_1 b_2 \ldots b_q)$. We assign to b_i ($i = 1, 2, \ldots, q$) the number

$$\Gamma_j(b_i, S) = \begin{cases} 0, \text{ if } b_i = \Lambda \text{ or } \{\chi\}_j \text{ has} \\ \quad \text{no formulas containing } b_i \\ \sum\limits_t x_{tj}^{\alpha} \text{ in all other cases.} \end{cases}$$

The summation is taken over t numbers for which $\{\chi\}_j$ includes the formula $P(i, b_i S) \longrightarrow \tilde{Q}_{jt}^{f,1}$. If $\tilde{Q}_{jt}^{f,1} = Q_{jt}^{f,1}$, then $\alpha = 1$. If $\tilde{Q}_{jt}^{f,1} = \overline{Q}_{jt}^{f,1}$, then $\alpha = 2$.

The object S is given the estimate

$$\Gamma_j(S) = \frac{1}{q}\sum_{i=1}^{q}\Gamma_j(b_i, S)\gamma_i.$$

The quantities $\gamma_1, \ldots, \gamma_q, \ldots$ are numerical parameters.

We described a model of structural recognition algorithms with the parameters δ_{ut}^1, δ_{ut}^2, $x_{tj}^1(r, a_i)$, $x_{tj}^2(r, a_i)$, and γ_i.

This model transforms the structural learning information and the descriptions of the objects $S_1, \ldots, S_q$ to be recognized into a numerical estimation matrix, $\{\Gamma_{ij}\}_{q \times l}$. The model parameters can be chosen for prior considerations or sought as a solution to an extremum problem.

§ 8. Decision Rules in Recognition Algorithms

The decision rule r_A of a recognition algorithm A is the operator that transforms the estimation matrix $\{\alpha_{ij}\}_{q \times l}$ into the matrix $\{\alpha_{ij}^A\}_{q \times l}$ of the information vectors of the elements to be recognized. In what follows, we review three basic types of decision rules.

I. Functional decision rules r_A of order 1. One evaluates α_{ij}^A as follows. Introduce a numerical function $f(x)$. In the set M_f of values of $f(x)$, three disjoint sets M_f^{Δ}, M_f^1, and M_f^0 are selected:

$$\alpha_{ij}^A(a_{ij}) = \begin{cases} \Delta, \text{ if } f(a_{ij}) \in M_f^{\Delta} \\ 1, \text{ if } f(a_{ij}) \in M_f^0 \\ 0, \text{ if } f(a_{ij}) \in M_f^1. \end{cases} \qquad (55)$$

The function $f(x)$ is the basis of the decision rule r_A.

II. Functional decision rules of order l. One specifies l functions $f_1(x_1, \ldots, x_l), \ldots, f_l(x_1, \ldots, x_l)$. These functions form the basis of the decision rule. The functions f_j ($j = 1, 2, \ldots, l$) are not necessarily distinct.

As in I above, the set Mf_j of values of f_j is partitioned into three subsets, $M_{f_j}^{\Delta}$; $M_{f_j}^1$; and $M_{f_j}^0$ such that

$$\alpha_{ij}^A = \begin{cases} \Delta, \text{ if } f_j(a_{i1}, \ldots, a_{il}) \in M_{f_j}^{\Delta} \\ 0, \text{ if } f_j(a_{i1}, \ldots, a_{il}) \in M_{f_j}^0 \\ 1, \text{ if } f_j(a_{i1}, \ldots, a_{il}) \in M_{f_j}^1. \end{cases} \qquad (56)$$

Another class of decision rules of order l is specified by a basis consisting of a single function, $f(x_1, \ldots, x_l)$. Let $\{\tilde{\alpha}\}$ be a set of information vectors of length l and suppose that the subsets $M_f^{\tilde{\alpha}}$, $\tilde{\alpha} \in \{\tilde{\alpha}\}$, are selected in the set M_f such that

$$\begin{gathered} f(a_{i1}, \dots, a_{il}) = (\alpha_{i1}^A, \dots, \alpha_{il}^A), \text{ if} \\ f(a_{i1}, \dots, a_{il}) \in M_f^{\tilde{\alpha}}. \end{gathered} \tag{57}$$

Out of the functional decision rules of order l, those used most often are *linear* rules. The basis in these rules consists of linear functions $b_{j1}x_1 + \dots + b_{jl}x_l + b_{jl+1}$ ($j = 1, 2, \dots, l$) or the function $a_1x_1 + \dots + a_lx_l + a_{l+1}$.

The partition in linear rules is done by means of constants c_{1j}, c_{2j}, $c_{1j} \le c_{2j}$ ($j = 1, 2, \dots, l$).

III. Decision rules of denumerable order. The basis is a sequence of functions, $f_1^j(x_1), \dots, f_k^j(x_1, \dots, x_k), \dots$.

We select a function $f_{q \cdot l}^j(x_1, \dots, x_{q \cdot l})$. Let $(\{\alpha_{ij}\}_{q \times l}) = N_{q \times l}$ be a collection of information matrices of dimension $q \times l$.

For each $N \in N_{q \times l}$, we pick in $M_{f_{q \cdot l}}$ the subsets $M_{f_{q \cdot l}}(N)$

$$\begin{gathered} r_A(\{a_{ij}\}_{q \times l}) = N, \text{ if} \\ f_{q \cdot l}(a_{11}, \dots, a_{ij}, \dots, a_{q \cdot l}) \in M_{f_{q \cdot l}}(N). \end{gathered} \tag{58}$$

CHAPTER III
THE GENERAL THEORY OF RECOGNITION (CLASSIFICATION) ALGORITHMS

§ 1. The Definition of the Class of Recognition Algorithms [26]

The analysis of recognition models suggests the following definition for the recognition algorithm.

Let there be specified $I = \{I_0(K_1, \dots, K_l)\}$ and for each S in M let there be defined the class of admissible descriptions $I_S = \{I(S)\}$.

Definition 19. The algorithm A is a recognition algorithm if it transforms the learning information $I_0(K_1, \dots, K_l)$ and the descriptions of an arbitrary finite number q of objects, $I(S_1'), \dots, I(S_q')$, into a matrix, $\{\alpha_{ij}^A\}_{q \times l}$ composed of elements $\{1, 0, \Delta\}$.

In the definition, it is assumed that $I(S_u') \in I_{S_u'}$ ($u = 1, 2, \dots, q$) and $I_0 \in I$.

The collection $(I(S_1'), \dots, I(S_q'))$ is further denoted by $I(S_1', \dots, S_q')$. Then

$$A(I_0(K_1, \dots, K_l), I(S_1', \dots, S_q')) = \{\alpha_{ij}^A\}_{q \times l} \tag{59}$$

is the symbolic notation of the recognition algorithm. The learning information $I_0(K_1, \dots, K_l)$ is denoted sometimes by $I_0(l)$ and the collection $I(S_1', \dots, S_q')$ by $I_S(q)$. Then instead of (59), we can write

$$A(I_0(l), I_S(q)) = \{\alpha_{ij}^A\}_{q \times l}. \tag{60}$$

In what follows, the standard interpretation is adopted for the elements α_{ij}^A:

$\alpha_{ij}^A = 1$: the element S_i' belongs to K_j;

$\alpha_{ij}^A = 0$: the element S_i' does not belong to K_j;

$\alpha_{ij}^A = \Delta$: the algorithm A failed to establish whether or not the element S_i' belongs to K_j ($i = 1, 2, \dots, q$; $j = 1, 2, \dots, l$).

In specific models of recognition algorithms, the algorithm A was executed in two stages. At the first stage, the collection $(I_0(l), I_S(q))$ was acted upon by an operator that transformed $(I_0(l), I_S(q))$ into a numerical matrix, $\{a_{ij}\}_{q \times l}$. At the second stage, the decision rule was applied to transform the numerical matrix $\{a_{ij}\}_{q \times l}$ into an information matrix, $\{\alpha_{ij}^A\}_{q \times l}$.

We introduce the subclass of recognition algorithms executed in two stages.

Definition 20. The operator R_A is called a recognition operator if it processes $(I_0(K_1, \dots, K_l), I(S_1', \dots, S_q'))$ into a numerical matrix, $\{a_{ij}\}_{q \times l} = M_{q \times l}$, where $I_0(K_1, \dots, K_l) \in I$, S_i' are arbitrary admissible objects ($i = 1, 2, \dots, q$), q is an arbitrary integer and $I(S_i') \in I_{S_i'}$.

Clearly, $M_{q \times l} = M_{q \times l}(I_0(l), I_S(q))$. The symbolic notation for the recognition operator is

$$R_A(I_0(K_1, \dots, K_l), I(S_1', \dots, S_q')) = \{a_{ij}\}_{q \times l}$$

or in compact form

$$R_A(I_0(l), I_S(q)) = \{a_{ij}\}_{q \times l}. \tag{61}$$

Definition 21. The operator r_A is called a decision rule if it transforms an arbitrary numerical matrix $\{a_{ij}\}_{q \times l} = M_{q \times l}$ into an information matrix $\{\alpha_{ij}^A\}_{q \times l} = I_{q \times l}$; that is, a matrix whose elements are $\{1, 0, \Delta\}$.

The symbolic notation for a decision rule is

$$r_A(\{a_{ij}\})_{q \times l} = \{\alpha_{ij}^A\}_{q \times l}, \quad r_A(M_{q,l}) = I_{q \times l}. \tag{62}$$

We clearly defined the operation of the product of operators R_A and r_A, which is written $R_A \cdot r_A$. It reduces to the successive application of R_A and r_A.

Definition 22. The operator $R_A \cdot r_A = \tilde{A}$ is called a standard recognition algorithm or SRA for short.

In what follows, we are mainly concerned with SRAs. The symbolic notation of the SRA $\tilde{A}$ is the same as for the arbitrary recognition algorithm

$$\tilde{A}(I_0(l), I_S(q)) = \{\alpha_{ij}^{A}\}_{q \times l} = I_{q \times l}. \qquad (63)$$

Clearly, the SRA is a recognition algorithm; that is, $\{\tilde{A}\} \subseteq \{A\}$.

It is easy to show that any recognition algorithm A can be transformed into the standard $\tilde{A}$. Indeed, suppose $A(I_0(l), I_S(q)) = \{\beta_{ij}\}_{q \times l}$. Consider the algorithm $O_N\{\beta_{ij}\}_{q \times l} = \{\alpha_{ij}\}_{q \times l}$, where $\{a_{ij}\}$ is a numerical matrix defined by the following relations

$$\text{if } \beta_{ij} \in \{0, 1\}, \text{ then } a_{ij} = N \cdot \beta_{ij}, N > 1;$$

$$\text{if } \beta_{ij} = \Delta, \text{ then } a_{ij} = 1.$$

Now we apply the decision rule $\tilde{r}_A$ to $\{a_{ij}\}_{q \times l}$:

$$\text{if } a_{ij} \geq N, \text{ then } S_i' \in K_j, \text{ i.e., } \beta_{ij} = 1; \text{ at } a_{ij} \leq 0\text{: } S_t' \,\bar{\in}\, K_j, \text{ i.e., } \beta_{ij} = 0;$$

$$\text{if } 0 < a_{ij} < N, \text{ then } \beta_{ij} = \Delta.$$

The definitions of O_N and $\tilde{r}_A$ imply that $A \cdot O_N \cdot \tilde{r}_A = A$ and the algorithm $(A \cdot O_N)\tilde{r}_A$ is thus a standard recognition algorithm. We proved

Theorem 8. *For each recognition algorithm A there exists a standard recognition algorithm $\tilde{A}$ such that the equality*

$$A(I_0(l), I_S(q)) = \tilde{A}(I_0(l), I_S(q))$$

holds for each pair $(I_0(l), I_S(q))$.

In what follows, we are concerned solely with decision rules $\tilde{r}_A$ that satisfy the additional condition.

Let $S_1', \ldots, S_q'$ be an arbitrary finite collection of admissible objects and $P_j(S_i')$ be a predicate which is rendered as $S_i' \in K_j$ $(j = 1, 2, \ldots, l)$. Recall that the information matrix $\{\alpha_{ij}^{A}\}_{q \times l}$, such that $\alpha_{ij}^{A} \in \{0, 1\}$ is called true for $S_1', \ldots, S_q'$ if $\alpha_{ij}^{A} = P_j(S_i')$ $(i = 1, 2, \ldots, q, j = 1, 2, \ldots, l)$.

Definition 23. The decision rule $\tilde{r}_A$ is well-defined if for any finite collection of admissible objects, $S_1', \ldots, S_q'$, there exists a numerical matrix $\{a_{ij}\}_{q \times l}$ such that $\tilde{r}_A$ transforms $\{a_{ij}\}_{q \times l}$ into a matrix true for $S_1', \ldots, S_q'$.

In what follows, we consider solely well-defined decision rules.

§ 2. The Linear Space of Recognition Operators

In the recognition operator set $\{R_A\}$, the operations of addition and multiplication by a scalar are introduced in a natural way.

Let $R_A^1(I_0(l), I_S(q)) = \{a_{ij}\}_{q \times l}$ and $R_A^2(I_0(l), I_S(q)) = \{b_{ij}\}_{q \times l}$.

Definition 24. The operator $R_A = R_{A_1} + R_{A_2}$ is the sum of the operators R_{A_1} and R_{A_2} if $R_A(I_0(l), I_S(q)) = \{c_{ij}\}_{q \times l} = \{a_{ij} + b_{ij}\}_{q \times l}$.

The recognition operator $R_A = c \cdot R_A^1$, where c is a scalar, is called the product of the scalar c times the operator R_A^1 if

$$R_A(I_0(l), I_S(q)) = \{c \cdot a_{ij}\}_{q \times l}.$$

Theorem 9. *Under the operations of addition and multiplication by a scalar, the set of recognition operators forms a linear vector space. The multiplication by the scalar is commutative.*

Proof. It readily follows from the fact that the set of numerical matrices of the same order forms a commutative group under addition. It readily follows from this fact and from the definition of the recognition operator that the set of recognition operators forms an Abelian group under addition. The axioms related to the multiplication by the scalar and the distributivity of the multiplication by the scalar can be verified with ease. The commutativity of the multiplication by the scalar is likewise obvious; that is, $a \cdot b \cdot R_A = b \cdot a \cdot R_A$. The theorem is proved.

Thus, from the specified series of SRAs, $A_1 = (R_A, r_A), \ldots, A_t = (R_{A_t} r_A)$, we can generate new algorithms, $A = (\sum_{i=1}^{t} a_i R_{A_i} r_A)$, where a_i are constants. In this case, the algorithm A is called linear in $A_1 \ldots A_t$. The collection $A_1 \ldots A_t$ is the basis of A. Let there be specified a collection $\{A\}$ of SRAs with a fixed decision rule r_A.

Definition 25. The linear closure $L\{A\}$ of the set (model} $\{A\}$ is the collection of recognition algorithms linear in all possible finite bases from $\{A\}$, with a fixed decision rule r.

Clearly, each algorithm B in $L\{A\}$ is representable as $A = (\sum_{A_t \in \{A\}} a_i R_{A_t} r)$. The definition permits the construction of a linear closure for any model $\mathfrak{M}(A)$ of recognition algorithms.

If SRA is replaced by recognition operators in Definition 25, the result is the definition of the linear closure $L\{R_A\}$ for the set $\{R_A\}$ of recognition operators.

Finally, let there be specified a set (model) of SRAs, $\mathfrak{M} = \{(R_A r_A)\}$, with different decision rules. We represent the model $\mathfrak{M}$ as $\bigcup_{r_A} \mathfrak{M}(r_A)$. Here, $\mathfrak{M}(r_A)$

is the collection of all the algorithms of the model $\mathfrak{M}$ with the fixed decision rule r_A.

Definition 26. The set $L(\mathfrak{M}) = \bigcup_{r_A \in \{r_A\}} L(\mathfrak{M}(r_A))$ is called the linear closure of the model $\mathfrak{M}$.

Let there be given a finite set of models, $\mathfrak{M}_1, \ldots, \mathfrak{M}_t$.

$$\mathfrak{M}_i = \{(R_A^i r_A^i)\},\quad R_A^i \in \{R_A^i\},\quad r_A^i \in \{r_A^i\},$$
$$i = 1, 2, \ldots, t.$$

Definition 27. The linear closure of the models $\mathfrak{M}_1, \ldots, \mathfrak{M}_t$ is the set $L(\bigcup_{i=1}^t L(\mathfrak{M}_i)) = L(\mathfrak{M}_1, \ldots, \mathfrak{M}_t)$.

Any algorithm A in $L(\mathfrak{M}_1, \ldots, \mathfrak{M}_t)$ can be constructed as follows. From the set $\bigcup_{i=1}^t \mathfrak{M}_i$, we select the basis $A'_u, \ldots, A^p_u$ (i.e., an arbitrary collection of algorithms with the same decision rule r), $A = (R_A r_A)$, $r_A = r$, and $R_A = \sum_{i=1}^{p} c_i R_{A^i_u}$.

The proof of the last assertion stems readily from the definitions of the linear closure of a model and of the linear closure of the collection of models. It suffices to demonstrate that

$$L\left(\bigcup_{i=1}^{t} \mathfrak{M}_i\right) = L\left(\bigcup_{i=1}^{t} L(\mathfrak{M}_i)\right). \qquad (64)$$

Generally, recognition operators R_A can be applied to parts of I_0.We, henceforth, consider linear closures of models, where different terms are the same operators applied to different parts of $I(K_1, \ldots, K_l)$. In these cases, we explicitly indicate the part of I_0 where the operator R_A is active.

§ 2. Extremum Algorithms in Linear Closures of Models. Complete Models

Consider the general form of the quality functional

$$f_q(\rho(\tilde{\alpha}(S'_1), \tilde{\alpha}^A(S'_1)), \ldots, \rho(\tilde{\alpha}(S'_q), \tilde{\alpha}^A(S'_q))),$$

where $\tilde{\alpha}(S'_i)$ is the true information vector of S'_i, and $\tilde{\alpha}^A(S'_i)$is the information vector of S'_i formed by the algorithm A.

Recall that $f_q(0, \ldots, 0)$ is the absolute maximum over the entire domain of f_q; that is, f_q achieves its absolute maximum if $\tilde{\alpha}(S'_1) = \tilde{\alpha}^A(S'_i)$ $(i = 1, 2, \ldots, l)$.

Let there be specified a learning information I_0 and an arbitrary finite control sample $S'_1, \ldots, S'_q$ in M, such that I_0 belongs to the set of admissible learning informations, $I_0 \in \{I_0\}$.

Definition 28. The model $\mathfrak{M}$ is well-defined if for any I_0, q, $S'_1, \ldots, S'_q$ in $\mathfrak{M}$ there exists the algorithm A such that the quality functional f_q achieves its absolute maximum on A.

Let the model $\mathfrak{M}$ be composed of $A = R_A r_A$. Let $\mathfrak{M}(R_A)$ denote the collection of recognition operators of the model $\mathfrak{M}$.

Definition 29. The model $\mathfrak{M}$ is called complete if for any I_0, q, $S'_1, \ldots, S'_q$ the set of matrices of dimension $q \times l$: $R_A\{I_0(K_1, \ldots, K_l), I(S'_1, \ldots, S'_q)\} = M(R_A)$, $R_A \in \mathfrak{M}\{R_A\}$, contains the basis in the space of numerical matrices of dimension $q \times l$. The collection $\mathfrak{M}\{R_A\}$ of operators in the complete model is called the complete collection.

Theorem 10. *If the model $\mathfrak{M}$ is complete and the algorithms in $\mathfrak{M}$ have only well-defined decision rules, the model $\mathfrak{M}$ has a well-defined linear closure $L(\mathfrak{M})$.*

Proof. Let there be given an arbitrary finite control sample $S'_1, \ldots, S'_q$, initial information I_0 and a set of descriptions $\{I(S'_1), \ldots, I(S'_q)\} = I_S(q)$. The basis in $M_{q\times l}$ consists of $q \cdot l$ matrices $M^1, \ldots, M^{q\cdot l}$.

Consider the decision rule r. Since we are concerned solely with well-defined decision rules, there exists a numerical matrix $M_q(S) = \{a_{ij}\}_{q\times l}$ such that $r(M_q(S)) = \{P_j(S'_i)\}_{q\times l}$; that is, the information matrix $r(M_q(S))$ is a matrix of true information vectors for objects $S'_1, \ldots, S'_q$ (recall that the predicate $P_j(S'_i)$ is rendered as "object S'_1 is a member of K_j").

Now it suffices to construct the matrix $M_q(S)$ using $(I_0(l), I_S(q))$. Since the set of matrices, $M\{R_A\}$, contains the basis and it is finite (consists of $q \cdot l$ elements), it is possible to select $q \cdot l$ operators $R_{A_1}, \ldots, R_{A_{q\,l}}$ that, when applied to $(I_0(l), I_S(q))$, generate the basis matrices $M^1, \ldots, M^{q\cdot l}$.

Then (by the definition of the basis in linear space) there exist constants $a_1, \ldots, a_{q\cdot l}$ such that

$$M_q(S) = \sum_{i=1}^{q\cdot l} a_i \cdot M^i.$$

The recognition operator $R_A = \sum_{i=1}^{q\ l} a_t \cdot R_{A_t}$ is part of the closure of the system of operators, which contains the operators $R_{A_1}, \ldots, R_{A_{q\,l}}$.

Consider the algorithm $(R_A r) = A$, where $R_A = \sum_{i=1}^{q\ l} a_i \cdot R_{A_i}$. Clearly, $A(I_0(l), I_S(q)) = \{\alpha_{ij}^A\}_{q\times l} = \left\{P_j(S'_i)\right\}_{q\times l}$.

Then $\rho(\tilde{\alpha}_A(S'_i), \tilde{\alpha}(S'_i)) = \rho(\tilde{x}, \tilde{x}) = 0$; i.e., the function ρ satisfies the axiom of distance: $\rho(a, a) = 0$. Therefore, the quality functional is

$$f_q(\rho(\tilde{\alpha}(S'_1), \alpha_A(S'_1)), \dots, \rho(\tilde{\alpha}_q(S'_q), \tilde{\alpha}_A(S'_q)))$$
$$= f(0, 0, \dots, 0, 0).$$

By definition, the functional f_q reaches its absolute maximum on the set (0, 0, …, 0, 0). So, the linear closure $L(\mathfrak{M})$ of the model $\mathfrak{M}$ is correct. The theorem is proved.

Corollary. *If the model $\mathfrak{M}$ is complete (and hence well-defined), then the model $\mathfrak{M}'$ with one fixed decision rule r and the collection of recognition operators $\mathfrak{M}(R_A)$ is complete and thus well-defined.*

Consider one application of Theorem 10.

We review the case where the information $I_0(K_1, \dots, K_l) = I_0(l)$ and the descriptions $\{I(S'_1), \dots, I(S'_q)\} = I_S(q)$ of the objects to be recognized are specified in standard form.

Consider R algorithms with piecewise-linear separating surfaces and parameters $\gamma_1, \dots, \gamma_m$ (the model $\mathfrak{M}^1(\tilde{\gamma}, l)$ where $R_1 = \dots = R_l$. The descriptions of objects $S_1, \dots, S_m$ included in I_0 and the descriptions $I(S'_1), \dots, I(S'_q)$ are points in n-dimensional Euclidean space. The separation is done by one surface R for all classes. The algorithm is defined by giving the surface R and the parameters $\gamma_1, \dots, \gamma_m$, which are the weights of objects $S_1, \dots, S_m$.

The following additional constraints are imposed further on the standard learning information $I_0 = (I(S_1), \tilde{\alpha}(S_1), \dots, I(S_m), \tilde{\alpha}(S_m))$:

1. The vectors $\tilde{\alpha}(S_i)$ do not contain symbols Δ and are true for S_i; that is, if $\tilde{\alpha}(S_i) = (\alpha_{i1} \dots \alpha_{il})$, then α_{ij} is the value of the predicate $P_j(S_i)$, which is rendered as "S_i is a member of K_j."

2. Represent the set $\{S_1, \dots, S_m\} = W^1_j \cup W^0_j$, the subsets W^1_j and W^0_j being disjoint and composed of the elements S_i for which $\alpha_{ij} = 1$ and $\alpha_{ij} = 0$, respectively ($i = 1, 2, \dots, m; j = 1, 2, \dots, l$).

Condition: $W^1_u \not\subseteq W^1_v$ if $v \neq u$ $(u = 1, 2, \dots, l)$. Condition 2 means that for each pair of classes, K_u and K_v, the learning sample has at least one object that is a member of K_u and is not a member of K_v.

In addition, we assume that $S'_t \mathrel{\overline{\in}} \{S_1, \dots, S_m\}$ $(t = 1, 2, \dots, q)$.

Theorem 11.[1] *The class of recognition operators R_A defined by a piecewise-linear separating surface R and parameters $\{\gamma_1, \dots, \gamma_m\} = \{\gamma(S_1), \dots, \gamma(S_m)\}$, $(\gamma_i \geq c > 0)$ is complete.*

Proof. Fix an arbitrary control sample, $S'_1, \dots, S'_q$. Demonstrate that the set of matrices $M_{q \times l}$ generated by the operators R_A from the standard initial information $(I_0(l), I_S(q))$ contains the basis in the linear space of numerical matrices of dimension $q \times l$.

The proof consists in direct construction of the operators $R^{i,l}_A$ in $L(R_A)$ (the linear closure of the class R_A) which transforms $(I_0(l), I_S(q))$ into a numerical matrix, $\{b_{uv}\}^{i,j}_{q \times l}$, such that $b_{ij} = 1$ and the remaining $b_{uv} = 0$ $(i = 1, 2, \dots, q, j = 1, 2, \dots, l)$. It is obvious that the matrices $\{b_{uv}\}^{i,j}_{q \times l}$ form the basis in the set $M_{q \times l}$ of numerical matrices of dimension $q \times l$.

I. Construct an arbitrary piecewise-linear surface R_1 subject to the following constraints:

1°. $R_1(S'_1) > 0$;

2°. $R(I(S_u)) > 0$ for all S_u in W^1_j;

3°. $R_1(I(S'_v)) \leq 0$, $v = 1, 2, \dots, i - 1, i + 1, \dots, q$;

4°. $R_1(I(S_u)) \leq 0$ for all S_u in W^0_j.

Suppose $\gamma_1 = \dots = \gamma_m = N^k$ where $N > 1$ and k is a natural number. Evaluate the matrix $\{r^1_{uv}\}_{q \times l}$ into which the operator R_1 transforms the initial information. Recall that for the objects S'_u such that

$$R(S'_u) > 0: \; r^1_{uv} = \frac{Q^{11}_v + Q^{00}_v}{Q^{01}_v + Q^{10}_v + 1};$$

and for the objects S'_u such that

$$R(S'_u) \leq 0: \; r^1_{uv} = \frac{Q^{01}_v + Q^{10}_v}{Q^{00}_v + Q^{11}_v + 1},$$

where $Q^{\alpha_1\alpha_2}_v$ is the sum of the weights γ_t of the elements S_t belonging to $S^{\alpha_1\alpha_2}_v$, such that $\alpha_1, \alpha_2 \in \{0, 1\}$; α_2 is the value of the vth coordinate in the information vector $\tilde{\alpha}(S_t)$; α_1 is equal to 1 if $S^{1\alpha_2}_v$ includes the objects S_t that satisfy the inequality $R(S_t) > 0$ and $\alpha_1 = 0$ otherwise.

[1] After the manuscript had been sent to press, the author received more general results [81].

By construction of R_1 (constraint 1°): $R_1(I(S_i'))>0$. For it, the set S_v^{11} consists of the objects that belong to W_j^1 (constraint 2° for the construction of R_1). and the set S_v^{00} consists of the objects that belong to W_j^0 (constraint 4° in the construction of R_1). Since $\{S_1, \dots, S_m\} = W_j^0 \cup W_j^1$, the sets S_v^{01} and S_v^{10} are empty for S_i'. Considering that $\gamma_1 = \dots = \gamma_m = N^k$, we get $\overset{1}{r}_{ij} = m \cdot N^k$.

We evaluate $\overset{1}{r}_{iv}$ for $v = 1, 2, \dots, j-1, j+1, \dots, l$. Let us show that in this case $\overset{1}{r}_{uv} \le m-1$. Indeed, under additional constraint 2 on the initial information, $W_j^1 \not\supseteq W_v^1$. Therefore, there exists at least one element S_t in W_j^1 which does not belong to W_v^1, or $S_t \in W_v^1$ and $R(S_t) \le 0$.

Consider the sets $S_v^{\alpha_1\alpha_2}$ for S_i'. We have $R(S_i')>0$ and $R(S_t)>0$, ($S_t \in W_j^1$) and the vth coordinate of the information vector $\tilde{\alpha}(S_t)$ is 0. Therefore, for S_i' the element S_t belongs to the set S_v^{10}.

Consequently, there is no term $\gamma_t = \gamma(S_t) = N^k$ in the numerator of the expression for r_{iv}, whereas it appears in the denominator. We have

$$\overset{1}{r}_{iv} \le \frac{m \cdot N^k - N^k}{N^k + 1} < m-1. \qquad (65)$$

It is easy to show that for $u = 1, 2, \dots, i-1, i+1, \dots, l$ the quantities $\overset{1}{v}_{uj}$ are 0. Indeed, by the construction of $R_1 : R_1(S_u) < 0$ and for S_u $S_j^{00} = W_j^0$, $S_j'' = W_j^1$, and the sets S_j^{01} and S_j^{10} are empty.

II. Draw again an arbitrary piecewise-linear surface R_2 that satisfies constraints 1° through 4°. Here, constraints 2° through 4° are the same as in I and constraint 1° is altered: $R_2(S_i')<0$. Again, all $\gamma_i = N^k$.

Similar to I, we analyze the matrix $\{\overset{2}{r}_{uv}\}_{q\times l}$ into which the operator defined by the specified set R_2, γ_1, ..., γ_m transforms $(I_0(l), I_S(q))$.

Note that for the objects $S_1', \dots, S_{i-1}', S_{i+1}', \dots, S_q'$ nothing changed as compared to I. Therefore,

$$\overset{1}{r}_{uv} = \overset{2}{r}_{uv} \ \text{ for } \ u \neq i, \ \ v = 1, 2, \dots, l. \qquad (66)$$

It is easy to show that

$$\overset{2}{r}_{ij} = 0. \qquad (67)$$

Indeed, the sets Q_{01} and Q_{10} are empty for S_i' relative to R_2 and K_j.

The inequality

$$r_{iv} \le m-1, \ \ v = 1, 2, \dots, j-1, j+1, \dots, l \qquad (68)$$

can be proved in the same way.

The following assertion follows from (66) through (68) and from similar results in I:

Let $R_A^{1,k}$ and $R_A^{2,k}$ be the operators defined by the sets $(R_1, \gamma_1 = \dots \gamma_m = N^k)$ and $(R_2, \gamma_1 = \dots = \gamma_m = N^k)$, respectively. Then the operator $R_A^k = \frac{1}{m \cdot N^k}(R_A^{1,k} - R_A^{2,k})$ appears in the linear closure $L(R_A)$ of the class of operators defined by the piecewise-linear separating surface R and by the set of parameters $\gamma_1 \dots \gamma_m$ and

$$M_{q\times l}^k = R_A^k(I_0(l), I_S(q))$$

$$= \begin{pmatrix} 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ a_{i1} & a_{i2} & \dots & a_{ij-1} & 1 & a_{ij+1} & \dots & a_{il-1} & a_{il} \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Here,

$$|a_{iu}| \le \frac{2(m-1)}{m \cdot N^k}, \ u = 1, 2, \dots, j-1, j+1, \dots, l.$$

Clearly, the sequence $M_{q\times l}^k$ converges uniformly to the matrix $\{b_{uv}\}_{q\times l}^{i,j}$, where $b_{ij} = 1$ and the remaining elements are 0.

The collection of the numerical matrices $M_{q\times l}$ generated by operators in the linear closure $L(R_A)$ form a finite-dimensional linear vector space. It contains its limit points with a bounded norm. So, $\{b_{uv}\}_{q\times l}^{i,j} \in M_{q\times l}$ ($i = 1, 2, \dots, q, j = 1, 2, \dots, l$). The theorem is proved.

Corollary. *The class of recognition algorithms* (R_A, r_A) *defined by the collection of piecewise-linear separating surfaces* $R_1, \dots, R_l$ *(the elements of each class are separated by a surface of their own) and by the parameter set* $\gamma_1, \dots, \gamma_m$ *(the model* $\mathfrak{M}_k^L(\tilde{\gamma}, l)$*) is complete.*

The proof follows from the fact that the subclass

$$R_1 = \dots = R_l = R$$

is complete.

§ 3. Extremum Algorithms in Linear Closures of Models. Weak Completeness of Models[2]

The definitions of a well-defined and complete model introduced in the preceding section make it possible to formulate two constraints under which the linear closure of a model contains an algorithm exactly accurate on each control sample of fixed length.

As a result, weak constraints are imposed on the class of decision rules and rather strong constraints on the class of recognition operators of the model $\mathfrak{M}$. Let us introduce another condition for the model to be well-defined, in which more rigorous constraints are imposed on the class of decision rules and weaker constraints on the class of recognition operators of the model $\mathfrak{M}$.

Definition 30. The model $\mathfrak{M}$ (the collection of recognition operators, $\mathfrak{M}(R_A)$) is complete in the weak sense if for any I_0, q, $S'_1, \dots, S'_q$ the set of matrices $M(R_A) = \{R_A(I_0(l), I_S(q))\}$ of dimension $q \times l$ includes the subset of matrices $\{M^{i,j}_{q \times l}\}$ ($i = 1, 2, \dots, q$, $j = 1, 2, \dots, l$) such that

1°. The element that is maximum in modulus in $M^{i,j}_{q \times l} = \{c_{ij}\}_{q \times l}$ is c_{ij};

2°. $c_{ij} > c_{ut}$ for $(u, t) \neq (i, j)$.

In what follows, we assume that the elements $S_1, \dots, S_m$ whose descriptions constitute I_0 satisfy the condition

$$\varnothing \neq W^1_j = K_j \cap \{S_1 \dots S_m\} \neq W^1_u = K_u \cap \{S_1 \dots S_m\},$$
$$j = 1, 2, \dots, l,\ u \neq j.$$

Let the learning information I_0 include descriptions $I(S_u) = (a_{u1}, \dots, a_{un})$ and the control sample include descriptions $I(S'_i) = (b_{i1}, \dots, b_{in})$ ($u = 1, 2, \dots, m$, $i = 1, 2, \dots, q$).

Definition 31. The set $I = \{I(S_1), \dots, I(S_m), I(S'_i), \dots, I(S'_q)\}$ is called correct if for each pair (S'_v, S'_w) and for each K_j one can find a $S_u \in K_j$ and a feature $r(u = u(v, w)$, $r = r(u, w))$ such that $\rho_r(a_{ur}, b_{vr}) < \rho_r(a_{ur}, b_{wr})$ ($1 \leq v$, $w \leq n$, $v \neq w$, $1 \leq r \leq n$). The standard information that contains the correct set I is called correct.

Consider the estimates calculation model $\mathfrak{M}(\vec{\varepsilon}, \vec{p}, \vec{\gamma})$ defined as follows.

1. The system of reference subsets consists of subsets of length 1 of the set $\{1, 2, \dots, n\}$; that is, in the notation of Section I, Chapter II the parameter k is 1.

2. The proximity function $B_0^{\vec{\varepsilon}}$ is a threshold one where $\varepsilon = 0$ and it depends on the parameters $\varepsilon_1, \dots, \varepsilon_n$. The parts of $\tilde{\omega} S$ in the subclass in question consist each of one element. Suppose $\tilde{\omega} S_u = a_{ui}$, $\tilde{\omega} S'_t = b_{ti}$; $B_0^{\vec{\varepsilon}}(\tilde{\omega} S_u, \tilde{\omega} S'_t) = 1 = B(a_{ui}, b_{ti})$ if and only if $\rho_i(a_{ui}, b_{ui}) \leq \varepsilon_i$, $1 \leq i \leq n$.

3. $\Gamma_j(S'_t) = \tilde{Q} \cdot \sum\limits_{S_u \in W^1_j} \gamma(S_u) \cdot \sum\limits_{i=1}^{n} p_\iota\, B(a_{ui}, b_{ti})$.

The estimates $\Gamma_j(S'_t)$ ($t = 1, 2, \dots, q$, $j = 1, 2, \dots, l$) are functions of the parameters $\vec{\varepsilon} = (\varepsilon_1 \dots \varepsilon_n)$, $\vec{p} = (p_1 \dots p_n)$, and $\vec{\gamma} = (\gamma_1 \dots \gamma_m)$. The set of the operators that evaluate the estimates $\Gamma_j(S'_t)$ from the initial information is denoted by $R_A(\vec{\varepsilon}, \vec{p}, \vec{\gamma})$.

Let $\{I_0, I(S'_1), \dots, I(S'_q)\} = \mathfrak{M}\{I_0, q\}$ be the collection of sets for which $\{I(S_1), \dots, I(S_m), I(S'_1), \dots, I(S'_q)\}$ is correct.

Theorem 12. *The set $R_A(\vec{\varepsilon}, \vec{p}, \vec{\gamma})$ is weakly complete on $\mathfrak{M}(I_0, q)$ ($\tilde{Q} = 1$).*

Proof. It is required to demonstrate that for all pairs (i, j) ($i = 1, 2, \dots, q$; $j = 1, 2, \dots, l$) the linear closure $L(R_A(\vec{\varepsilon}, \vec{p}, \vec{\gamma}))$ contains an operator $R_A^{i,j}$ which transforms the *initial information in* $\mathfrak{M}(I_0, q)$ into a matrix $\{\Gamma_{ut}\}_{q \times l}$ where the element Γ_{ij} is the only maximal element. It takes several steps to construct $R_A^{i,j}$.

1. Fix an object S'_ι and a class K_j. By Definition 31, for any S'_w it is possible to find a feature r and an element $S_u \in W'_j$ such that $\rho_r(a_{ur}, b_{ir}) < \rho_r(a_{ur}, b_{wr})$. Introduce the operator R_{ij}. In defining it, all p_i, except p_r, are set equal to 0 and $p_r = P > 0$; all parameters $\vec{\gamma}$, except γ_u, are set equal to 0 and $\gamma_u = 1$.

The parameter ε_r is chosen so as to satisfy the inequality

$$\rho_r(a_{ur}, b_{ir}) < \varepsilon_r < \rho(a_{ur}, b_{wr}). \tag{69}$$

The remaining ε_i are selected arbitrarily.

Equation (69) implies that the operation of $R^w_{\iota j}$ on $(I_0(l), I_S(q))$ yields the matrix $\{\Gamma^w_{ij}\}$ in which

$$\Gamma_{\iota j} = P,\ \Gamma_{wj} = 0,\ \Gamma_{vs} \leq P,\ (v, s) \neq (w, j). \tag{70}$$

Clearly, the operator $\sum_{w \neq \iota} R^w_{ij} = R_1(S'_1)$ occurs in the linear closure $L(R_A(\vec{\varepsilon}, \vec{p}, \vec{\gamma}))$. It follows from the definition of $R_1(S'_1)$ and (70) that in the matrix

$$R_1\{I_0(l), I_S(q)\} = \{\Gamma^1_{uv}\}_{q \times l}$$

[2] See Note 1.

1°. $\Gamma_{ij}^1 = (q-1)\cdot P$;

2°. $\Gamma_{uj}^1 \le (q-2)\cdot P,\ u \ne i$;

3°. $\Gamma_{vt}^1 \le (q-1)\cdot P,\ v = 1, 2, \dots, q, t = 1, 2, \dots, j-1, j+1, \dots, l$.

2. Fix the class K_j. Since $W_j^1 \not\subseteq W_r^1$, it is possible in the set $S_1, \dots, S_m$ to select the elements S_{i_r} $(r = 1, 2, \dots, j-1, j+1, \dots, l)$ that belong to K_j and do not belong to K_r, respectively. Construct operators R_r^j in which the parameters are defined as follows: $\gamma(S_{i_r}) = \gamma_{i_r} = 1$ and the remaining $\gamma_i = 0$ and $p_1 = \dots = p_n = P$ (the quantity P here is the same as in constructing the operators R_{ij}^w); if $I(S_i) = (a_{i1} \dots a_{in})$ and $I(S_u') = (b_{u1} \dots b_{un})$, then $\varepsilon_t > \rho_t(a_{it}, b_{ut})$.

The operator R_r^j transforms the initial information into a numerical matrix, $\{\Gamma_{vw}(r)\}$ where

4°. $\Gamma_{ir}(r) = 0$;

5°. $\Gamma_{ij}(r) = n \cdot P$;

6°. The remaining elements $\Gamma_{vw}(r) \le n \cdot P$.

The operator $R_2^j = \sum\limits_{r \ne j} R_r^j$ occurs, obviously, in $L\{R_A(\vec{\varepsilon}, \vec{p}, \vec{\gamma})\}$. As follows from its definition and the properties in 4° through 6°, in the matrix $R_2^j(I_0(l), I_S(q)) = \{\Gamma_{uv}^2\}_{q \times l}$.

7°. $\Gamma_{ij}^2 = (l-1)\cdot n \cdot P$;

8°. $\Gamma_{wj}^2 \le \Gamma_{ij}^2, w \ne i$;

9°. $\Gamma_{wv}^2 \le (l-2)\cdot n \cdot P < (l-1)\cdot n \cdot P = \Gamma_{ij}^2$.

3. The operator $R_{ij} = R_1(S_i') + R_2^j$ transforms the initial information into the matrix $\{\Gamma_{vw}\}_{q \times l}$, such that

10°. $\Gamma_{ij} = P((l-1)\cdot n + q - 1)$;

11°. $\Gamma_{ij} - \Gamma_{rt} \ge P$, $(rt) \ne (i, j)$.

We set $R_A^{i,j} = R_{ij}$. The theorem is proved.

Corollary. *In* $L\{R_A(\vec{\varepsilon}, \vec{p}, \vec{\gamma})\}$ *for each pair* (i, j) $(i = 1, 2, \dots, q; j = 1, 2, \dots, l)$ *and each correct standard learning information there exists an operator* $R_N(S_t', K_j)$, *where N is an arbitrary integer, that transforms the correct standard sample into a numerical matrix,* $\{\Gamma_{uv}\}_{q \times l}$, *such that*

$$\Gamma_{ij} - \Gamma_{uv} \ge N, \quad (uv) \ne (ij).$$

The proof follows from 11° in the proof of the theorem if we set $P = N$.

Generally speaking, the operator model $\mathfrak{M}(R_A)$, complete in the weak sense, is not complete. But the addition of one more operator can make it complete. Consider an operator R in the space of numerical matrices

$$R(\{a_{ij}\}_{q \times l}) = \{b_{ij}\}_{q \times l},$$

where $b_{uv} = 1$ if $a_{uv} = \max a_{ij}$ $(i = 1, 2, \dots, q, j = 1, 2, \dots, l)$ and $b_{uv} = 0$ in the other cases.

Definition 32. The model $\mathfrak{M}(R_A \cdot R)$ is called an elementary extension of the recognition operator model $\mathfrak{M}\{R_A\}$.

Theorem 13. *The elementary extension* $\mathfrak{M}(R_A \cdot R)$ *of the weakly complete model* $\mathfrak{M}(R_A)$ *is a complete model.*

Proof. The application of operators in $\mathfrak{M}\{R_A\}$ to I_0 and $I(S_1', \dots, S_q')$ in the weakly complete model $\mathfrak{M}\{R_A\}$ can generate a number of $q \cdot l$ matrices $\{\Gamma_{vw}^u\}_{q \times l}$ $(u = 1, 2, \dots, q \cdot l)$ such that the only maximal element of $\{\Gamma_{vw}^u\}$ is Γ_{ij}. The application of the operator R to these matrices produces matrices that form the basis in the space of numerical matrices $\{a_{uv}\}_{q \times l}$. The theorem is proved.

Let $\mathfrak{M}\{A\}$ be an SRA model, $\mathfrak{M} = \{R_A r_A\}$, with well-defined decision rules r_A, such that $\mathfrak{M}\{R_A\}$ is weakly complete. Let also R be the operator that locates the maximum.

From the foregoing, it is easy to prove

Theorem 14. *The model* $\mathfrak{M}\{A\} = \{R_A \cdot R \cdot r_A\}$ *is complete and, hence, well-defined.*

The model $\tilde{\mathfrak{M}}\{A\}$ is called an elementary extension of the model $\mathfrak{M}\{A\}$.

In many cases, the completeness of a model is technically more difficult to prove than its weak completeness. On the other hand, an elementary addition to the set of recognition operators does not practically complicate the model. The new model is complete on the same set $\{I_0\}$ and, hence, well-defined.

For sufficiently broad classes of decision rules, the models in which the collection of recognition operators is weakly complete have the property of being well-defined.

Let the set $\mathfrak{M}(R_A)$ posses on $\{I_0\}$ the property of weak completeness and also an additional property. For any positive Q and N, $N < Q$, there exist operators $R_{ij}^{Q,N}$ $(i = 1, 2, \dots, q, j = 1, 2, \dots, l)$ such that $R_{ij}^{Q,N}(I_0) = \{\Gamma_{uv}\}_{q \times l}$, $\max \Gamma_{uv} = \Gamma_{ij} = Q$ and $\Gamma_{ij} - \Gamma_{uv} \ge N$, $(uv) \ne (i, j)$.

Weakly complete models possessing the additional property are called regular models.

Theorem 15. *The model* $\mathfrak{M} = \{R(\vec{\varepsilon}, \vec{p}, \vec{\gamma})\}$ *is a regular model.*

Proof. 1. As follows from the proof of Theorem 12 and its corollary, there exists, in $L(\mathfrak{M})$, an operator R_{ij} (see 10° in Theorem 12) which is an operator $R_{ij}^{Q\,N}$ with an arbitrary Q' and $N' = \dfrac{Q'}{(l-1)\cdot n+q-1}$. Then it is easy to indicate y such that $y \cdot R_{ij}^{Q\,N'} = R_{ij}^{Q'\backslash N'\backslash}$, where $N'y = N$ and N is an arbitrary positive number smaller than $Q' \cdot y$.

2. In the model $L\{R(\vec{\varepsilon}, \vec{p}, \vec{\gamma})\}$ for any constant c, there exists an operator R_c that transforms I_0 into a matrix, $\{\Gamma_{ij}\}_{q\times l}$, where all $\Gamma_{ij} = c$.

Clearly, it suffices to prove the assertion for $c = 1$. If $I(S_u) = (a_{u1} \dots a_{un})$ $(u = 1, 2, \dots, m)$ and $I(S_t') = (b_{t1} \dots b_{tn})$, then $\varepsilon_i > \max\rho_i(a_{ui}, b_{ti})$ $(i = 1, 2, \dots, n, t = 1, 2, \dots, q)$.

With the parameters $\varepsilon_1, \dots, \varepsilon_n$, thus chosen, $\tilde{Q} = \dfrac{1}{l\mu(W_j^1)}$ and

$$\Gamma_j(S_i') = \frac{1}{l}\cdot\frac{1}{\mu(W_j^1)}\sum_{S_i \in W_j^1}\gamma_i\sum_{i=1}^{n}p_i.$$

We set $\dfrac{1}{l}\cdot\dfrac{1}{\mu(W_j^1)} = N$. Then for $p_i = \dfrac{1}{n}$, $i = 1, 2, \dots, n$, and $\gamma_i = \dfrac{1}{N|W_j^1|}$ we have $\Gamma_j(S_i') = 1$ for $i = 1, 2, \dots, q$ and $j = 1, 2, \dots, l$.

3. Clearly, $R_{ij}^{Q\ \backslash, N} + R_{Q-Q'\cdot\backslash} = R_{ij}^{Q,N}$.

The theorem is proved.

To begin with, consider the simplest functional decision rules of order 1 with constants $0 < c_{1j} < c_{2j}$ $(j = 1, 2, \dots, l)$: $S_l' \in K_j$ if $a_{ij} > c_{2j}$ and $S_i' \mathrel{\overline{\in}} K_j$ if $a_{ij} < c_{1j}$ for $c_{1j} \le a_{ij} \le c_{2j}$, the decision rule produces $\alpha_{ij}^{A} = \Delta$.

Theorem 16. *The model $\mathfrak{M}\{R_A \cdot r_A\}$ with a regular set $\mathfrak{M}\{R_A\}$ and decision rules of order* 1 *with constants c_{1j} and c_{2j} $(j = 1, 2, \dots, l)$ is well-defined.*

Proof. Consider an arbitrary finite control sample of admissible objects $S_1', \dots, S_q'$ and their true information vectors $\tilde{\alpha}(S_1') = \tilde{\alpha}_1 = (\alpha_{11} \dots \alpha_{1l}), \dots, \tilde{\alpha}(S_q') = \tilde{\alpha}_q = (\alpha_{q1} \dots \alpha_{ql})$. Partition the objects numbered (i, j) $(i = 1, 2, \dots, q; j = 1, 2, \dots, l)$ into subsets M_0, such that: $(i, j) \in M_0$ if $\alpha_{ij} = 0$ and M_1: $(i, j) \in M_1$ if $\alpha_{ij} = 1$.

Suppose $\max\limits_j c_{2j} = b_2$, $\min\limits_j c_{1j} = b_1$ and $b_{12} = b_2 - b_1$. For each pair of objects numbered (i, j), from M_1 we select in $L\{\mathfrak{M}(R_A)\}$ an operator $R_{ij}^{Q,N}$ (see the definition of the regular model) such that

$$Q = b_2 + \varepsilon > b_2 \ge c_{2j}, \quad Q > N > Q - \frac{b_2 - b_{12}}{\mu(M_1)}, \quad (71)$$

where $\mu(M_1)$ is the cardinality of the set M_1.

The operator $\sum_{(i,j)\in M_1} R_{ij}^{Q,N}$ on I_0 produces a matrix $\{\Gamma_{ij}\}_{q\times l}$, where

1°. If $(i, j) \in M_1$, then $\Gamma_{ij} > b_2 + \varepsilon > c_{2j}$

2°. If $(i, j) \in M_0$, then $\Gamma_{ij} < c_{1j}$.

The first assertion trivially follows from (71). Let us prove 2°. In $R_{ij}^{QN}(I_0) = \{\Gamma_{ij}^{Q,N}\}$ the quantity $\Gamma_{uv}^{QN} \le Q - N < \dfrac{b_2 - b_{12}}{\mu(M_1)}$. Then $\Gamma_{uv} \le \mu(M_1)(Q - N) < b_2 - b_{12} = \min\limits_j c_{1j}$.

By applying the decision rule to $\{\Gamma_{ij}\}_{q\times l}$, we get true information vectors $\tilde{\alpha}(S_i')$ $(i = 1, 2, \dots, q)$.

The above construction is feasible because $\min\limits_j c_{1j} > 0$ and so $N < Q$. The theorem is proved.

Corollary 1. *If the model $\mathfrak{M} = \{R_A \cdot r_A\}$ with the regular $\mathfrak{M}\{R_A\}$ contains for each I_0 an $R_A^{I_0}$ such that $R_A^{I_0}(I_0) = \{\Gamma_{ij}\}_{q\times l}$ and $\Gamma_{ij} \equiv$* const, *then the model $\mathfrak{M}\{R_A \cdot r_A\}$is well-defined under decision rules with arbitrary c_{1j} and c_{2j}, $c_{1j} < c_{2j}$.*

Proof. It suffices to consider the case where $\max\limits_j c_{2j} \ge 0$ and $\min\limits_j c_{1j} \le 0$. To begin with, consider the new decision rules where $\tilde{c}_{2j} = c_{2j} + \max\limits_j|c_{1j}| + \delta$ and $\tilde{c}_{1j} = c_{1j} + \max\limits_j|c_{1j}| + \delta''$, $\delta'' > 0$. These new decision rules obey Theorem 16. Having considered then the operator $\sum_{(i,j)\in M_1} R_{ij}^{Q,N} - R\max\limits_j|c_{1j}| + \delta''$, we prove Corollary 1.

Corollary 2. *The model $\mathfrak{M}(\vec{\varepsilon}, \vec{p}, \vec{\gamma})$ where $\mathfrak{M}(R_A) = \{R(\vec{\varepsilon}, \vec{p}, \vec{\gamma})\}$ and the decision rules are of order* 1 *with arbitrary constants c_{1j} and c_{2j} such that $c_{1j} < c_{2j}$ $(j = 1, 2, \dots, l)$ is well-defined.*

Indeed, in proving the theorem we constructed the operator $R(\vec{\varepsilon}, \vec{p}, \vec{\gamma})$ which for an arbitrary correct I_0 constructs the matrix $\{\Gamma_{ij}\}_{q\times l}$, where $\Gamma_{ij} \equiv 1$.

If we consider an arbitrary decision rule of order 1 with the functions $f_1(x), \dots, f_l(x)$ monotonically increasing, defined on the entire number axis, and unbounded from either above or below, then clearly, the theorem remains valid as do its corollaries.

For threshold decision rules of order f with constants c_{1j} and c_{2j}, the theorem on the well-defined property of models remains valid under additional constraints on the basis of the decision rule. Let $f_l^j(x_1, \ldots, x_l)$ satisfy the conditions:

1°. $f_j^j(x_1, \ldots, x_l) \equiv \psi_l^j(x_j - x_1, \ldots, x_j - x_{j-1}, x_j, x_{j+1} - x_j, \ldots, x_j - x_l)$;

2°. ψ_l^j is monotonically increasing in each of the arguments $x_j - x_1, \ldots, x_j, \ldots, x_j - x_l$;

3°. $\lim\limits_{x_j \to \infty} \psi_i^j(c_1, \ldots, c_{j-1}, x_j, c_{j+1}, \ldots, c_l) = \infty$;

4°. $\lim\limits_{x_j \to -\infty} \psi_l^j(c_1, \ldots, c_{j-1}, x_j, c_{j+1}, \ldots, c_l) = -\infty$.

Conditions 3° and 4° are satisfied for any constant in $c_1, \ldots, c_{j-1}, c_{j+1}, \ldots, c_l$.

The decision rule where all functions of the basis satisfy conditions 1° through 4° is a monotone rule.

We say that $L(\mathfrak{M}(R_A))$ contains constants if for any initial information $(I_0(l), I_S(q))$ there exists R_A in $L(\mathfrak{M}(R_A))$ such that $R_A(I_0(l), I_S(q)) = \{a_{ij}\}_{q \times l}$, where $a_{ij} \equiv 1$.

Theorem 17. *The regular model whose linear closure contains constants and whose decision rules are monotone functional threshold rules of order l is a well-defined model.*

Proof. It is exactly the same as that of Theorem 16. So, we give only an outline without going into a detailed explanation of individual steps.

Suppose $c_1^* = \min\limits_j c_{1j}$ and $c_2^* = \max\limits_j c_{2j}$.

Divide the pairs of objects numbered (i, j) into sets M_0 and M_1 using the information vector matrix $\{\tilde{\alpha}(S_i')\}_{q \times l}$.

Using the property of regularity and construction of the theorem, form on the basis of $(I_0(l), I_S(q))$ a matrix $\{a_{ij}\}_{q \times l}$ such that its elements a_{ij}, $(i, j) \in M_1$ are so great that $\psi_l^j(a_{ij} - a_{i1}, \ldots, a_{ij}, \ldots, a_{ij} - a_{il}) \gg c_2^*$, and a_{ij}, $(i, j) \in M_1$, are significantly greater than a_{ij} for $(i, j) \in M_0$.

The difference between a_{ij}, $(i, j) \in M_1$ and a_{ij} $(i, j) \in M_0$ is chosen to make possible a shift of the elements of the matrix $\{a_{ij}\}$ (the subtraction of the constant N from the elements of the matrix) such that

$$\begin{aligned}
&\psi_l^j(a_{ij} - a_{i1}, \ldots, a_{ij} - N, \ldots, a_{ij} - a_{il}) > c_2^*, \quad (i, j) \in M_1, \\
&\psi_l^j(a_{ij} - a_{i1}, \ldots, a_{ij} - N, \ldots, a_{ij} - a_{il}) < c_1^*, \quad (i, j) \in M_0.
\end{aligned} \tag{72}$$

Any shift by N is feasible because the model contains constants.

The existence of N such that conditions (72) are satisfied is guaranteed by the regularity of the model and by properties 1° through 4° of the basis of the decision rule.

CHAPTER IV
OPERATIONS ON RECOGNITION ALGORITHMS

Similarly to how the addition of recognition operators and their multiplication by a scalar was introduced in Chapter III, we can treat operations on recognition algorithms as operations on the information matrices

$$\{\alpha_{ij}^A\}_{q \times l} = A(I_0(K_1, \ldots, K_l), I(S_1', \ldots, S_q')).$$

However, whereas the set of recognition operators forms linear vector space, i.e., algebra with "good properties," such good operations are nonexistent in sets of information matrices. We can introduce and analyze operations with poorer properties. By applying them to simple models of recognition algorithms, we can obtain and investigate more sophisticated models.

§ 1. Element-Wise Operations on Information Matrices

Let the addition $A + B$ of recognition algorithms be defined as the element-wise addition of appropriate information matrices. That is, if $A(I_0) = \{\alpha_{ij}^A\}$ and $B(I_0) = \{\alpha_{ij}^B\}$, then $(A + B)(I_0) = \{\alpha_{ij}^A + \alpha_{ij}^B\}$. Since the elements of information matrices take only the values $\{0, 1, \Delta\}$, the addition of algorithms in this case is defined by specifying the operation on a set of three elements. Then it is natural to require that the following conditions be satisfied:

1°. Associativity;

2°. $a + a = a$, preservation of an element;

3°. $0 + \Delta \in \{0, \Delta\}$, $1 + \Delta \in \{1, \Delta\}$, preservation of a set;

4°. commutativity.

It is easy to prove

Theorem 18. *There exist seven operations of addition of algorithms with properties* 1° *through* 4°. *All of these operations are listed in Table* 4.

Proof. That conditions 1° through 4° are satisfied for the $\oplus$ operations specified by Table 4 can be checked directly. The other operations (totaling 2^4) do not have the property of associativity. Suppose, for example, $0 + 1 = \Delta$. We set $0 + \Delta = a_{0\Delta}$ and $1 + \Delta = a_{\Delta}$. Then

1. $a_{0\Delta} = 0$, $a_{1\Delta} = 1$, then $(0 + 0) + 1 = \Delta \neq 0 + (0 + 1) = 0$.

2. $a_{0\Delta} = 0$, $a_{1\Delta} = \Delta$, then $(0 + 0) + 1 \neq 0 + (0 + 1)$.

3. $a_{0\Delta} = \Delta$, $a_{1\Delta} = \Delta$, operation 1, element 0 of operation is absent.

4. $a_{0\Delta} = \Delta$, $a_{1\Delta} = 1$, then $(1 + 1) + 0 \neq 1 + (1 + 0)$.

The other cases can be analyzed in the same way. The theorem is proved.

Corollary. *Of the seven semigroup operations in Table* 4, *none is a group operation. Two of them have a zero operation element.*

None of the operations introduced in Table 4 are interesting.

Suppose that, on evaluating a property $S_i' \in K_j$, algorithms $A_1, A_2, \ldots, A_n$ delivered the answers α_{ij}^1, $\alpha_{ij}^2, \ldots, \alpha_{ij}^n$. Then $\alpha_{ij}^1 \oplus \ldots \oplus \alpha_{ij}^n = \Delta$ if among α_{ij}^q there is at least one Δ or $\alpha_{ij}^q \not\equiv 1$ or $\alpha_{ij}^q \not\equiv 0$ under the first operation ($q = 1, 2, \ldots, n$). The remaining operations reduce to specifying the order in $\{0, 1, \Delta\}$:

2nd: $\Delta > 1 > 0$;
3rd: $1 > \Delta > 0$;
4th: $1 > 0 > \Delta$;
5th: $\Delta > 0 > 1$;
6th: $0 > \Delta > 1$;
7th: $0 > 1 > \Delta$.

The application of these operations to $\alpha_{ij}^1 \oplus \ldots \oplus \alpha_{ij}^n$ yields a result equal to the maximum element in the respective order in the set $\{\alpha_{ij}^1, \ldots, \alpha_{ij}^q\}$. This means, for example, that under operation 4 one positive answer, $S_i' \in K_j$, cancels any count of negative answers $S_i \overline{\in} K_j$ and "I do not know" answers.

§ 2. Multiplace Operations on Information Matrices

The results in § 1 show that operations on information matrices and, hence, on recognition algorithms are bound to depend significantly on the number of algorithms involved in a given operation and, possibly, on the order in which the algorithms are applied to the initial information. Such operations were applied in developing new models. Here are two examples.

I. Consider R algorithms $A_1, \ldots, A_n$ with linear separating surfaces $R_1, \ldots, R_n$. The problem is one of classification with two intersecting classes, K_1 and K_2. With respect to the object S_i', each of the algorithms A_u delivers the answer: $S_i' \in K_1$ ($R_u(S_i') \geq 0$, $\alpha_{i1} = 1$) or $S_i' \overline{\in} K_1$ ($R_u(S_i') < 0$, $\alpha_{i1}^n = 0$). To the vector $(\alpha_{i1}^1 \ldots \alpha_{i1}^n)$ thus obtained, we apply the Boolean function $l(x_1, \ldots, x_n)$

$$f(\alpha_{i1}^1, \ldots, \alpha_{i1}^n) = \begin{cases} 1, & \text{if the number of ones among } (\alpha_{i1}^1 \ldots \alpha_{i1}^n) \text{ is greater than or equal to } \left[\frac{n}{2}\right] + 1, \\ 0 & \text{in all other cases.} \end{cases}$$

Table 4

α	β	$\alpha + \beta$						
		1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
0	1	Δ	1	1	1	0	0	0
0	Δ	Δ	Δ	Δ	0	Δ	0	0
1	0	Δ	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1
1	Δ	Δ	Δ	1	1	Δ	Δ	1
Δ	0	Δ	Δ	Δ	0	Δ	0	0
Δ	1	Δ	Δ	1	1	Δ	Δ	1
Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ

That is how the committee method is described [61].

II. The descriptions of objects are defined by collections of binary features. It is known that objects $S_1, \ldots, S_m$ are members of K_1 and objects $S_{m+1}, \ldots, S_t$ are members of K_2. The classes are disjoint.

Consider the not everywhere defined functions

$$F_1(I(S_i)) = 1, \; i = 1, 2, \ldots, m;$$

$$F_1(I(S_i)) = 0, \; i = m+1, \ldots, t,$$

on the other collections, the function F_1 is not defined;

$$F_2(I(S_i)) = 0, \; i = 1, 2, \ldots, m;$$

$$F_2(I(S_i)) = 1, \; i = m+1, \ldots, t,$$

on the other collections, the function F_2 is not defined.

Consider contracted or irreducible DNF's Φ_1 and Φ_2 which implement the functions F_1 and F_2, respectively [5, 6].

Let S be an object to be recognized and $I(S)$ be its description by the system of binary features.

Algorithm A_1: $\alpha_1^1(S) = 1$ $(S \in K_1)$ if $F_1(I(S)) = 1$,
$\alpha_1^1(S) = 0$ $(S \overline{\in} K_1)$ if $F_1(I(S)) = 0$.

Algorithm A_2: $\alpha_1^2(S) = 1$ $(S \in K_1)$ if $F_2(I(S)) = 0$.
$\alpha_1^2(S) = 0$ $(S \overline{\in} K_1)$ if $F_2(I(S)) = 0$.

Algorithm A_3: This is an estimation algorithm with the learning information $I(S_1)$, $\tilde{\alpha}(S_1), \ldots, I(S_t)$, $\tilde{\alpha}(S_t)$. The algorithm is chosen so that $\alpha_1^3(S) \in \{0, 1\}$.

Perform on the collection $\alpha_1^1(S)$, $\alpha_1^2(S)$, $\alpha_1^3(S)$ the operation defined by the Boolean function $f(x, y, z)$: if

$x = y$, then $f(x, y, z) = x$; if $x \neq y$, then $f(x, y, z) = z$. The algorithm that reduces to performing f on $\alpha_1^1(S)$, $\alpha_1^2(S)$, $\alpha_1^3(S)$ is described in [5, 6]. The function f admits a simple notation: $f = x \cdot (y \vee z) \vee y \cdot z$. It implements the committee method as applied to algorithms A_1, A_2, and A_3.

The examples analyzed above make the basis for the following definitions.

Let there be given the model $\mathfrak{M}$ of recognition algorithms and all possible finite ordered collections of algorithms $A_1, \dots, A_n$ in $\mathfrak{M}$. Let the evaluation of the predicate $P_j(S_i') = (S_i' \in K_j)$ by the algorithms $A_1, \dots, A_n$ result in $\alpha_{ij}^1, \dots, \alpha_{ij}^n$, $\alpha_{ij}^t \in \{0, 1, \Delta\}$, respectively.

Definition 33. In a model $\mathfrak{M}$, the operation $F = (f_1(x), \dots, f_n(x_1, \dots, x_n), \dots)$ is specified, if a sequence $f_1, \dots, f_n, \dots$ of ternary logic functions is defined.

The arguments of the functions f_i take values from the set $\{0, 1, \Delta\}$ and the domain of definition for f_i is likewise $\{0, 1, \Delta\}$.

We use the symbolic notation $f_n(A_1, \dots, A_n)$ for the new algorithm $(A_1, \dots, A_n, f_n)$:

$$(A_1, \dots, A_n, f_n)(\alpha_{ij}^1, \dots, \alpha_{ij}^n) = f_n(\alpha_{ij}^1, \dots, \alpha_{ij}^n).$$

The collection of algorithms $(A_1, \dots, A_n, f_n)$, $n = 1, 2, \dots, r, \dots$ is denoted by $F(\mathfrak{M})$ (here we consider all possible ordered collections of algorithms $A_1, \dots, A_n$ from the model $\mathfrak{M}$).

Definition 34. The set $F(\mathfrak{M})$ is called the functional closure of $\mathfrak{M}$ over the sequence F.

Since there exist 3^{3n} distinct ternary logic functions of variables $x_1, \dots, x_n$, the function f_n in F can be chosen in 3^{3n} distinct ways. Therefore, a very great number of different functional closures $F(\mathfrak{M})$ can be constructed from the model $\mathfrak{M}$ by means of different F.

Note that ternary logic functions were investigated in detail by Yablonskii [74]. On this basis and adding some later studies [1, 2, 59], we can offer formulas that present ternary logic functions as superpositions of elementary functions. A convenient system of these elementary functions is

$$f_1(x_1) \equiv 0, \quad f_2(x) \equiv 1, \quad f_3(x) \equiv \Delta,$$

$$j_i(x) = \begin{cases} \Delta & \text{for } x = i \\ 0 & \text{for } x \neq i, \end{cases}$$

where $i = 0, 1, \Delta$, $\min(x, y)$, and $\max(x, y)$. The last two functions are defined relative to the order $0 < 1 < \Delta$ [74].

The presentation of ternary logic functions as the superposition of the above elementary functions is similar to the presentation of a Boolean function as DNF. Indeed,

$$f(x_1, \dots, x_{n-1}, x_n) = \max\{\min j_0(x_n),$$
$$f(x_1, \dots, x_{n-1}, 0), \min[j_1(x_n), f(x_1, \dots, x_{n-1}, 1)],$$
$$\min[j_\Delta(x_n), f(x_1, \dots, x_{n-1}, \Delta)]\}.$$

Thus, formulas can be set up for functions in F and the algorithms $(A_1, \dots, A_n, f_n)$ operate efficiently.

For the considerations of content, F can not include arbitrary f_n.

Consider the classes of ternary logic functions from which the functions $f_n(x_1, \dots, x_n)$ can be chosen in F.

The first series of constraints is:

1°. $f_n(a, a, \dots, a, a) = a$, $a \in \{0, 1, \Delta\}$;

2°. $f_n(a_1, \dots, a_n) \in \{a_1, \dots, a_n\}$ if $\{a_1, \dots, a_n\}$ is $\{0, \Delta\}$ or $\{1, \Delta\}$.

Indeed, if none of the algorithms $A_1, \dots, A_n$ evaluated $(S \in K_j) = 1$, then it makes no sense to combine the results and to assume that $(S \in K_j) = 1$. The same holds for $(S \in K_j) = 0$.

The set of f that satisfy conditions 1° and 2° form a closed class σ_1 of the functions that preserve the sets $\{0\}$, $\{1\}$, $\{\Delta\}$, $\{0, \Delta\}$ and $\{\Delta, 1\}$. This class has a finite basis.[3]

The second series of conditions is as follows. Let $f_n(\alpha_1, \alpha_2, \dots, \alpha_n) = \alpha$, $\alpha \in \{0, 1, \Delta\}$. Replace any of the values $\alpha_1, \alpha_2, \dots, \alpha_n$ by α. We obtain a new collection, $(\alpha_1', \dots, \alpha_n')$:

3°. $f_n(\alpha_1', \dots, \alpha_n') = \alpha$.

The system of the functions f that satisfy condition 3° forms the class σ_2.

It is natural to consider below the functions in F from $\sigma_1 \cap \sigma_2$.

The third series of conditions is as follows. Let x, y and z be the numbers 0, 1 and Δ in the collection $x_1, \dots, x_n$.

4°. $f_n(x_1, \dots, x_n) \equiv \varphi(x, y, z)$.

The functions that satisfy 4° do not form a closed class.

5°. Suppose $\varphi(x, y, z) = f_n(x_1, \dots, x_n) = \alpha$.

Then, if the collection $(x_1', \dots, x_n')$ has x' ones, y' zeros, and $z' - \Delta$ symbols, such that

(a) $x' \geq x$, $y' \leq y$, $z' \leq z$,

(b) $x' \leq x$, $y' \geq y$, $z' \leq z$,

(c) $x' \leq x$, $y' \leq y$, $z' \leq z$,

and if $\alpha \in \{0, 1, \Delta\}$ and the corresponding conditions (a), (b) and (c) were satisfied, then

$$f_n(x_1', \dots, x_n') = \varphi(x', y', z') = \alpha.$$

[3] It was Yablonskii who told the author that the class σ_1 has a finite basis.

The functions that satisfy conditions 4° and 5° are naturally called committee functions. The class of these functions is denoted by K.

The models $F(\mathfrak{M})$ are naturally generalized into models $(F_1 \ldots F_l)(\mathfrak{M})$, where the property $S \in K_j$ is evaluated for each K_j by the functions F_j $(j = 1, 2, \ldots, l)$.

§ 3. Other Operations on Recognition Algorithms

1. The product of recognition operators. Let the operators R_A in the model $\mathfrak{M}$ include in their domain the learning informations where the objects are described by collections of numerical features.

Suppose $R_A(I_0(l), I_S(q)) = \{a_{ij}\}_{q \times l}$.

Then

$$R_A(I(S_1), \tilde{\alpha}(S_1), \ldots, I(S_m), \tilde{\alpha}(S_m); I(S_1, \ldots, S_m))$$

$$= \{b_{ij}\}_{m \times l} = R_A(I_0(l), I_S(m)).$$

Suppose $I^A(S_i) = (b_{i1}, \ldots, b_{il})$ and $I^A(S'_u) = (a_{u1}, \ldots, a_{ul})$ $(i = 1, 2, \ldots, m,\ u = 1, 2, \ldots, q)$. Then

$$R_{A'}(I^A(S_1), \tilde{\alpha}(S_1), \ldots, I^A(S_m), \tilde{\alpha}(S_m), I^A(S'_1 \ldots S'_q))$$

$$= \left\{ a'_{ij} \right\}_{q \times l} = R_A \cdot R_A \cdot (I_0(l), I_S(q)).$$

The successive application of R_A and $R_{A'}$ to $(I_0(l), I_S(q))$ is called the product of R_A and $R_{A'}$ (denoted by $R_A \cdot R_{A'}$).

By considering the operations of addition, multiplication, and multiplication by a scalar on recognition algorithms we can easily introduce polynomials from recognition operators.

2. Additions to the feature space. The estimates $(a_{i1}, \ldots, a_{il})$ calculated by operators for S'_i and similar estimates for $S_1, \ldots, S_m$ can be introduced as values of new features in $I(S'_i)$ and $I(S_u)$ $(i = 1, 2, \ldots, q,\ u = 1, 2, \ldots, m)$. A new feature is interpreted as follows: The algorithm A (the operator R_A) established that the estimate of S'_i (S_u) for class K_j is a_{ij} (a_{uj}). By the same token, new features can be added when the algorithm A, $A \in \mathfrak{M}$, established that the property $S'_i \in K_j$ has the value $\alpha_{ij}^A \in \{0, 1, \Delta\}$.

The operations examined in this section can derive new models from those constructed previously.

CHAPTER V
METHODS FOR CONSTRUCTION OF STANDARD RECOGNITION ALGORITHMS OPTIMAL IN TERMS OF THE QUALITY FUNCTIONAL

§ 1. Optimization Within One Parametric Model

Consider the model $\mathfrak{M}\{A\} = \{R_A r_A\}$ of recognition algorithms represented as $(R_A r_A)$ such that $R_A(I_0(l), I(S'_1, \ldots, S'_q)) = \{a_{ij}^A\}_{q \times l}$, and $r_A\{a_{ij}\}_{q \times l} = \{\alpha_{ij}^A\}$, where a_{ij}^A are numbers and $\alpha_{ij}^A \in \{0, 1, \Delta\}$ $(i = 1, 2, \ldots, q, j = 1, 2, \ldots, l)$.

We assume that the matrix $\{\alpha_{ij}^A\}$ is defined by learning information, by the collection of descriptions of objects to be recognized, and by the collection of numerical parameters $\pi_1^1, \ldots, \pi_t^1$, so that

$$a_{ij}^A = a_{ij}^A(\pi_1^1, \ldots, \pi_t^1).$$

The matrix $\{\alpha_{ij}^A\}$ is defined by the matrix $\{a_{ij}^A\}$ and by the collection of numerical parameters $\pi_1^2, \ldots, \pi_v^2$. In this case, the model $\mathfrak{M}$ is called a parametric model.

We consider functional decision rules with threshold constants c_{1j} and c_{2j}.

The linear forms L_j $(j = 1, 2, \ldots, l)$ of all or some elements of the matrix $\{a_{ij}\}_{q \times l}$ are usually taken to form the basis of decision rules.

In what follows, we consider solely linear quality functionals (Definition 10 in Chapter I).

The problem of selecting from $(I_0(l), I_S(q))$ the algorithm A extremal in the model $\mathfrak{M}$ consists in finding a collection of parameters $\hat{\pi}_1^1, \ldots, \hat{\pi}_t^1$ of the recognition operator and $\hat{\pi}_1^2, \ldots, \hat{\pi}_v^2$ of the decision rule which defines the algorithm $\hat{A} \in \mathfrak{M}$ such that

$$\varphi(\hat{A}) = \varphi(\hat{\pi}_1^1, \ldots, \hat{\pi}_t^1, \hat{\pi}_1^2, \ldots, \hat{\pi}_v^2) = \max_{A \in \mathfrak{M}} \varphi(A)$$

$$= \max_{A \in A(\pi_1^1, \ldots, \pi_t^1, \pi_1^2, \ldots, \pi_v^2)} \varphi(\pi_1^1, \ldots, \pi_t^1, \pi_1^2, \ldots, \pi_v^2).$$

The second problem is to select in $\mathfrak{M}$ a set $\hat{\mathfrak{M}}$ that contains algorithms extremal in terms of an arbitrary $(I_0(l), I_S(q))$, where $I_0 \in \{I_0\}$ is the set of admissible learning informations and $I(S_1^1), \ldots, I(S'_q)$ are admissible descriptions of an arbitrary finite collection of admissible objects.

To begin with, consider the first problem. Let there be given a functional decision rule $f_j(x_{11}, \ldots, x_{q \cdot l})$ with threshold constants c_{1j} and c_{2j}. Then the value which the function $f_j(a_{11}, \ldots, a_{q \cdot l})$ takes on the elements of the matrix $\{a_{ij}\}$ determines the choice of the number 1, 0, or Δ, according to which the following inequality is satisfied: $f_j > c_{2_j}$ or $f_j < c_{1j}$ $(c_{1j} \le f_j \le c_{2j})$. From this value, the reward γ_{ij} $(i = 1, 2, \ldots, q, j = 1, 2, \ldots, l)$ is then uniquely determined. Recall that f_j can depend solely on a part of the element $a_{11}, \ldots, a_{q \cdot l}$ $(f_j = f(a_{ij}), f_j = f_j(a_{i1}, \ldots, a_{il}))$.

Having a method whereby the quality functional can be evaluated for any fixed collection of parameter values, we can seek an extremum algorithm using the local descent method, the random search method, the directed search method, or a combination thereof. A great number of such studies were undertaken for estimation algorithms [15–17, 23, 38, 39, 42] and for other algorithm models.

Special mention should be made of the studies reported in [40, 41, 70], which offer a way to find a good initial approximation for model parameters.

Another idea advanced in [20, 21, 62] is that the quality functional is replaced by a simpler functional and optimal parameter values are found for it explicitly or by simple computational procedures. The applicable optimization methods are many and diverse.

In this area, there is one large subclass of problems that can be described and solved in a unified manner [35].

Consider a functional $\varphi(A)$ which is equal to the fraction of correct predictions (Definition 10). Given threshold functional decision rules, we can easily write the condition for $P_j(S_i^{'})$ to be correct calculated as an inequality. If $P_j(S_i^{'}) = 0$, then

$$f_j(a_{i1}, \ldots, a_{il}) < c_{1j} \tag{73}$$

or

$$f_j(a_{\iota 1}(\pi_1^1, \ldots, \pi_t^1), \ldots, a_{il}(\pi_1^1, \ldots, \pi_t^1)) < c_{1j}.$$

For $P_j(S_\iota^{'}) = 1$,

$$f_j(a_{11}(\pi_1^1, \ldots, \pi_t^1), \ldots, a_{il}(\pi_1^1, \ldots, \pi_t^1)) < c_{2j}. \tag{74}$$

Writing $q \cdot l$ inequalities of the form (73) or (74) for each pair $(K_j, S_i^{'})$ gives the system

$$\begin{gathered} f_j(a_{i1}, \ldots, a_{il}) \lesseqgtr c_{uj}, \quad u \in \{1, 2\}, \\ i = 1, 2, \ldots, q, \quad j = 1, 2, \ldots, l. \end{gathered} \tag{75}$$

The optimal quality functional defined by Table 3 corresponds to the collection of parameters $\pi_1^1, \ldots, \pi_t^1$ that define f_j and of threshold constants c_{1j} and c_{2j} such that the maximum number of inequalities is satisfied in (75). To find an algorithm optimal in terms of the quality functional, it is necessary to pick in (75) a maximally consistent subsystem and to solve it.

The parameters thus found define the optimum algorithm.

The problem of selecting a maximally consistent subsystem in a system of inequalities for any assortment of functions f_j can be reduced to a standard discrete extremum problem.

Assign Boolean variables $y_1, \ldots, y_{q \cdot l}$ to the inequalities in (75). Introduce the Boolean function $\chi(y_1, \ldots, y_{q \cdot l})$ as follows. Let $\tilde{\alpha} = (\alpha_1, \ldots, \alpha_{q \cdot l})$ be a Boolean vector such that $\alpha_i \in \{0, 1\}$. In $\tilde{\alpha}$, we select all coordinates that take value 1. Let them be numbered $i_1, i_2, \ldots, i_k$.

In (75), we select a subsystem of inequalities composed of the inequalities numbered $i_1, i_2, \ldots, i_k$. We denote this subsystem by $\sigma(i_1, i_2, \ldots, i_k)$ or by $\sigma(\tilde{\alpha})$.

We put

$$\chi(\alpha_1, \ldots, \alpha_{q \cdot l}) = \begin{cases} 1, & \textit{if the subsystem } \sigma(i_1, \ldots, i_k) \\ & \text{is inconsistent} \\ 0, & \text{if the subsystem } \sigma(i_1, \ldots, i_k) \\ & \text{is consistent.} \end{cases}$$

Recall the definition of the monotone Boolean function f. Let, on a set of binary collections $\tilde{\alpha}$ of the same length, there be specified the partial order relation: $\tilde{\alpha} = (\alpha_1 \ldots \alpha_n) \le \tilde{\beta} = (\beta_1 \ldots \beta_n)$ if $\alpha_i \le \beta_i$ $(i = 1, 2, \ldots, n)$. The function f is monotone if the relation $\tilde{\alpha} \le \tilde{\beta}$ implies that $f(\tilde{\alpha}) \le f(\tilde{\beta})$. Upper 0 (or lower 1) of the monotone f is a point $\tilde{\gamma}$ such that $f(\tilde{\gamma}) = 0$ (or $f(\tilde{\gamma}) = 1$) and $f(\tilde{\delta}) = 1$ for all $\tilde{\delta} \ne \tilde{\gamma}$ such that $\tilde{\delta} \ge \tilde{\gamma}$ (or $f(\tilde{\delta}) = 0$ for $\tilde{\delta} \le \tilde{\gamma}$).

The upper zero $\tilde{\alpha}$ is called maximal (or the lower unit $\tilde{\alpha}$ is called minimal) if $\|\tilde{\alpha}\|$ is maximal (or minimal) over the set of upper 0's (or of lower 1's). As usual, by the norm of the binary vector $\|\alpha\|$ is meant the number of unit coordinates in $\tilde{\alpha}$.

The consistent subsystem of inequalities is a irreducible one if the addition of any inequality makes it inconsistent.

The monotone function is uniquely defined by specifying the set of its lower 1's or of its upper 0's. With respect to the function χ, there is

Theorem 19. *The function* $\chi(x_1, \ldots, x_{q \cdot l})$ *is a monotone Boolean function. The set of consistent subsystems* (75) *is in one-to-one correspondence with the set of* 0's *of the function* χ. *The set of irreducible subsystems is in one-to-one correspondence with the set of upper* 0's *of the function* χ. *The set of upper* 0's *with a maximum norm is in one-to-one correspondence with the set of maximally consistent subsystems.*

Proof. By its construction, the set of 0's of the function χ is in one-to-one correspondence with the set of consistent subsystems of the system (75).

The monotony of the function χ follows from two obvious assertions:

(a) If an inequality is added to an inconsistent subsystem, the new subsystem remains inconsistent.

(b) If an inequality is removed from a consistent subsystem, the new subsystem remains consistent.

If a subsystem corresponds to other than a maximal upper zero $\tilde{\alpha}$, then a $\tilde{\beta}$ can be found such that $\chi(\tilde{\beta}) = 0$, $\tilde{\beta} \geq \tilde{\alpha}$; all unit coordinates $\tilde{\alpha}$ retain that value and at least one new unit coordinate is added. This new coordinate is numbered u. By construction of χ, the system $\sigma(\tilde{\alpha}) \cup u$ is consistent, and, hence, the system $\sigma(\tilde{\alpha})$ is not a irreducible one. The correspondence of maximal upper 0's to maximal consistent subsystems is obvious. The theorem is proved.

We reduced the problem of finding a maximally consistent subsystem in a system of inequalities to the search for the maximal upper 0 of a monotone Boolean function.

This latter problem is a well-known problem and its solution was the subject of many studies. Some of them [51, 52] were done in terms of the theory of monotone Boolean functions and others in terms of the theory of inequalities. They can be restated in terms of the theory of monotone Boolean functions, though. The standard formulation in the search for the maximal upper 0 is this. Let there be specified an elementary operator B which, given a point $\tilde{\alpha}$, evaluates $\chi(\tilde{\alpha})$, supplements the definition of χ in terms of monotony, and verifies the termination criterion. The termination at $\tilde{\alpha}$ guarantees that $\tilde{\alpha}$ is the maximal upper 0 of the function χ.

It is required to construct an algorithm that can indicate an upper 0 for χ with the operator B applied a number of times which is minimal for the given χ. Actually, the application of the operator B is not always efficient because there is a need, in applying the operator B, to verify the consistency of a subsystem. We do not dwell on this issue in this paper.

The problem of minimizing the number of times for the application of the operator B was solved in Shannon's statement [43].

We denote the minimal number of times that the operator B is applied in the search for the maximal upper 0 of the function $f(x_1, \ldots, x_n)$ by $\mu_f(n)$ and $L(n) = \max \mu_f(n)$ over the monotone functions of n variables. It is required to evaluate $L(n)$ and to construct the algorithm that would guarantee that for any monotone $f(x_1, \ldots, x_n)$ the maximal upper 0 is found in at most $L(n)$ steps. This algorithm was constructed in [43] where it is shown that

$$L(n) = C_n^{\left[\frac{n}{2}\right]+1}.$$

Actually, the algorithm optimal in Shannon's sense has a limited domain of applicability ($L(n)$ increases with increasing n as $c\frac{2^n}{\sqrt{n}}$). The algorithms used in reality either (1) limit themselves to finding a certain number of upper 0's and selecting an element with a maximum norm among them [51, 52] or (2) start from a certain initial point $\tilde{\alpha}$, where $\chi(\tilde{\alpha})$ is evaluated and conduct a directed selection of subsequent points for the application of B, considering the information accumulated about the function χ. The procedure can be terminated at any step and 0 is delivered, having the maximum norm among the other 0's. Here is one example of this procedure [35].

To begin with, several points $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$ are chosen at random, the operator B is applied there and the functions $\chi(\tilde{\alpha}_1), \ldots, \chi(\tilde{\alpha}_t)$ are evaluated. The procedure is carried on until the values thus obtained include at least one zero and at least one unit. Among $\chi(\tilde{\alpha}_i)$ ($i = 1, 2, \ldots, t$), we chose zero with a maximum norm N and put $\chi = 0$ at all points where the norm is not greater than N.

The number of these points is denoted by Q_0 and their set by M_0. Count the number of points $\tilde{\beta}_t$ located in the set $E_n \backslash M_0$, not smaller than the number of 1s of the function χ (the set M_1). We denote the number of these points by Q_1. We set $\chi = 0$ on M_0 and $\chi = 1$ on M_1.

The above procedure is called the recovery of the function that has at least one upper 0 with a maximum norm, common with χ, or the χ recovery.

In the set $\chi(\tilde{\alpha}_1), \ldots, \chi(\tilde{\alpha}_t)$, we fix at least one 0 with a maximum norm. This 0 is delivered when the procedure terminates after calculations at $\tilde{\alpha}_1, \ldots, \tilde{\alpha}_t$.

The next point is selected by applying the following procedure.

Suppose

$$p_0(\chi) = \frac{Q_0}{Q_0 + Q_1}, \quad p_1(\chi) = \frac{Q_1}{Q_0 + Q_1},$$

where $\tilde{\beta}$ is an arbitrary point where the value of χ is not known in the course of χ recovery. This point is assigned the quantity

$$Q(\tilde{\beta}) = p_0 \Delta_0(\tilde{\beta}) + p_1 \cdot \Delta_1(\tilde{\beta}), \qquad (75')$$

where $\Delta_0(\tilde{\beta})$ and $\Delta_1(\tilde{\beta})$ are the numbers of points where the value of χ is not known or where it will be known provided $\chi(\tilde{\beta}) = 0$ or, respectively, $\chi(\tilde{\beta}) = 1$ after the χ recovery from the point $\tilde{\beta}$ of the function χ constructed previously. For the next application of B, we select the point $\tilde{\beta}$ at which the procedure yields $\max\limits_{\tilde{\beta}}\chi(\tilde{\beta})$.

In actual computational schemes, the form of (75') can be modified: either an upper or a lower estimate Q can be calculated instead of the exact value $Q(\tilde{\beta})$, or, lastly, $\tilde{\beta}$ can be selected to deliver a local extremum for $Q(\tilde{\beta})$.

If $\chi(\tilde{\beta}) = 0$, then the point $\tilde{\beta}$ is fixed; if $\chi(\tilde{\beta}) = 1$, then 0 selected at the previous step is retained. In the case of termination at some step, the fixed 0 is delivered.

The second problem, that of identifying the set which is known to contain all algorithms extremal on arbitrary finite samples of admissible objects, is solved as follows.

Consider a submodel $\mathfrak{M}'(A)$ in the model $\mathfrak{M}(A)$, and ascertain whether the model $\mathfrak{M}'(R_A)$ is complete or weakly complete.

If $\mathfrak{M}'(R_A)$ is a complete model and r is an arbitrary well-defined decision rule, then the model $L\{\mathfrak{M}'(R_A) \cdot r\}$ obviously satisfies the statement of the second problem.

In the case of weak completeness, an arbitrary functional decision rule with a basis of monotonically increasing functions can be taken as r. In some cases, it can be additionally required for the threshold constants to be strictly positive.

For example, it is shown in Chapter III that in an estimates calculation model with parameters $(k, \varepsilon, \vec{\varepsilon}, \vec{p}, \vec{\gamma})$, it would suffice to set $k = 1$ and $\varepsilon = 0$ and to seek an extremum solely on the operators representable by linear combinations of operators in $L\{\mathfrak{M}(\vec{\varepsilon}, \vec{p}, \vec{\gamma})\}$.

§ 2. Optimization in Linear Closures

Let there be given parametric models $\mathfrak{M}_1 = \{(R_{A_1} r_{A_1})\}, \ldots, \mathfrak{M}_k = \{(R_{A_k} r_{A_k})\}$. Form a linear closure $L(\mathfrak{M}_1, \ldots, \mathfrak{M}_k)$; that is, a collection of algorithms $L = \{c_1 R_A^{1t} + \ldots + c_k R_A^{kt}, r\}$, where $r \subseteq \bigcup_{\iota=1}^{k} r_A^i$ and $R_A^{it} \in R_A^i$ $(i = 1, 2, \ldots, k)$. Then $L(\mathfrak{M}_1, \ldots, \mathfrak{M}_k)$ is again a parametric model: the parameters of models $\mathfrak{M}_1, \ldots, \mathfrak{M}_k$ are combined and the parameters $c_1, \ldots, c_k$ are added. In the new model, we can again solve the optimization problem. But with large k or when some $\mathfrak{M}_i$ are themselves specified by a large number of parameters, the real optimization in L runs into major difficulties. An approximate optimization can be achieved in two stages, as follows.

1°. Select extremum algorithms A_i^* in $\mathfrak{M}_i$ $(i = 1, 2, \ldots, k)$ (or representative algorithms in each of the models). Suppose $A_i^* = (R_{A_\iota}^*, r_{A_\iota})$.

2°. Consider the algorithms $(\sum c_i R_{A_\iota}^*, r_{A_\iota})$, where $r_{A_\iota} \in L$. From this set, select an algorithm extremal in terms of $c_1, \ldots, c_k$ and in terms of the parameters of the decision rule r_{A_ι}.

Let $R_{A_\iota}^*(I_0(l), I_S(q)) = \{a_{ut}^i\}$ $(i = 1, 2, \ldots, k)$. Hence, if $A = (R_A \cdot r)$, where $R_A = \sum\limits_{\iota=1}^{k} c_i R_{A_\iota}^*$, then

$$R_A(I_0(l), I_S(q)) = \left\{\sum_{i=1}^{k} c_i a_{ut}^i\right\} = \{d_{ut}\}_{q \times l}.$$

If decision rules are defined by the linear forms $b_1^j x_1 + \ldots + b_l^j x_l$ and $c_{1j} = c_{2j} = 0$, then the system in (75) consists of the inequalities

$$b_1^j\left(\sum_{i=1}^{k} c_i \cdot a_{uj}^i\right) + \ldots + b_l^j\left(\sum_{i=1}^{k} c_i \cdot a_{uj}^i\right) > 0 \qquad (76)$$

for pairs (S_u', K_j) such that $S_u' \in K_j$ and

$$b_1^j\left(\sum_{i=1}^{k} a_{tj}^i \cdot c_i\right) + \ldots + b_l^j\left(\sum_{i=1}^{k} c_i \cdot a_{tj}^i\right) < 0 \qquad (77)$$

for pairs (S_t', K_j) such that $S_t' \,\overline{\in}\, K_j$.

By combining (76) and (77), we obtain a system of $q \cdot l$ inequalities (or l in the case of disjoint classes) bilinear in $b_1^j, \ldots, b_l^j$ and $c_1, \ldots, c_k$ in $k \cdot l$ unknowns. Then the maximally consistent subsystem can be selected by any of the known methods (e.g., see [60]). Thus, once algorithms were chosen in each model, the coefficients of a linear combination of algorithms and the coefficients of linear decision rules can be found by solving a relatively simple extremum problem.

Note that linear closures can involve operators from one model but acting on different parts of the initial information.

§ 3. Optimization in Functional Closures

Similarly to § 2, let $\mathfrak{M} = \mathfrak{M}_1 \cup \ldots \cup \mathfrak{M}_n$, i.e., let $\mathfrak{M}$ be a system of models $\mathfrak{M}_1, \ldots, \mathfrak{M}_n$. Let an algorithm A_i^* $(i = 1, 2, \ldots, n)$, optimal with respect to the control sample $S_1', \ldots, S_q'$, be constructed in each of $\mathfrak{M}_i$. The objective of the optimization is to match a ternary logic function f_n such that $f_n(A_1^*, \ldots, A_n^*)$ is an algorithm optimal in terms of the quality functional $\varphi(A)$.

Consider linear functionals $\varphi(A)$. Suppose that the evaluation of the property $S_i' \in K_j$ by the algorithm A_u^* resulted in α_{ij}^u ($i = 1, 2, \ldots, q, j = 1, 2, \ldots, l$, and $u = 1, 2, \ldots, n$). Suppose that actually the predicate $S_i' \in K_j$ is equal to α_{ij}. Then, for $S_i' \in K_j$ to be evaluated correctly, the necessary and sufficient condition is

$$f_n(\alpha_{ij}^1, \ldots, a_{ij}^n) = \alpha_{ij}, \qquad \alpha_{ij}^t \in \{0, 1, \Delta\}, \quad \alpha_{ij} \in \{0, 1\}. \tag{78}$$

Similar conditions are written for all (i, j).

The result is a system of $q \cdot l$ equations for f_n. Generally, this system is inconsistent.

Suppose that in defining the function f_n the collection $\alpha_{ij}^1, \ldots, \alpha_{ij}^n$ is assigned the value $\tilde{\alpha}_{ij} \in \{0, 1, \Delta\}$. In a linear quality functional, the pair $(\alpha_{ij}, \tilde{\alpha}_{ij})$ is assigned a reward, $\gamma_{ij} = \gamma(a_{ij}, \tilde{\alpha}_{ij})$. The value of the functional is $\frac{1}{q \cdot l}\sum_{i, j}\gamma_{ij}$. The objective is to define f_n so as to maximize $\frac{1}{q \cdot l}\sum_{i, j}\gamma_{ij}$.

In the space of collections $(\alpha_{ij}^1, \ldots, \alpha_{ij}^n) = \vec{\alpha}_{ij}$ ($i = 1, 2, \ldots, q$ and $j = 1, 2, \ldots, l$, we select all subsets $\{M\}$ of identical collections.

Consider this arbitrary subset $M = \{\vec{\alpha}_{i_1 j_1}, \ldots, \vec{\alpha}_{i_t j_t}\}$. If it consists of a single element $\vec{\alpha}_{ij}$, then, in keeping with (78),

$$f_n(\vec{\alpha}_{ij}) = \alpha_{ij}. \tag{79}$$

Then the corresponding equation in (78) is satisfied. If the subset $M = \{\vec{\alpha}_{i_1 j_1}, \ldots, \vec{\alpha}_{i_t j_t}\}$ consists of more than one element, then, considering that $\vec{\alpha}_{i_1 j_1} = \ldots = \vec{\alpha}_{i_t j_t}$, we have $f_n(\vec{\alpha}_{i_1 j_1}) = \ldots = f_n(\vec{\alpha}_{i_t j_t}) = \alpha_{ij}^t$. The quantity α_{ij}^t can take one of three values, 0, 1, or Δ. Suppose $\alpha \in \{0, 1, \Delta\}$. Set

$$\varphi_M^\alpha(A) = \frac{1}{q \cdot l}\sum_{(i, j) \in \{(i_1, j_1), \ldots, (i_t j_t)\}} \gamma(\alpha, \alpha_{ij}). \tag{80}$$

Select the value α^* that leads to $\max\limits_{\alpha} \varphi_M^\alpha(A)$. It is assumed that $\alpha_{ij}^t = f_n(\vec{\alpha}_{i_1 j_1}) = \ldots = f_n(\vec{\alpha}_{i_t j_t}) = \alpha^*$.

By selecting the subsets M in turn, we define f_n on all collections $(\alpha_{ij}^1, \ldots, \alpha_{ij}^n)$ in a similar way.

Theorem 20. *The function f_n defined by the algorithm described above delivers an algorithm $f_n(A_1^*, \ldots, A_n^*)$ optimal in terms of the linear quality functional.*

Proof. Due to the linearity, $\varphi(A) = \sum\limits_{M \in \{M\}} \varphi_M^\alpha(A)$.

It suffices to demonstrate that all $\varphi_M^\alpha(A)$ are maximal on the selected f_n. For M composed of more than one element, the maximality of $\varphi_M^\alpha(A)$ follows from the construction of f_n on collections in M. For one-element M, the optimality of $\varphi_M^\alpha(A)$ follows from the constraint: $\gamma(\alpha_{ij}, \alpha_{ij}) \ge \gamma(\alpha_{ij}, \tilde{\alpha}_{ij})$ for $\tilde{\alpha}_{ij} \ne \alpha_{ij}$. The last inequality readily stems from the definition of the quality functional as a sequence of functions such that each function $f(\rho_1, \ldots, \rho_m)$ achieves its absolute maximum on the collection $(0, \ldots, 0)$. Since $\gamma(x, y) = \gamma(\rho(x, y))$, then $\gamma(\rho(\alpha_{ij}, \alpha_{ij})) = \gamma(0) \ge \gamma(\rho(\alpha_{ij}, \tilde{\alpha}_{ij}))$. The theorem is proved.

Note that it takes not more than $3 \cdot \left[\frac{q \cdot l}{2}\right]$ computations of functionals $\varphi_M^\alpha(A)$, such that $\alpha \in \{0, 1, \Delta\}$, to construct the optimal f_n.

The definition of f_n can be supplemented to include collections which are not members of $\{\alpha_{ij}^1, \ldots, \alpha_{ij}^n)\}$ in more than one way. For example, we can select a supplemental definition such that f_n is realized by the simplest formula in the selected basis. It is most convenient to consider the system of functions 0, 1, Δ, $j_0(x)$, $j_1(x)$, $j_\Delta(x)$, $\max(x, y)$, and $\min(x, y)$ in the order $0 < 1 < \Delta$.

For such a system, it is easy to introduce a ternary logic equivalent of DNF. Select a supplementary definition in which this equivalent DNF has a minimal complexity or cannot be simplified any more in a system of identity transformation (an equivalent of the irreducible DNF).

If f_n is chosen in the class of functions that satisfy the first series of constraints (§ 2, Chapter IV), then the construction of an optimal f_n is complicated only slightly. Then $f_n(0, \ldots 0) = 0$, $f_n(1, \ldots, 1) = 1$, and $f_n(\Delta, \ldots, \Delta) = \Delta$. On the sets M composed of collections containing solely $\{0, \Delta\}$, or solely $\{1, \Delta\}$, we can check the values 0, Δ or 1, Δ, respectively, disregarding the number of collections in M.

The problem of choosing an optimal f_n remain rather simple to achieve even if we require that f_n should satisfy both the first series of constraints and the constraint $f_n(x_1, \ldots, x_n) \equiv \varphi(x, y, z)$ from the third series. Then all classes M which have the same numbers of 1s and 0s are combined in one class $M(x, y\ z)$. On the collections of each class from $M(x, y\ z)$ the function f_n should then take one value. Therefore, for each $M(x, y, z)$, the value α is chosen from the sets $\{0, \Delta\}$, or $\{1, \Delta\}$, or $\{0, 1, \Delta\}$ according to $\varphi_{M(x, y, z)}^\alpha(A)$.

REFERENCES

1. Abdugaliev, U.A., On Normal Forms of k-valued Logic, *Kibernetika*, 1967, no. 1, pp. 16–20.
2. Abdugaliev, U.A., On the Representation of k-valued Logic Functions by Normal Forms: I., *Diskretnyi Analiz* (Translation), Novosibirsk: Inst. Math., Siber, Div., USSR Acad. Sci., 1969, no. 15, pp. 3–24.
3. Eizenberg, N.N. and Tsitkin, A.I., On the Use of Simple Testers, *Kibernetika*, 1974, no. 1, pp. 135–141.
4. Eizerman, M.A., Braverman, E.M., and Rozenoer, L.I., *Metod potentsial'nykh funktsii v teorii obucheniya mashin* (The Potential Function Method in the Theory of Machine Learning), Moscow: Nauka, 1970.
5. Aslanyan, L.A., On a Recognition Method Based on Class Separation by Disjunctive Normal Forms, *Kibernetika*, 1975, no. 5, pp. 103–110.
6. Aslanyan, L.A., Recognition Algorithms with Logical Separators, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 116–131.
7. Aliev, E.M. and Kamilov, M.M., Choice of Essential Parameters of an Industrial Process by the Estimates Calculation Method, *Trudy Mezhdunarodnogo simpoziuma po prakticheskim primeneniyam metodov raspoznavaniya obrazov* (Proc. Int. Symp. Applic. Pattern Recogn. Methods), Moscow: Computer Center, USSR Acad. Sci., 1973, pp. 219–225.
8. Bagranovskaya, Z.V., On Some Properties of V (Voting) Algorithms, *Kibernetika*, 1975, no. 6, pp. 96–104.
9. Bak Khyng Khang, On Geometrical Problems Related to Accuracy Estimation of a Class of Algorithms, *Zh, Vych. Mat. Mat. Fiz.*, 1974, vol. 14, no. 6, pp. 1571–1580.
10. Vapnik, V.N. and Chervonenkis, A.Ya., *Teoriya raspoznavaniya obrazov* (Theory of Pattern Recognition), Moscow: Nauka, 1974.
11. Guberman, Sh.A. and Maripov, T.M., Computer-assisted Assessment of Long-term Prospects of Hydrothermal Ores, *Trudy MINKh i GP im. Gubkina* (Translation), Moscow: Nedra, 1966, no. 62.
12. Gulyamov, D.Kh. and Zhuravlev, Yu.I., Quality Assessment of Experts Formulating the Learning Table for Pattern Recognition Problems, *Vych. Mat. Mat. Fiz.*, 1972, vol. 12, no. 4, pp. 1066–1070.
13. Gulyamov, D.Kh., Experience in Application of Voting Algorithms to Prediction of Double Chemical Compounds, *Trudy Mezhdunarodnogo simpoziuma po prakticheskim primeneniyam metodov raspoznavaniya obrazov* (Proc. Int. Symp. Applic. Pattern Recogn. Methods), Moscow: Computer Center, USSR Acad. Sci., 1973, pp. 231–236.
14. Gourevitch, I.B. and Zhuravlev, Yu.I., Minimization of Boolean Functions and Efficient Recognition Algorithms, *Kibernetika*, 1974, no. 3, pp. 16–20.
15. Dadashev, T.M., On Recognition Algorithms Extremal in Terms of Caution Thresholds, *Vych. Mat. Mat. Fiz.*, 1973, vol. 13, pp. 737–749.
16. Dadashev, T.M., On Logical Orbits of Recognition Systems, *Kibernetika*, 1974, no. 6, pp. 121–133.
17. Dadashev, T.M., Locally Extremal Recognition Algorithms Based on Estimates Calculation, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 132–153.
18. Dmitriev, A.N., Zhuravlev, Yu.I., and Krendelev, F.P., On Mathematical Principles for Classification of Objects and Phenomena, *Diskretnyi Analiz* (Translation), Novosibirsk: Inst. Mat., Siber. Div., USSR Acad. Sci., 1966, no. 7, pp. 3–11.
19. Dmitriev, A.N., Zhuravlev, Yu.I., and Krendelev, F.P., On a Principle for Classification and Forecasting of Geological Structures and Phenomena, *Izv. Sib. Otd. Akad. Nauk SSSR, Geologiya i Geofizika*, 1968, no. 5, pp. 50–64.
20. Doktorovich, A.B., The Problem of Choosing a Recognition Algorithm in a Class of Voting Algorithms, *Kibernetika*, 1974, no. 4, pp. 135–140.
21. Doktorovich, A.B., Study into the Efficiency Criterion of Estimates Calculation Algorithms, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, pp. 154–166.
22. Dyukova, E.V., On an Algorithm for Construction of Irredundant Tests, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 167–185.
23. Ench, V., An Algorithm for Parameter Optimization of a Class of Recognition Algorithms, *Trudy Mezhdunarodnogo simpoziuma po prakticheskim primeneniyam metodov raspoznavaniya obrazov* (Proc. Int,. Symp. Applic. Pattern Recogn. Methods), Moscow: Computer Center, USSR Acad. Sci., 1973, pp. 237–245.
24. Zhuravlev, Yu.I., Extremum Problems Arising in Substantiation of Heuristic Procedures, *Problemy prikladnoi matematiki i mekhaniki* (Topics in Applied Mathematics and Mechanics), Moscow: Nauka, 1971.
25. Zhuravlev, Yu.I., Estimates Calculation Recognition Algorithms, Informative Meaning of Algorithm-defining Parameters, *Trudy Mezhdunarodnogo simpoziuma po prakticheskim primeneniyam metodov raspoznavaniya obrazov* (Proc. Int. Symp. Applic. Pattern Recogn. Methods), Moscow: Computer Center, USSR Acad. Sci., 1973, pp. 205–218.
26. Zhuravlev, Yu.I. Extremum Algorithms in Mathematical Models for Recognition and Classification Problems, *Dokl. Akad. Nauk SSSR*, 1976, vol. 231, no. 3, pp. 532–535.
27. Zhuravlev, Yu.I. and Tulyaganov, Sh.E., Measuring the Importance of a Feature, *Voprosy Kibernetiki* (Translation), Tashkent: Inst. Cyb. Com. Center, Uzbek Acad. Sci., 1970, no. 38, pp. 29–42.
28. Zhuravlev, Yu.I. and Nikiforov, V.V., Recognition Algorithms Based on Estimates Calculation, *Kibernetika*, 1971, no. 3, pp. 1–11.
29. Zhuravlev, Yu.I. and Yunusov, R., On a Method to Refine the Taxonomy Algorithm by Voting-Type Recognition Methods, *Zh. Vych. Mat. Mat. Fiz.*, 1971, vol. 11, no. 5, pp. 1344–1347.\
30. Zhuravlev, Yu.I. and Tulyaganov, Sh.E., Importance Measures of Objects in Complex Systems, *Zh. Vych. Mat. Mat. Fiz.*, 1972, vol. 12, no. 1, pp. 170–184.
31. Zhuravlev, Yu.I., Kamilov, M.M., and Tulyaganov, Sh.E., Estimates Calculation Algorithms and Their Applications, Tashkent: Fan, 1974.
32. Zhuravlev, Yu.I. and Mihalewicz, M., Estimates Calculation Recognition Algorithms for Problems with Intersecting

Classes, *Proc. Comp. Center, Acad. Sci. of Poland*, 1974, no. 145.

33. Zhuravlev, Yu.I., Karczewski, E. and Mihalewicz, M., Recognition Algorithms for Problems Specified by Collections of Qualitative Features, *Proc. Comp. Center, Acad. Sci. of Poland*, 1974, no. 146.
34. Zhuravlev, Yu.I., Zenkin, A.I., Skorobogatov, I.B., and Uzakov, Ya.A., Experience in Application of Pattern Recognition Methods to Assessing the Business Qualities of an Employee, *Voprosy Radioelektroniki, Ser. Obshchetekh.*, 1974, no. 3, pp. 28–44.
35. Zhuravlev, Yu.I., Miroshnik, S.N., and Shvartin, S.M., On an Approach to Optimization in the Class of Parametric Recognition Algorithms, *Zh. Vych. Mat. Mat. Fiz.*, 1976, vol. 16, no. 1, pp. 209–218.
36. Zhuravlev, Yu.I., Zenkin, A.I., and Ryazanov, V.V., Algorithms to Forecast the Status of Industrial Processes, *Voprosy Radioelektroniki, Ser. ASU*, 1976, no. 1, pp. 32–42.
37. Ishchuk, V.I., On the Search of Optimal Weight Coefficients for a Class of Estimates Calculation Algorithms, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 186–194.
38. Kamilov, M.M., The PRASK-1 Recognition Software Package, *Voprosy Kibernetiki* (Translation), Tashkent: Inst., Cyb, & Comp. Center, Uzbek Acad., Sci., 1972, no. 51, pp. 63–65.
39. Kamilov, M.M., On Optimization and Applications of Estimation Algorithms, *Trudy Mezhdunarodnogo simpoziuma po prakticheskim primeneniyam metodov raspoznavaniya obrazov* (Proc. Int. Symp. Applic. Pattern Recogn. Methods), Moscow: Computer Center, USSR Acad. Sci., 1973, pp. 246–258.
40. Kamilov, M.M. and Aliev, E.M., Selecting the Length of Voting Collections in Estimates Calculation Algorithms, *Voprosy Kibernetiki* (Translation), Tashkent: Inst. Cyb. & Comp. Center, Uzbek Acad. Sci., 1971, no. 44, pp. 162–165.
41. Kamilov, M.M., Aliev, E.M., and Kim, A.N., On Computation of ε-Thresholds in Object Recognition by Voting Algorithms, *Voprosy Kibernetiki* (Translation), Tashkent: Inst. Cyb. & Com. Center, Uzbek Acad. Sci., 1971, no. 43.
42. Kamilov, M.M. and Kim, A.N., Recognition Software Package Based on Estimates Calculation Algorithms, *Voprosy Kibernetiki* (Translation), Tashkent: Inst. Cyb. & Com. Center, Uzbek Acad. Sci., 1972, no. 52, pp. 68–74.
43. Katerinochkina, N.N., Search for the Maximal Upper Zero of a Monotone Boolean Function, *Dokl. Akad. Nauk SSSR*, 1975, vol. 224, no. 3, pp. 557–560.
44. Kol'tsov, P.P., Recognition Problem under Time Constraints, *Zh. Vych. Mat. Mat. Fiz.*, 1974, vol. 14, no. 5, pp. 1292–1308.
45. Konstantinov, R.M. and Koroleva, Z.E., Application of Test Algorithms to Biological Prediction Problems, *Trudy Mezhdunarodnogo simpoziuma po prakticheskim primeneniyam metodov raspoznavaniya obrazov* (Proc. Int. Symp. Applic. Pattern Recogn. Methods), Moscow: Computer Center, USSR Acad. Sci., 1973, pp. 194–204.
46. Konstantinov, R.M., Koroleva, Z.E., and Kudryavtsev, V.B., A Combinatorial Logical Approach to Forecasting of Ore Occurrence, *Problemy Kibernetiki* (Translation), Moscow: Nauka, 1976, no. 31, pp. 5–33.
47. Koroleva, Z.E., On Comparison of Test Recognition Algorithms, *Zh. Vych. Mat. Mat. Fiz.*, 1975, vol. 15, no. 3, pp. 749–756.
48. Krendelev, F.P., Dmitriev, A.N., and Zhuravlev, Yu.I., Comparison of the Geological Structure of Foreign Precambrian Conglomerate Deposits by Discrete Mathematics, *Dokl. Akad. Nauk SSSR*, 1967, vol. 173, no. 5, pp. 1149–1152.
49. Kudryavtsev, V.B. and Kudryavtsev, V.B., On a Test Approach to the Community Prognosis Problem, in *Issledovanie operatsii* (Operations Research), Moscow: Computer Center, USSR Acad. Sci., 1972.
50. Kuznetsov, V.E., On a Stochastic Algorithm for Computation of Information Characteristics of Tables by the Test Method, *Diskretnyi Analiz* (Translation), Novosibirsk: Inst. Math. Siber. Div., USSR Acad. Sci., 1973, no. 23, pp. 8–23.
51. Kul'yanov, E.G., On Optimization of a Class of Recognition Algorithms, *Zh. Vych. Mat. Mat. Fiz.*, 1974, vol. 14, no. 3, pp. 756–767.
52. Kul'yanov, E.G., An Algorithm for the Search of an Upper Zero for an Arbitrary Monotone Function of Boolean Algebra, *Zh. Vych. Mat. Mat. Fiz.*, 1975, vol. 15, no. 4, pp. 1080–1082.
53. Leonovich, A.S., Construction of an Algorithm for Recognition of Spoilage Causes in Steel Making, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 212–222.
54. Madatyan, Kh.A., Construction of an Optimum Algorithm in a Class of Recognition Problems, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 223–232.
55. Mazurov, Vl.D., On the Committee of a Convex Inequality System, *Trudy ICM-1966* (Proc. ICM-1966), Moscow: Mosk. Gos. Univ., 1966.
56. Mazurov, Vl.D., Committees of the Inequality System and Recognition Problems, *Kibernetika*, 1971, no. 3.
57. Miroshnik, S.N., A Voting Algorithm with a Continuous Metric, *Kibernetika*, 1972, no. 2, pp. 54–63.
58. Noskov, V.N., On Irredundant and Minimal Tests for a Class of Tables, *Diskretnyi Analiz* (Translation), Novosibirsk: Inst. Math., Siber. Div., USSR Acad. Sci., 1968, no. 12, pp. 27–49.
59. Nurlybaev, A., On Normal Forms of k-valued Logic, *Sbornik rabot po matematicheskoi kibernetike* (Translation), Moscow: Computer Center, USSR Acad. Sci., 1976, no. 1, pp. 56–68.
60. Ryazanov, V.V., Optimization of Estimation Algorithms from Parameters Characterizing Representativity of Template Rows, *Zh. Vych. Mat. Mat. Fiz.*, 1976, vol. 16, no. 6, pp. 1556–1570.
61. *Sbornik: Metod komitetov v raspoznavanii obrazov* (Collected Volume on the Committee Method of Pattern Recognition), Sverdlovsk: Ural Sci. Center, USSR Acad. Sci., 1974.
62. Sebast'yan, G.S., *Protsessy prinyatiya reshenii pri raspoznavanii obrazov* (Decision Making Processes in Pattern Recognition), Moscow: Tekhnika, 1965.
63. Slepyan, V.A., Probabilistic Measures of the Irredundant Test Distribution, *Diskretnyi Analiz* (Translation),

Novosibirsk: Inst. Math., Siber. Div., USSR Acad. Sci., 1968, no. 12, pp. 50–74.

64. Slepyan, V.A., Irredundant Test Distribution Parameters and Information Weights of Columns in Binary Tables, *Diskretnyi Analiz* (Translation), Novosibirsk, Inst. Math., Siber. Div., USSR Acad. Sci., 1969, no. 14, pp. 28–43.
65. Slepyan, V.A., Length of a Minimal Test for a Class of Tables, *Diskretnyi Analiz* (Translation), Novosibirsk: Inst. Math., Siber. Div., USSR Acad. Sci., 1973, no. 23, pp. 59–71.
66. Slutskaya, T.L., An Algorithm to Compute Information Weights of Features, *Diskretnyi Analiz* (Translation), Novosibirsk: Inst. Math., Siber. Div., USSR Acad. Sci., 1968, no. 12, pp. 75–90.
67. Slutskaya, T.L., On Efficiency of Voting Algorithms for a Class of Binary Tables, *Kibernetika*, 1973, no. 2, pp. 80–86.
68. Terenkov, V.N., On Accuracy of Estimates Calculation Algorithms for Tables Generated by Monotone Boolean Functions, *Zh. Vych. Mat. Mat. Fiz.*, 1973, vol. 13, no. 6, pp. 1620–1625.
69. Terenkov, V.N., On Accuracy of Estimates Calculation Algorithms for a Class of Tables, *Kibernetika*, 1974, no. 3, pp. 27–31.
70. Uzakov, Ya.A., On a Definition of the Importance Measure of Features Involved in an Object Description, *Izv. Akad. Nauk Uzb.SSR, Ser. Tekh. Nauk*, 1976, no. 1, pp. 3–6.
71. Chegis, I.A. and Yablonskii, S.V., Logical Methods of Electric Circuit Testing, *Trudy Mat. Inst. im. V.A. Steklova Akad. Nauk SSSR*, 1958, vol. 51, pp. 270–360.
72. Shautsukova, L.Z., Assessment of Informative Content of Functional Signs and Their Collection in the Dynamics of Medical Treatment, *Zh. Vych. Mat. Mat. Fiz.*, 1975, vol. 15, no. 3, pp. 803–806.
73. Shautsukova, L.Z., Solution of a Classification Problem by the Method of Estimates Calculation, *Zh. Vych. Mat. Mat. Fiz.*, 1975, vol. 15, no. 4, pp. 1020–1030.
74. Yablonskii, S.V., Functional Construction in *k*-valued Logic, *Trudy Mat. Inst. im. V.A. Steklova Akad. Nauk SSSR*, 1958, vol. 51, pp. 5–142.
75. Yablonskii, S.V., Demidova, N.G., Konstantinov, R.M., Koroleva, Z.E., Kudryavtsev, V.B., and Sirotinskaya, S.V., Test Approach to Qualitative Assessment of Geological and Structural Factors and Scale of Ore Occurrence (with Template to Mercury Deposits), *Geologiya Rtutnykh Mestorozhdenii*, 1971, vol. 13, no. 2, pp. 30–42.
76. Ablow, C.M. and Kaylor, D.J., A Committee Solution of the Pattern Recognition Problem, *IEEE Trans.*, 1965, vol. IT-11, no. 3.
77. Grenander, U., Foundations of Pattern Analysis, *Quarterly of Applied Mathematics*, 1969, vol. 27, no. 1, pp. 1–55.
78. Fu, K.S., *Syntactic Approaches to Pattern Recognition*, New York: Academic, 1974.
79. Zadeh, L.A., Fuzzy Sets, *Inform. Contr.*, 1965, no. 8, pp. 338–353.
80. Zhuravlev, Yu.I., Algorithms for Assessing the Quality of Expert Data, *Int. Sem. Trends in Math. Modeling-80*, Berlin: Springer, 1973, pp. 238–242.
81. Zhuravlev, Yu.I., Formal Algebras on a Set of Informal (Heuristic) Algorithms: I, II, *Kibernetika*, 1977, nos. 4, 6.

Yurii I. Zhuravlev. Born in 1935. Graduated from the Department of Mechanics and Mathematics, Moscow State University, in 1957. Received a PhD (Cand. Sci.) in 1959 and a postdoctoral degree (Dr. Sci. degree) in physics and mathematics in 1965. Since 1969, with the Computer Center of the Russian Academy of Sciences, Moscow, first as Laboratory Head and then as Deputy Director. The Chairman of the Scientific Council "Cybernetics". Holds a Chair in the Department of Computational Mathematics and Cybernetics of Moscow State University. A full member of the Russian Academy of Sciences (elected in 1992) and of the Academy of Doctors of Spain (elected in 1993).

Scientific interests: mathematical logic; algebra; discrete optimization; pattern recognition; and the use of mathematical and computational methods in applied problems of data processing and computer-assisted research in industry, medicine, geology and sociology. Author of more than 170 papers on information technology and applied mathematics.

Editor-in-Chief of *Pattern Recognition and Image Analysis* and of the yearbook *Raspoznavanie, klassifikatsiya, prognoz: matematicheskie metody i ikh primenenie* (Recognition, Classification and Forecasting: Mathematical Methods and Their Application) and a member of the editorial boards of several international and Russian journals on information technology and applied mathematics.