Discrete (Logic) Recognition Procedures: Principles of Construction, Complexity of Realization, and Basic Models[1]

E. V. Djukova

Computer Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 119991 Russia
e-mail: djukova@ccas.ru

Abstract—Discrete, or logic, recognition procedures are described. For the basic models, new estimates of computational complexity are obtained. An efficient algorithm for finding irreducible coverings of a Boolean matrix (searching for maximal conjunctions of a monotone Boolean function specified by a conjunctive normal form) is constructed.

INTRODUCTION

Very generally, the recognition problem is as follows. Some set of objects M is considered. It is known that M can be represented as the union of l subsets $K_1, \ldots, K_l$, called classes. The objects from M are described by some system of features $\{x_1, \ldots, x_n\}$. There is a finite set $S_1, \ldots, S_m$ of objects from M such that it is known to which classes they belong. These are precedents, or training objects. Suppose that their descriptions are $S_1 = (a_{11}, \ldots, a_{1n})$, $S_2 = (a_{21}, \ldots, a_{2n})$, …, $S_m = (a_{m1}, \ldots, a_{mn})$; here, a_{ij} is the value of the feature x_j for the object S_i. Given a set of feature values which describes some object from M (generally, it is not known to which class it belongs), it is required to determine the class containing this object.

Usually, in real-life recognition problems, the initial descriptions of objects contain all characteristics or parameters that can be observed or measured. As a result, objects are described by dozens, or even hundreds, of variables. Such a situation is typical, in particular, of problems of medical diagnosis, geological, technical, and sociological prediction, etc. Initially, this direction was believed to be a part of mathematical statistics. The analysis of complex descriptions by statistical methods involved taking additional probability hypotheses for granted, i.e., imposing fairly strong requirements on the spaces of objects under examination. In addition, the obtaining of reliable results on the basis of the statistical approach required very large arrays of precedents, i.e., sufficiently representative training samples. It turned out that employing large sets of precedents usually involved expensive and time consuming processing and was sometimes entirely impossible as, for example, in predicting rare metals. There were no adequate mathematical methods for solving similar problems, and they had to be created on the basis of completely new ideas. One of the approaches, which is the subject matter of this paper, consists in a combinatorial analysis of feature descriptions of objects. Such an analysis is aimed at determining the most informative subdescriptions (or fragments of descriptions) of training objects. For instance, a subdescription is informative if it makes it possible to distinguish a given object from all objects not belonging to the same class as the object under consideration.

Searching for informative fragments is based on the apparatus of discrete mathematics, including Boolean algebra, the theory of DNFs, and the theory of coverings of Boolean and integer matrices. Pioneering work was done by Yu.I. Zhuravlev. In one of the first papers on this topic [2], the problem of predicting gold-bearing deposits was considered and a recognizing algorithm was constructed with the use of the notion of test, which is well known in discrete mathematics. This notion was introduced by S.V. Yablonskii, who applied it to the theory of control systems [14]. In control problems, a test is a set of tuples of values of variables that makes it possible to distinguish between proper and improper functioning of a system and find possible failures.

The mentioned paper by Dmitriev, Zhuravlev, and Krendelev and papers by Vaintsvaig and Bongard, where the Kora recognizing algorithm was described, have initiated extensive application of discrete analysis methods to problems of recognition, classification, and forecasting. A whole class of complex heuristics called discrete, or logic, recognition procedures emerged (these heuristics can be constructed with the use of the apparatus of logic functions, as we shall demonstrate later on).

Thus, the application of the apparatus and methods of discrete mathematics to recognition problems has a

[1] This work was supported by the Russian Foundation for Basic Research, project no. 01-01-00575, and by the Presidium of the Russian Academy of Sciences, program “Intelligent Computer Systems,” project no. 1.1.

Received February 26, 2003

number of advantages, the most important of which is the possibility of obtaining a result when there is no information about the distribution functions and when the training samples are small. However, the application of the discrete approach often involves purely computational difficulties related to searching, which arise at the stage of determining informative fragments of object descriptions. Because of the necessity of a large-scale exhaustive search and the initially low performance of computing facilities, the main efforts during many years were directed towards the development of a general complexity theory for discrete data analysis in problems of recognition and synthesis of asymptotically optimal algorithms [3–7].

In this paper, new estimates of computational complexity of discrete (logic) recognition procedures are obtained.

For simplicity, consider the case of binary data. Suppose that the objects are described by a system of features $\{x_1, \ldots, x_n\}$, where $x_j \in \{0, 1\}$ for $j = 1, 2, \ldots, n$, and let $f_K(x_1, \ldots, x_n)$ be a partial Boolean function taking the value 1 at the tuples that are descriptions of training objects from the class K and 0 at the tuples describing the remaining training objects. By an elementary classifier, we understand an arbitrary fragment of the description of a training object. The elementary classifiers determine special conjunction functions f_K, and each recognizing algorithm A is determined by some set of such conjunctions. In the most typical cases, e.g., in constructing algorithms of voting over representative sets (Kora-type algorithms), there arise problems of determining admissible and maximal conjunctions of the function f_K. Of most computational complexity is the search for maximal conjunctions.

In practice, constructions based on searching for irreducible coverings of Boolean matrices are used more frequently. As is known, the problem of searching for irreducible coverings can be stated as the problem of searching for maximal conjunctions of a monotone Boolean function specified by a conjunctive normal form (CNF). In this paper, we construct a computationally efficient algorithm for solving these problems (an algorithm with polynomial time delay). A shortcoming of the algorithm is that it involves repeated steps. We specify conditions under which this shortcoming does not substantially affect the speed of the algorithm. This algorithm is a modification of an algorithm suggested earlier by the author which was used to efficiently solve the same problem under the same conditions with an asymptotic accuracy.

1. GENERAL PRINCIPLES OF CONSTRUCTION OF DISCRETE RECOGNITION PROCEDURES

We assume that the initial descriptions of objects are given in the form of tuples of values of features with integer ranges (preferably, not too large). In addition, we assume that objects from different classes have different descriptions.

Let $H = \{x_{j_1}, \ldots, x_{j_r}\}$ be a set of r different features, and let $S = (a_1, \ldots, a_n)$ be an object from M; here, a_j is the value of the feature x_j for $j = 1, 2, \ldots, n$.

The set of features H determines a fragment $(a_{j_i},$ $\ldots, a_{j_r})$ in the description of the object S. If $S \in \{S_1, \ldots,$ $S_m\}$, we call the fragment $(a_{j_1}, \ldots, a_{j_r})$ an *elementary classifier* and denote it by (S, H).

We use the following notation: N is the set of all elementary classifiers and $N(K)$, where $K \in \{K_1, \ldots, K_l\}$, is the set of all elementary classifiers from N generated by training objects from the class K.

Suppose that two objects $S' = (a'_1, \ldots, a'_n)$ and $S'' =$ $(a''_1, \ldots, a''_n)$ from M are given. We estimate the closeness of the objects S' and S'' with respect to a feature set $H = \{x_{j_1}, \ldots, x_{j_r}\}$ by the value

$$B(S', S'', H) = \begin{cases} 1 & \text{if } a'_{j_t} = a''_{j_t} \text{ for } t = 1, 2, \ldots, r \\ 0 & \text{otherwise.} \end{cases}$$

Thus, the objects S' and S'' are close to each other with respect to the feature set H (i.e., $B(S', S'') = 1$) if and only if the elementary classifiers (S', H) and (S'', H) coincide.

Let us describe the general scheme of the recognition algorithm A.

At the first stage (training), for each class K, a set of elementary classifiers with given properties, i.e., a subset $N_A(K)$ of $N(K)$, is constructed. An elementary classifier is considered informative if and only if it belongs to one of the sets $N_A(K_1), \ldots, N_A(K_l)$.

One of the most typical examples is the models of algorithms of voting over representative samples (Kora-type algorithms).

Definition. An elementary classifier (S, H) from $N(K)$ is called a *representative sample* (also referred to as a representative descriptor or representative set) for K if $B(S, S', H) = 0$ for any training object $S' \notin K$ (i.e., the elementary classifiers (S, H) and (S', H) do not coincide).

Thus, a representative sample for K is a fragment of the description of some training object from K which makes it possible to distinguish this object from any other training object not included in K.

In models with representative samples, the set $N_A(K)$ is formed by representative samples for K. Short representative samples are considered more informative. For this reason, in such models, only representative samples whose lengths do not exceed a certain preset threshold, rather than all of them, are usually constructed.

Suppose that $H = \{x_{j_1}, \ldots, x_{j_r}\}$ and $H^{(t)} = \{x_{j_1}, \ldots, x_{j_{t-1}}, x_{j_{t+1}}, \ldots, x_{j_r}\}$ for $t \in \{1, 2, \ldots, r\}$.

Definition. A representative sample (S, H) for K is called *irredundant* if, for any $t \in \{1, 2, \ldots, r\}$, the elementary classifier $(S, H^{(t)})$ is not a representative sample for K.

For a representative sample, the property of being irredundant means that this sample is irreducible; i.e., it loses the ability to distinguish the object that has generated it from some objects from other classes under contraction. The irredundant representative samples can be taken as $N_A(K)$.

An algorithm with representative samples was described for the first time in [1].

Definition. A set of features H is called a *test* if, for any $K \in \{K_1, \ldots, K_l\}$ and any $S \in K$, the elementary classifier (S, H) is a representative sample for K.

Obviously, a set of features $\{x_{j_1}, \ldots, x_{j_r}\}$ is a test if, for any two training objects S_i and S_p from different classes, we can find a $t \in \{1, 2, \ldots, r\}$ such that $B(S_i, S_p, \{x_t\}) = 0$. In other words, a test is a set of features making it possible to correctly classify the training material.

Definition. A test is called *irredundant* if none of its proper subsets is a test.

As $N_A(K)$, we can take the subset of $N(K)$ generated by tests. As in the case of representative samples, only tests of bounded length or only irredundant tests should be taken.

The first model of a test algorithm is described in [2].

Thus, at the first (training) stage, we construct fragments of descriptions of training objects that are informative for the algorithm under consideration; i.e., we define the set $N_A = N_A(K_1) \cup \ldots \cup N_A(K_l)$. After N_A is defined, a voting procedure is implemented. In the simplest modification of the algorithm, it is assumed that an elementary classifier (S', H) from $N_A(K)$ votes for the membership of an object S in the class K if $B(S', S, H) = 1$, i.e., if the corresponding fragments of the descriptions of the objects S and S' coincide.

In the simplest modification, the number $\Gamma(S, K_j)$ of "votes" cast by pairs from $N_A(K_j)$ for the membership of the object S in the class K_j is evaluated by the formula

$$\Gamma(S, K_j) = \frac{1}{m_j} \sum_{(S', H) \in N_A(K_j)} B(S, S', H),$$

where m_j is the number of training objects from the class K_j; $\Gamma(S, K_j)$ serves as a criterion for the membership of S in the class K_j.

The estimates $\Gamma(S, K_1), \ldots, \Gamma(S, K_l)$ are calculated. If

$$\Gamma(S, K_t) = \max_{1 \le j \le l} \Gamma(S, K_j)$$

and $\Gamma(S, K_t) \ne \Gamma(S, K_u)$ for $u \ne t$, then the algorithm A assigns the object S to the class K_t. If

$$\Gamma(S, K_t) = \Gamma(S, K_u) = \max_{1 \le j \le l} \Gamma(S, K_j),$$

then the algorithm A refuses to recognize the object S.

The quality of a recognizing algorithm A is often evaluated with the use of a cross-validation test, which is as follows. For each i from $\{1, 2, \ldots, m\}$, the estimates $\Gamma(S_i, K_j)$, where $j = 1, 2, \ldots, l$, are evaluated on the training subsample $\{S_1, \ldots, S_m\} \backslash \{S_i\}$. Let q be the number of correctly recognized objects among all S_i with $i = 1, 2, \ldots, m$. Then, the quality of the algorithm A is estimated by the functional $\varphi_{st}(A) = q/m$.

For large-scale problems, the described procedure requires substantial computational efforts. In the case where A is an algorithm of voting over representative samples, there is a method for fast computation of the estimates $\Gamma(S_i, K_j)$ ($i = 1, 2, \ldots, m, j = 1, 2, \ldots, l$) in the cross-validation procedure [17]. This method makes it possible to reduce computation time by approximately m times.

The quality of an algorithm A can also be evaluated by an independent test with the use of a set of t objects such that they are not included in the training sample but it is known to which classes they belong. Such a set of objects is called a test sample. If q is the number of correctly recognized objects from a test sample, then the quality of the algorithm is estimated by the functional $\varphi_{\text{test}}(A) = q/(m + t)$.

In [9, 11, 17], theoretical and experimental results concerning the construction of effective (in the sense of quality) recognition algorithms based on the notion of a representative sample are given.

Models used in practice are more complex: they involve additional parameters characterizing the representativeness (typicality) of training objects and their subdescriptions with respect to the classes containing them and the informativeness of features.

In models with representative samples, most "weighty" representative samples are selected; these may be samples of values taken by given features for a sufficiently large number of objects from the class under consideration [6, 9, 11, 16]. Models with almost representative samples are also used. In these models, a classifier from $N(K)$ is considered informative for the class K if, for a given set of features, it is encountered sufficiently often in descriptions of objects from this class and sufficiently rarely in descriptions of objects from other classes.

In test models, speed is often increased by applying stochastic algorithms, where constructing the set of all irredundant tests is replaced by constructing a sufficiently representative random sample of this set. In this case, the estimates $\Gamma(S, K_j)$, where $j \in \{1, 2, \ldots, l\}$, are calculated approximately and the error possibly involved in such an approximation is estimated [4, 12].

In [8, 9, 17], a number of new discrete heuristic procedures are suggested; they assume a sample of admissible feature values to be informative for a class K if it is missing from the descriptions of all training objects from the class K.

The models under consideration can be modified for the case of real-valued data [13]. For each feature x_j, where $j \in \{1, 2, \dots, n\}$, a real-valued parameter E_j characterizing the precision of the measurement of the feature x_j is specified. In the simplest modification, the value $B(S', BS'', H)$ is defined as follows. For $S' = (a'_1, \dots, a'_n)$, $S'' = (a''_1, \dots, a''_n)$, and $H = \{x_{j_1}, \dots, x_{j_r}\}$, we set

$$B(S', S'', H)$$
$$= \begin{cases} 1 & \text{if } \left|a'_{j_t} - a''_{j_t}\right| \le E_{j_t} \text{ for } t = 1, 2, \dots, r \\ 0 & \text{otherwise.} \end{cases}$$

Another approach to processing real-valued data (suggested by Zhuravlev) is to recode the initial information into integer-valued data from the very beginning (models using this approach are called models with recoding).

Remark. The described scheme for constructing a recognition algorithm gives a general impression of the structure of estimation-type models (or algorithms of voting over support sets) [10]. Historically, the first models of this type were test models (to be more precise, algorithms of voting over sets of irredundant tests). Test algorithms required too much computations. A theoretical examination of the statistical properties of irredundant tests showed that irredundant tests are almost always of approximately equal lengths. Generally, this length depends on the relation between the parameters m and n. For instance, if the data are binary ($l = 2$) and $(m_1m_2)^\alpha \le n$, where $\alpha > 1$, $n \longrightarrow \infty$, then almost all irredundant tests have lengths of order $\log(m_1m_2n)$ for almost all tables. This observation has served as one of the substantiations for constructing estimation algorithms for which the role of $N_A(K)$ is played by the set of all possible elementary classifiers (S, H) from $N(K)$, where H has fixed cardinality r. The value of r can be determined from a preliminary analysis of the training sample. The model is largely aimed at processing real-valued data. For this reason, parameters E_j and P_j, where $j \in \{1, 2, \dots, n\}$, characterizing, respectively, the precision of measurement and the informativeness of each feature x_j and parameters $\gamma(S')$, where $S' \in \{S_1, \dots, S_m\}$, characterizing the representativeness (typicality) of training objects with respect to their classes, are specified. For this model, there is an effective formula for evaluating the estimates $\Gamma(S, K_j)$ ($j \in \{1, 2, \dots, l\}$) [10, 13].

2. CONSTRUCTION OF ELEMENTARY CLASSIFIERS ON THE BASIS OF A TRANSFORMATION OF NORMAL FORMS OF LOGIC FUNCTIONS

Construction of discrete recognition procedures often uses the apparatus of logic functions.

All notions that are used but not defined below can be found in [15].

Let $f_K(x_1, \dots, x_n)$ be a partial (not everywhere defined) two-valued function which assigns the value 1 to the n-tuples that are descriptions of objects from the class K and 0 to the n-tuples describing the remaining training objects; i.e., f_K is the characteristic function of the class K. Let us show that the elementary classifiers generated by pairs from $N(K)$ correspond to special conjunctions of the function f_K and, therefore, each recognizing algorithm corresponds to some set of such conjunctions (each algorithm is based on constructing a set of such conjunctions). For simplicity, we consider the case where the objects are described by binary features.

Let E^n be a family of n-tuples of the form $(\alpha_1, \dots, \alpha_n)$, where $\alpha_i \in \{0, 1\}$ for $i = 1, 2, \dots, n$, and let A_K and B_K be the families of n-tuples from E_n at which the function f_K takes the values 1 and 0, respectively. Suppose that B is an elementary conjunction (EC) over the variables $x_1, \dots, x_n$ and N_B is the truth domain of the EC B.

Let us introduce some definitions.

Definition. EC B is called *almost admissible* for f_K if $N_B \cap A_K \neq \varnothing$.

The following statement is obvious.

Statement 1. EC $x_{j_1}^{\sigma_1} \dots x_{j_r}^{\sigma_r}$ is almost admissible for f_K if and only if $(\sigma_1, \dots, \sigma_r)$ is an elementary classifier for K generated by the set of features $\{x_{j_1}, \dots, x_{j_r}\}$.

Thus, each almost admissible conjunction B of the function f_K corresponds to a set of elementary classifiers from $N(K)$. Their number thereof equals $|N_B \cap A_K|$.

Definition. EC B is called *admissible* for f_K if $N_B \cap A_K \neq \varnothing$ and $N_B \cap B_K = \varnothing$.

The following statement is obvious.

Statement 2. EC $x_{j_1}^{\sigma_1}, \dots, x_{j_r}^{\sigma_r}$ is admissible for f_K if and only if $(\sigma_1, \dots, \sigma_r)$ is a representative sample for K generated by the set of features $\{x_{j_1}, \dots, x_{j_r}\}$.

Definition. EC B is called *irreducible* for f_K if there exists no EC B' such that $N_{B'} \subset N_B$ and $N_{B'} \cap B_K = N_B \cap B_K$.

EC B is irreducible if it is irredundant in a certain sense. Under the contraction of such a conjunction, the intersection of its truth domain N_B with the zero set of the function f_K increases. In particular, if B is an irreducible admissible conjunction, then it ceases to be admissible.

Definition. EC B is called *maximal* for f_K if it is admissible and there exists no admissible conjunction B' such that $N_{B'} \supset N_B$.

The definitions given above imply that an EC is maximal for f_K if and only if it is admissible and irreducible.

The following two statements are obvious.

Statement 3. A set of features $\{x_{j_1}, \dots, x_{j_r}\}$ generates an irredundant representative sample for K of form $(\sigma_1, \dots, \sigma_r)$ if and only if the conjunction $x_{j_1}^{\sigma_1} \dots x_{j_r}^{\sigma_r}$ is maximal for f_K.

Statement 4. A set of features $\{x_{j_1}, \dots, x_{j_r}\}$ is a test if and only if, for each function f_{K_t} with $t \in \{1, 2, \dots, l\}$, all almost admissible conjunctions of form $x_{j_1}^{\sigma_1}, \dots, x_{j_r}^{\sigma_r}$ are admissible.

Remark. The definitions of almost admissible, admissible, irreducible, and maximal conjunctions of a partial Boolean function can be completely transferred to the case where the Boolean function f_K is everywhere defined, i.e., $A_K = E^n \backslash B_K$.

Thus, constructing recognition algorithms with the use of the apparatus of logic functions involves constructing almost admissible, admissible, and maximal conjunctions of partial Boolean functions. Most difficult is the search for maximal conjunctions. One of the best known methods for constructing maximal conjunctions of partial Boolean functions is as follows.

Let us consider an everywhere defined Boolean function $F_K(x_1, \dots, x_n)$ which coincides with f_K on the set of 0 and 1 values; at the remaining n-tuples from E^n, $F_K(x_1, \dots, x_n)$ equals 1. The specification of the zero set of an everywhere defined Boolean function is equivalent to the specification of the perfect CNF of this function. Let B_K consist of n-tuples $(\beta_{11}, \dots, \beta_{1n})$, $(\beta_{21}, \dots, \beta_{2n})$, $\dots$, $(\beta_{u1}, \dots, \beta_{un})$. Then, obviously, the CNF of the function F_K is $D_1 \,\&\, \dots \,\&\, D_u$, where

$$D_i = x_1^{\bar{\beta}_{i1}} \vee \dots \vee x_n^{\bar{\beta}_{in}}, \quad i = 1, 2, \dots, u.$$

Statements 5 and 6, stated below, are easy to prove.

Statement 5. EC B is admissible for F_K if and only if each disjunction D_i, where $i \in \{1, 2, \dots, u\}$, contains at least one factor from B.

Statement 6. EC B of rank r is irreducible for F_K if and only if the CNF $D_1 \,\&\, \dots \,\&\, D_u$ contains r disjunctions $D_{i_1}, \dots, D_{i_r}$ such that each of them has precisely one factor from B and, if $r > 1$, $p, q \in \{i_1, \dots, i_r\}$, and $p \neq q$, then the disjunctions D_p and D_q contain different factors from B.

Statement 5 and 6 are valid for arbitrary (not necessarily perfect) CNFs.

Statement 5 implies that, by multiplying the logic parentheses and simplifying the resulting DNF with the use of the identities $x\bar{x} = 0$, $xx = x$, and $x \vee x = x$, we obtain a DNF consisting of all admissible conjunctions of the function F_K. Let us now remove the admissible conjunctions not being irreducible by using the identity $x \vee xx' = x$. We obtain a DNF consisting of all maximal conjunctions of the function F_K (or the reduced DNF). The set of maximal conjunctions for f_K is obtained by selecting the conjunctions admissible for f_K in the set of all maximal conjunctions for F_K.

Of most interest in the discrete approach is the case where the number of features is substantially larger than the number of objects. In this case, the number of parentheses is few in comparison with the number of variables. It can be shown that, almost always (i.e., for almost all CNFs of the form under consideration), the number of admissible conjunctions is an order larger than the number of maximal conjunctions when $n \longrightarrow \infty$. Therefore, the algorithm for constructing maximal conjunctions described above is not efficient. Theoretical and experimental studies show that, in the case under consideration, it is more expedient to start with constructing irreducible conjunctions of the function F_K and, then, verify the admissibility of each of them. In [5, 6], it is proved that, if $u \le n^{1-\varepsilon}$, where $\varepsilon > 0$, then the number of irreducible conjunctions almost always (for almost all CNFs of the form under consideration) asymptotically coincides with the number of maximal conjunctions of the function F_K as $n \longrightarrow \infty$. On the basis of this observation, in the same papers, an algorithm for searching for maximal conjunctions of the function F_K, which is asymptotically optimal in a certain sense with respect to computational complexity, was constructed. The initial problem of constructing all maximal conjunctions of the function F_K was replaced by the simpler problem of constructing all irreducible conjunctions of the function F_K; i.e., the problem was solved approximately. The complexity of the approximated solution was estimated by the number of conjunctive multiplications. For the case where $u \le n^{1-\varepsilon}$ with $\varepsilon > 0$, an algorithm for searching for all irreducible conjunctions of the function F_K was suggested; the number of conjunctive multiplications in this algorithm almost always asymptotically coincides as $n \longrightarrow \infty$ with the number of maximal conjunctions of the function F_K. In this algorithm, one conjunctive multiplication requires searching through no more than $O(un)$ variables in the given CNF.

Thus, it was shown that, if the number u of zeros of the function F_K is sufficiently small in comparison with the number n of variables, then it is almost always possible to construct a DNF that contains all maximal conjunctions of the function F_K and is of almost the same complexity (length) as the required DNF by performing a minimum (in a certain sense) number of “&” operations.

In [5–7], it was shown that these results were also valid for the more general case of an arbitrary (not necessarily perfect) initial CNF specifying a two-valued function F_K defined on the set of k-ary n-tuples, where $k \geq 2$. In [5–7], the case where the initial CNF realized a monotone Boolean function, i.e., contained no negations of variables, was distinguished (this case is of most practical importance).

3. CONSTRUCTION OF ELEMENTARY CLASSIFIERS ON THE BASIS OF A SEARCH FOR COVERINGS OF BOOLEAN MATRICES

Implementations of algorithms for searching for (irredundant) representative sets and (irredundant) tests often use constructions based on a search for irreducible coverings of Boolean matrices.

Let L be a Boolean matrix.

Definition. A set H of columns of the matrix L is called a *covering* if, for each row of the matrix L, there is at least one column from H such that the element in their intersection is 1.

Definition. A covering is called *irreducible* (*irredundant*) if none of its proper subsets is a covering.

Irreducible coverings are usually constructed with the use of the following criterion. A set H of columns of the matrix L is an irreducible covering if and only if (i) the submatrix L^H of L formed by the columns from the set H contains no rows of the form (0, 0, …, 0) and (ii) L^H contains all rows (1, 0, …, 0), (0, 1, …, 0), …, (0, 0, …, 1); i.e., L^H contains the identity submatrix.

How does the problem of constructing coverings and irreducible coverings arise in constructing (irredundant) tests and (irredundant) representative sets? To find the required set of elementary classifiers, a special Boolean matrix is constructed. Let us denote it by L^*. Each row of this matrix is formed as a result of comparing a pair of objects from different classes. The jth element of the row is 1 if the descriptions of the objects being compared differ in the jth feature and 0 otherwise. Thus, if we compare objects S_i and S_u and have $B(S_i, S_u, \{x_j\}) = 1$, then we put 0 in the row of the matrix L^* corresponding to the pair (S_i, S_u); otherwise, we put 1. Let L_i^* denote the submatrix of L^* formed by comparing a training object S_i with all training objects not belonging to the same class as S_i.

The following two statements are obvious.

Statement 7. The set of the features with numbers $j_1, \ldots, j_r$ is an (irredundant) test if and only if the set of the columns with numbers $j_1, \ldots, j_r$ in the matrix L^* is an (irreducible) covering.

Statement 8. An elementary classifier $(S_i, \{x_{j_1}, \ldots, x_{j_r}\})$ from $N(K)$ is an (irredundant) representative set for K if and only if the set of the columns with numbers $j_1, \ldots, j_r$ in the matrix L_i^* is an (irreducible) covering.

Thus, in constructing the basic models of discrete recognition procedures, searching for informative elementary classifiers is reduced to constructing coverings of Boolean matrices.

The problem of constructing the set of all irreducible coverings of a Boolean $u \times m$ matrix L can be stated as the problem of transforming a CNF of a monotone Boolean function into a reduced DNF [15]. Indeed, to the ith row, we assign the disjunction $D_i = x_{p_1} \vee \ldots \vee x_{p_q}$, where $p_1, \ldots, p_q$ are the positions occupied by 1 in the ith row. Let f_L be the monotone Boolean function realized by the CNF $D_1 \,\&\, \ldots \,\&\, D_u$.

Using Statement 5, it is easy to prove the following statement.

Statement 9. EC $x_{j_1}, \ldots, x_{j_r}$ is admissible for f_L if and only if the set H of the columns with numbers $j_1, \ldots, j_r$ in the matrix L is a covering.

Using Statement 6, it is easy to prove the following statement.

Statement 10. EC $x_{j_1}, \ldots, x_{j_r}$ is irreducible for f_L if and only if the set of the columns with numbers $j_1, \ldots, j_r$ in the matrix L contains the identity submatrix.

The following statement is implied by Statements 9 and 10.

Statement 11. EC $x_{j_1}, \ldots, x_{j_r}$ is maximal for f_L if and only if the set H of the columns with numbers $j_1, \ldots, j_r$ in the matrix L is an irreducible covering.

Statements 9–11 imply that algorithms for constructing maximal conjunctions of monotone Boolean functions can easily be modified for constructing irreducible coverings of Boolean matrices, and *vice versa*.

4. AN EFFICIENT ALGORITHM FOR CONSTRUCTING IRREDUCIBLE COVERINGS OF A BOOLEAN MATRIX

Let $L = (a_{ij})$, where $i = 1, 2, \ldots, u$ and $j = 1, 2, \ldots, n$, be a Boolean matrix. We assume that L contains no rows of form (0, …, 0). Let us denote the set of all irreducible coverings of the matrix L by $P(L)$.

We say that the ith row of the matrix L encloses the pth row if $a_{ij} \geq a_{pj}$ for $j = 1, 2, \ldots, n$.

Let L' denote the submatrix of L obtained by removing all enclosing rows from L. The following statement is obvious.

Statement 12. The set of the columns with numbers $j_1, \ldots, j_r$ in the matrix L is an (irreducible) covering if and only if the set of the columns with numbers $j_1, \ldots, j_r$ in the matrix L' is an (irreducible) covering.

Suppose that $i \in \{1, 2, \ldots, u\}$ and $j \in \{1, 2, \ldots, n\}$. If $a_{ij} = 1$, then we say that the ith row of the matrix L covers the jth column and, *vice versa*, the jth column covers the ith row. By $L(a_{ij})$, we denote the submatrix of L which remains after removing the columns covered by the ith row and the rows covered by the jth column from L. We assume that L has rows not covered by the jth column and columns not covered by the ith row; i.e., $L(a_{ij})$ is not empty. The following statement is obvious.

Statement 13. The submatrix $L(a_{ij})$ contains no rows of form $(0, \dots, 0)$ if and only if the ith row of the matrix L encloses none of the other rows of this matrix.

In what follows, we assume that L contains no enclosing rows.

Definition. Two different elements $a_{i_1j_1} = 1$ and $a_{i_2j_2} = 1$ of the matrix L are called consistent if $a_{i_1j_2} = a_{i_2j_1} = 0$. A set Q of r elements equal to 1 in the matrix E is called consistent if (i) $r = 1$ or (ii) $r > 1$ and any two elements in Q are consistent.

Let $S(L)$ denote the set of all consistent sets of unit elements of the matrix L. Each set from $S(L)$ determines an identity submatrix of the matrix L.

To each element a_{ij} in L, we assign the number $N[i, j] = (j - 1)u + i$; hereafter, we assume that, if $Q = \{a_{i_1j_1}, \dots, a_{i_rj_r}\} \in S(L)$, then $N[i_{t+1}, j_{t+1}] > N[i_t, j_t]$ for $t = 1, 2, \dots, r - 1$.

Suppose that $Q = \{a_{i_1j_1}, \dots, a_{i_rj_r}\} \in S(L)$. Let $L_0(Q)$ denote the submatrix of matrix L formed by the columns with numbers no smaller than j_1. By $L_t(Q)$, where $t \in \{1, 2, \dots, r\}$, we denote the submatrix of L formed by the rows not covered by the columns with numbers $j_1, \dots, j_t$ and by the columns with numbers larger than jt that are not covered by the rows with the numbers $i_1, \dots, i_t$.

In each of the submatrices $L_t(Q)$, where $t \in \{1, 2, \dots, r\}$, we retain the initial enumeration of rows and columns.

Definition. A set Q is called regular if, for any $t \in \{0, 1, \dots, r - 1\}$, the i_{t+1}th row of the submatrix $L_t(Q)$ encloses no other rows from $L_t(Q)$.

Denote the family of all regular sets in $S(L)$ by $\tilde{S}(L)$.

Let $Q = \{a_{i_1j_1}, \dots, a_{i_rj_r}\} \in \tilde{S}(L)$ and $t \in \{0, 1, \dots, r\}$. Removing all enclosing rows from the submatrix $L_t(Q)$, we obtain a submatrix $L'_t(Q)$, in which we also retain the initial enumeration of rows and columns. Let $G_t(Q)$ denote the set of all unit elements of the matrix L contained in $L'_t(Q)$. By construction, if $a_{ij} \in G_0(Q)$, then $\{a_{ij}\} \in \tilde{S}(L)$ and, if $a_{ij} \in G_t(Q)$, where $t \in \{1, 2, \dots, r\}$, then $\{a_{i_1j_1}, \dots, a_{i_tj_t}, a_{ij}\} \in \tilde{S}(L)$. We set $G(Q) = G_r(Q)$.

Let $\pi_t(Q)$ be the set of the numbers of rows that are enclosing in $L_t(Q)$, and let

$$\pi(Q) = \bigcup_{t=0}^{r} \pi_t(Q).$$

Definition. A regular set Q is called maximal if $G(Q) = \varnothing$.

Let us denote the family of all maximal sets in $S(L)$ by $S^*(L)$.

Using Statement 13, it is easy to prove the following statement.

Statement 14. The set H of the columns of the matrix L with numbers $j_1, \dots, j_r$ belongs to $P(L)$ if and only if $S^*(L)$ contains a set of the form $\{a_{i_1j_1}, \dots, a_{i_rj_r}\}$, where $i_j \in \{1, 2, \dots, u\}$ for $j = 1, 2, \dots, r$.

Statement 14 implies that $P(L)$ can be constructed by constructing sets from $S^*(L)$ (it should be taken into account that one set from $P(L)$ may correspond to several sets from $S^*(L)$).

Definition. A set $\{a_{i_1j_1}, \dots, a_{i_rj_r}\}$ belonging to $S^*(L)$ is said to be top if $\{a_{i_1j_1}, \dots, a_{i_rj_r}\} \in S^*(L)$ implies $i_t \le p_t$ for $t = 1, 2, \dots, r$.

We can determine whether a set Q is top by searching through the rows of the submatrix of L formed by the columns containing elements of Q and by the rows with numbers not belonging to $\pi(Q)$.

Let $R(L)$ be the set of all unit elements in the matrix L. By $e(R)$, we denote the element with minimum number in an arbitrary set $R \subseteq R(L)$.

Definition. A maximal set Q of form $\{a_{p_1j_1}, \dots, a_{p_rj_r}\}$ is called the initial set with first element $a_{i_1j_1}$ if $a_{i_{t+1}j_{t+1}} = e(G_t(Q))$ for any $t \in \{1, 2, \dots, r - 1\}$.

If the first element of an initial set Q is given, then the procedure for constructing Q is obvious. It is easy to see that the initial set with first element $e(R)$, where $R = R(L)$, is top.

Let us order $S^*(L)$ by specifying the immediate successor $\succ Q$ for each element Q from $S^*(L)$. Let $Q = \{a_{i_1j_1}, \dots, a_{i_rj_r}\}$. For $t \in \{1, 2, \dots, r\}$, we denote the set of unit elements in L with numbers larger than $N[i_t, j_t]$ by R_t. Below, we consider all possible cases and specify $\succ Q$ in each of them.

(i) If $r = 1$, then $\succ Q$ is the initial set from $S^*(L)$ with first element $e(R_1)$.

(ii) If $r > 1$ and $G_{r-1}(Q) \cap R_r \ne \varnothing$, then $\succ Q = (Q \backslash \{a_{i_rj_r}\}) \cup Q'$, where Q' is the initial set with first element $e(G_{r-1}(Q) \cap R_r)$.

(iii) If $r > 1$ and $G_{r-1}(Q) \cap R_r = \varnothing$, then $\succ Q$ is the initial set with first element $e(G_0(Q) \cap R_1)$ at $r = 2$ and $\succ Q = (Q \backslash \{a_{i_rj_r}, a_{i_{r-1}j_{r-1}}\}) \cup Q'$, where Q' is the initial set with first element $e(G_{r-2}(Q) \cap R_{r-1})$, at $r > 2$.

Note that $G_{r-2}(Q) \cap R_r = \varnothing$ at $r > 2$, because $a_{i_rj_r} \in G_{r-2}(Q) \cap R_{r-1}$. Note also that, if $G_{r-1}(Q) \cap R_r \ne \varnothing$ and the element $e = e(G_{r-1}(Q) \cap R_r)$ belongs to the j_rth column, then the set $\succ Q$ determines the same irreducible covering as Q. For this reason, to reduce the search, it is expedient to take the element with the least number in $G_{r-1}(Q)$, such that it belongs to a column with a number larger than j_r, instead of e.

The algorithm constructs $P(L)$ in $|S^*(L)|$ steps. At the ith step, the algorithm chooses a maximal set $Q[i, L\}$ in L. To eliminate repetitions, at $i \geq 2$, the condition that $Q[i, L]$ is a top set is verified. If this condition holds, then the set of columns containing elements of the set $Q[i, L]$ is included in the sought set of irreducible coverings. Otherwise, the set of irreducible coverings constructed at the preceding steps is not supplemented.

The maximal sets are chosen by the following rules:

(i) $Q[1, L\}$ is the initial set with first element $e(R(L))$;

(ii) if $Q[i, L]$ has an immediate successor in $S^*(L)$, then $Q[i + 1, L] => Q[i, L]$;

(iii) if $Q[i, L]$ does not have an immediate successor in $S^*(L)$, then the algorithm terminates.

It is seen from the description of the algorithm that it is based on a branching process, which can be conveniently represented as a tree.

A decision tree T_L whose vertices correspond to the elements of $\tilde{S}(L)$ is constructed. The only exception is the root vertex, which remains free. The suspended vertices correspond to maximal sets. In constructing the tree T_L, rules are applied that make it possible to proceed from one vertex to another in the order corresponding to a successive search through the branches of the tree. The transition from a vertex to the next one is polynomial with respect to the size of the matrix L (it requires searching through no more that $O(u^2n)$ elements of the matrix L). The transition from one suspended vertex to the next suspended vertex is also polynomial (it requires that no more than $O(u^3n)$ elements of the matrix L be searched through). The set $Q[1, L]$ is constructed in a time not exceeding $O(u^3n)$. Thus, in the worst case, the computational complexity of the algorithm equals $O(u^3n)|S^*(L)|$ and the time delay does not exceed $O(u^3n)$.

The total number of vertices in T_L equals $|\tilde{S}(L)| + 1$, and the number of suspended vertices equals $|S^*(L)|$. As is shown in [3–7], if $u \leq n^{1-\varepsilon}$ ($\varepsilon > 0$), then the value $|S(L)|$ almost always (for almost all $u \times n$ matrices L) asymptotically coincides with $|P(L)|$ as $n \longrightarrow \infty$. This implies that, typically, the number of suspended vertices asymptotically coincides with the number of irreducible coverings of the matrix L and, according to Statement 11, with the number of maximal conjunctions of the function f_L.

Remark. A modification of the asymptotically efficient algorithm for constructing maximal conjunctions of a monotone Boolean function specified by its CNF, which was mentioned in Section 3, gives an algorithm for constructing irreducible coverings of a Boolean matrix L based on searching for all identity submatrices of L. This algorithm was described for the first time in [3]; its computational complexity does not exceed $O(un)|S(L)|$ and, at $u \leq n^{1-\varepsilon}$ ($\varepsilon > 0$), almost never exceeds $O(un)|P(L)|$ as $n \longrightarrow \infty$.

CONCLUSIONS

This paper describes general principles of the discrete approach to recognition where the central problem is searching for informative fragments of feature descriptions of objects. To search for informative fragments, the apparatus of logic functions is applied, which includes methods of transformation of normal forms of Boolean functions and the theory of coverings of Boolean and integer matrices. Basic models of discrete or logic recognition procedures are considered, and new estimates of their computational complexities are obtained. Namely, an algorithm with a polynomial delay for finding irreducible coverings of a Boolean matrix (for searching for maximal conjunctions of a monotone Boolean function specified by its conjunctive normal form) is constructed. A shortcoming of the algorithm is that it involves repeated construction of irreducible coverings (maximal conjunctions). Conditions under which this shortcoming does not substantially affect the speed of the algorithm in typical situations are specified.

An abridged version of this paper was published in [16].

REFERENCES

1. Baskakova, L.V. and Zhuravlev, Yu.I., A Model of Recognizing Algorithms with Representative Sets and Systems of Support Sets, *Zh. Vychisl. Mat. Mat. Fiz.*, 1981, vol. 21, no. 5, pp. 1264–1275.
2. Dmitriev, A.I., Zhuravlev, Yu.I., and Krendelev, F.P., On Mathematical Principles of Classification of Objects or Phenomena, in *Diskretnyi analiz* (Discrete Analysis), Novosibirsk: Inst. Mat., Sibirsk. Otd., Akad. Nauk SSSR, 1966, issue 7, pp. 3–17.
3. Djukova, E.V., An Asymptotically Optimal Algorithm for Constructing Irredundant Tests, *Dokl. Akad. Nauk SSSR*, 1977, vol. 233, no. 4, pp. 527–530.
4. Djukova, E.V., Asymptotically Optimal Test Algorithms in Recognition Problems, in *Problemy kibernetiki* (Problems of Cybernetics), Moscow: Nauka, 1982, issue 39, pp. 165–199.
5. Djukova, E.V., On the Complexity of Implementation of Some Recognition Procedures, *Zh. Vychisl. Mat. Mat. Fiz.*, 1987, vol. 27, no. 1, pp. 114–127.
6. Djukova, E.V., Kora-Type Recognition Algorithms: Complexity of Implementation and Metric Properties, in *Raspoznavanie, klassifikatsiya, prognoz (matematicheskie metody i ikh primenenie)* (Recognition, Classification, Forecasting: Mathematical Methods and Their Application), Moscow: Nauka, 1989, issue 2, pp. 99–125.

7. Djukova, E.V. and Zhuravlev, Yu.I., Discrete Analysis of Feature Descriptions in Recognition Problems of Large Dimension, *Zh. Vychisl. Mat. Mat. Fiz.*, 2000, vol. 40, no. 8, pp. 1264–1278.
8. Djukova, E.V. and Inyakin, A.S., The Taxonomy Problem and Irredundant Coverings of Integer Matrices, in *Soobshcheniya po prikladnoi matematike* (Communications in Applied Mathematics), Moscow: Vychisl. Tsentr, Ros. Akad. Nauk, 2001.
9. Djukova, E.V. and Peskov, N.V., A Search for Informative Fragments of Object Descriptions in Discrete Recognition Procedures, *Zh. Vychisl. Mat. Mat. Fiz.*, 2002, vol. 42, no. 5, pp. 743–755.
10. O. Zhuravlev, Yu.I., On Algebraic Approach to Problems of Recognition or Classification, in *Problemy kibernetiki* (Problems of Cybernetics), Moscow: Nauka, 1978, issue 33, pp. 5–68.
11. Zhuravlev, Yu.I., On Recognition Algorithms with Representative Sets (on Logic Algorithms), *Zh. Vychisl. Mat. Mat. Fiz.*, 2002, vol. 42, no. 9, pp. 1425–1435.
12. Kuznetsov, V.E., On a Stochastic Algorithm for Computing Informative Characteristics of Tables by the Test Method, in *Diskretnyi analiz* (Discrete analysis), Novosibirsk: Inst. Mat., Sibirsk. Otd., Akad. Nauk SSSR, 1973, issue 23, pp. 8–23.
13. Ryazanov, V.V., On Construction of Optimal Recognition and Taxonomy (Classification) Algorithms in Solving Applied Problems, in *Raspoznavanie, klassifikatsiya, prognoz (matematicheskie metody i ikh primenenie)* (Recognition, Classification, Forecasting: Mathematical Methods and Their Application), Moscow: Nauka, 1988, issue 1, pp. 229–279.
14. Chegis, I.A. and Yablonskii, S.V., Logical Methods for Control of Electrical Circuits, *Tr. Mat. Inst. im. V.A. Steklova, Akad. Nauk SSSR*, 1958.
15. Yablonskii, S.V., *Vvedenie v diskretnuyu matematiku* (An Introduction to Discrete Mathematics), Moscow: Nauka, 1986.
16. Djukova, E.V., Discrete Recognition Procedures: The Complexity of Realization, *Pattern Recogn. Image Anal.*, 2003, vol. 13, no. 1, pp. 8–10.
17. Djukova, E.V. and Peskov, N.V., Selection of Typical Objects in Classes for Recognition Problems, *Pattern Recogn. Image Anal.*, 2002, vol. 12, no. 3, pp. 243–249.

Elena V. Djukova. Born 1945. Graduated from Moscow State University in 1967. Received her PhD (Kandidat Nauk) degree in physics and mathematics in 1979 and doctoral (Doctor Nauk) degree in physics and mathematics in 1997. Leading Researcher at the Computer Center, Russian Academy of Sciences. Scientific interests: discrete mathematics and mathematical methods of pattern recognition and image processing. Author of about 60 publications.