The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific and
Cultural Organization

IAEA
International Atomic Energy Agency

2060-1

Advanced School on Non-linear Dynamics and Earthquake Prediction

28 September - 10 October, 2009

Pattern Recognition Methods and Algorithms

V. Keilis-Borok
International Institute of Earthquake Prediction,
Theory and Mathematical Geophysics,
Moscow/Institute of Geophysics and Planetary Physics
Los Angeles
U.S.A.

A. Soloviev
International Institute of Earthquake Prediction,
Theory and Mathematical Geophysics
Moscow
RUSSIAN FEDERATION

www.mitp.ru/www.igpp.ucla.edu

Pattern Recognition Methods and Algorithms

V. Keilis-Borok(1, 2) & A. Soloviev (1)

(1) International Institute of Earthquake Prediction
Theory and Mathematical Geophysics
Russian Academy of Sciences
84/32 Profsouznaya st., Moscow 117997
Russian Federation
www.mitp.ru

(2) Institute of Geophysics and Planetary Physics &
Department of Earth and Space Sciences
University of California, Los Angeles,
405 Hilgard Ave., IGPP, Los Angeles, CA 90095-1567
USA
www.igpp.ucla.edu

I. INTRODUCTION

Pattern recognition is a useful tool for the analysis of behavior of nonlinear complex systems in absence of fundamental equations describing them. Application of this methodology belongs to the so-called "technical" analysis, consisting of a heuristic search for relationships between available system information and its features inaccessible for direct measurements.

Let a set of objects, phenomena or processes, which are connected with the same or similar systems, is considered. Certain information (for example, results of measurements) is available about each element of the set, and there is some feature, possessed only by a part of the elements. If possessing this feature by an element does not present evidently in the information available, then a problem arises to distinguish elements that possess this feature. This problem could be solved by constructing a model on the basis of mechanical, physical, chemical or other scientific laws, which could explain the relationship between the available information and the feature under consideration. But in many cases the complexity of the system makes the construction of such model difficult or practically impossible and it is natural to apply pattern recognition methods.

1.1 Examples of Problems to Apply Pattern Recognition Methods

Recognition of earthquake-prone areas (e.g., *Gelfand et al.*, 1976). A seismic region is considered. The problem is to determine in the region the areas where strong (with magnitude $M \geq M_0$ where M_0 is a threshold specified) earthquakes are possible. The objects are the selected geomorphological structures (intersections of lineaments, morphostructural nodes, etc.) of the region. The possibility for a strong earthquake to occur near the object is the feature under consideration. The available information is the topographical, geological, geomorphological and geophysical data measured for the objects.

The problem as the pattern recognition one is to divide the selected structures into two classes:

- structures where earthquakes with $M \geq M_0$ may occur;
- structures where only earthquakes with $M < M_0$ may occur.

Intermediate-term prediction of earthquakes (e.g., *Keilis-Borok and Rotwain*, 1990). A seismic region is considered. The problem is to determine for any time t will a strong (with magnitude $M \geq M_0$ where M_0 is a threshold specified) earthquake occur in the region within the period $(t, t + \tau)$. Here τ is a given constant. The objects are moments of time. The occurrence of a strong earthquake in time period τ after the moment is the feature under consideration. The available information is the values of functions on seismic flow calculated for the moment t.

The problem as the pattern recognition one is to divide the moments of time into two classes:

- moments, for which there is (or will be) a strong earthquake in the region within the period $(t, t + \tau)$;
- moments, for which there are not (or will not be) strong earthquakes in the region within the period $(t, t + \tau)$.

Recognition of strata filled with oil. The strata encountered by a borehole are considered. The problem is to determine what do the strata contain: oil or water. The objects are the strata. The filling of the strata with oil is the feature under consideration. The geological and geophysical data measured for the strata are the available information.

The problem as the pattern recognition one is to divide the strata into two classes:

- strata, which contain oil;
- strata, which contain water.

Medical diagnostics. A specific disease is considered. The problem is to diagnose the disease by using results of medical tests. The objects are examined people. The disease is the feature under consideration. The available information is the data obtained through medical tests.

The problem as the pattern recognition one is to divide examined people into two classes:

- people who have the disease;
- people who do not have it.

1.2 General Formulation of the Pattern Recognition Problem

One may give the general abstract formulation of the problem of pattern recognition as follows.

The set $W = \{ \mathbf{w}^i \}$ is considered, where objects $\mathbf{w}^i = (w_1^i, w_2^i, \ldots , w_m^i)$, $i = 1, 2, \ldots$ are vectors with real (integer, binary) components. We call these components by functions.

The problem is to divide the set W into two or more subsets, which differ in certain feature or according to clustering themselves.

There are two kinds of pattern recognition problems and methods:

- classification without learning;
- classification with learning.

1.3 Classification without Learning (Cluster Analysis)

The set W is divided into groups (clusters, see Fig. 1) on the basis of some measure in the m-dimensional space $w_1, w_2, \ldots, w_m$.

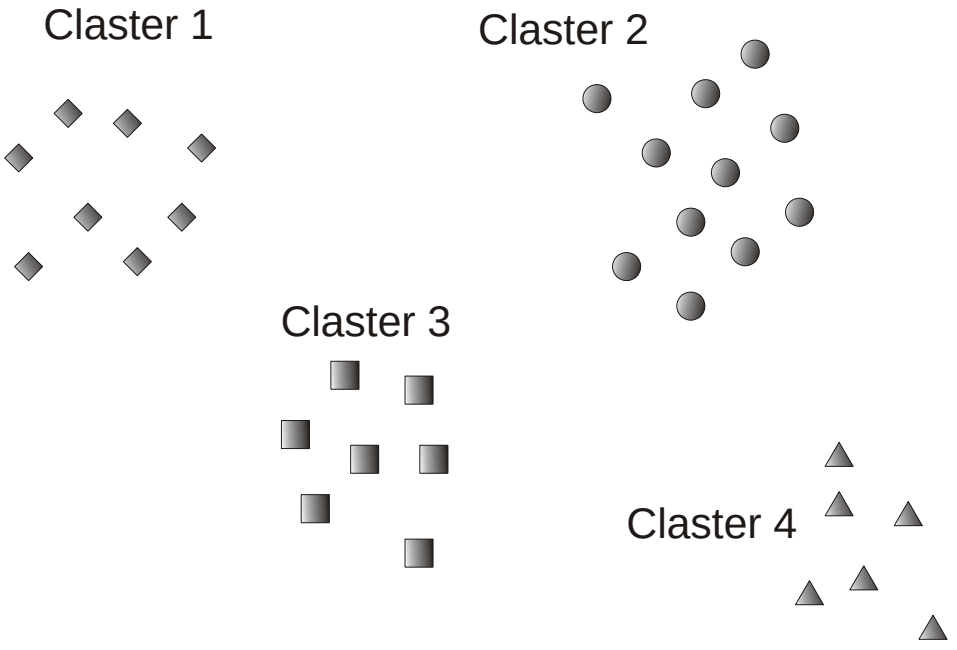

FIGURE 1 Clustering of objects in two-dimensional space

Denote $\rho(\mathbf{w}, \mathbf{v})$ a distance between two m-dimensional vectors $\mathbf{w} = (w_1, w_2,..., w_m)$ and $\mathbf{v} = (v_1, v_2,..., v_m)$.

To define classification and to estimate at the same time its quality the special function is introduced. The best classification gives the extremum of this function.

Examples of the functions. Let W is a finite set. The following two functions can be used.

$$J_1 = \frac{(K-1)\sum_{k=1}^{K}\rho_k}{2\sum_{k=1}^{K-1}\sum_{j=k+1}^{K}\rho_{kj}} \Rightarrow \min$$

$$J_2 = \frac{1}{K}\left(\sum_{k=1}^{K}\rho_k - \frac{2}{K-1}\sum_{k=1}^{K-1}\sum_{j=k+1}^{K}\rho_{kj}\right) \Rightarrow \min$$

Here K is the number of groups,

$$\rho_k = \frac{2}{m_k(m_k-1)}\sum_{i=1}^{m_k-1}\sum_{s=i+1}^{m_k}\rho(\mathbf{w}^i, \mathbf{w}^s),$$

$$\rho_{kj} = \frac{1}{m_k m_j}\sum_{i=1}^{m_k}\sum_{s=1}^{m_j}\rho(\mathbf{v}^i, \mathbf{v}^s),$$

m_k, m_j are the numbers of objects in the group numbered k and in the group numbered j respectively; $\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^{m_k}$ are the objects of the group numbered k; $\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^{m_j}$ are the objects of the group numbered j.

After the groups are determined the next problem can be formulated: to find common feature of objects, which belong to the same group.

1.4 Classification with Learning

If it is a priori known about some objects to what groups (classes) they belong, then this information can be used to determine classification for other objects.

As a rule the set W is divided into two classes, say D and N.

The a priori examples of objects of each class are given. They form the training set W_0:

$W_0 \subset W$,

$W_0 = D_0 \cup N_0$.

Here D_0 is the training set (the a priori examples) of objects belonging to class D, N_0 is the training set of objects belonging to class N.

The training set W_0 is used to determine a priori unknown distribution of objects of the set W_0 between the classes D and N.

The result of the pattern recognition is twofold:

- the rule of recognition; it allows to recognize which class an object belongs to knowing the vector $\mathbf{w}^i$ describing this object;
- the actual division of objects into separate classes according to this rule (Fig. 2):
  $W = D \cup N$
  or if there are objects with undefined classification then
  $W = (D \cup N) \cup U$.

Analysis of the obtained rule of recognition may give information for understanding the connection between the feature, which differs the classes D and N, on one hand and description of objects (components of vectors $\mathbf{w}^i$) on another.

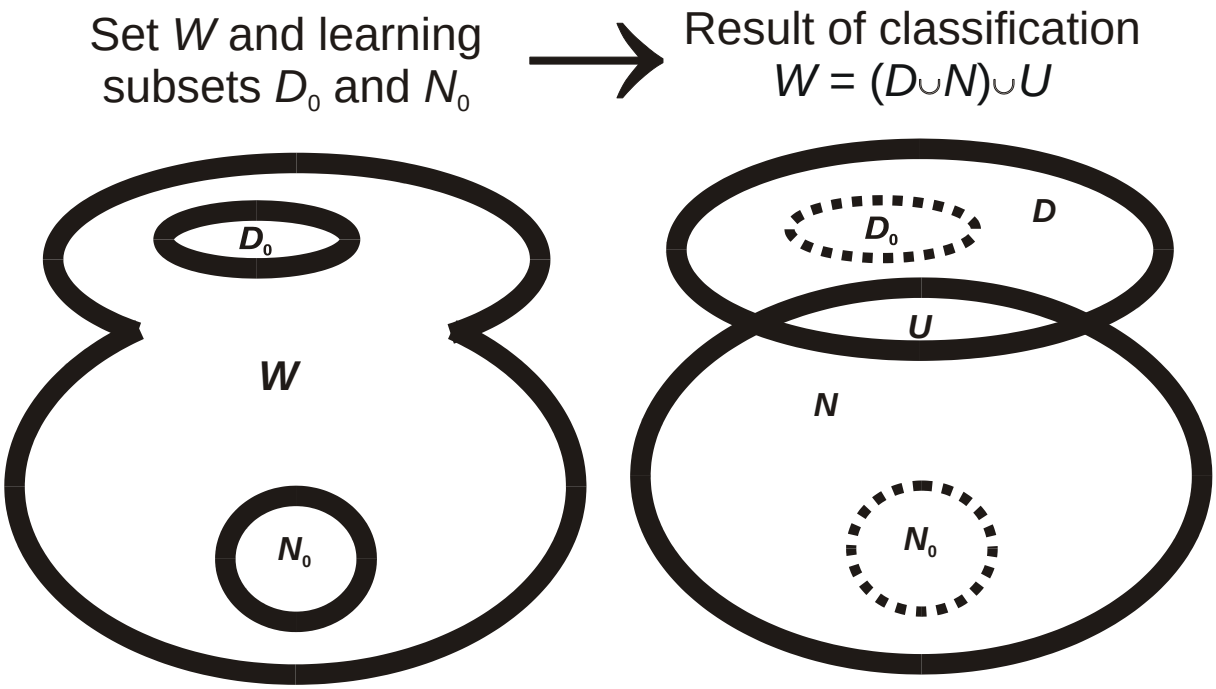

FIGURE 2 Classification with learning

II. EXAMPLES OF ALGORITHMS

Some algorithms used to solve problems of classification with learning are described below.

2.1 Statistical Algorithms

These algorithms are based on the assumption that distribution laws are different for vectors from classes D and N (see Fig. 3). The samples D_0 and N_0 are used to define the parameters of these laws.

The recognition rule includes calculating for each object $\mathbf{w}^i$ an estimation of conditional probabilities $P_D{}^i$ and $P_N{}^i$ that the object belongs to class D and N respectively. Classification of the objects according to these probabilities is performed as follows:

$$\mathbf{w}^i \in D, \text{ if } P_D{}^i - P_N{}^i \geq \varepsilon,$$
$$\mathbf{w}^i \in N, \text{ if } P_D{}^i - P_N{}^i < -\varepsilon,$$
$$\mathbf{w}^i \in U, \text{ if } -\varepsilon \leq P_D{}^i - P_N{}^i < \varepsilon,$$

where $\varepsilon \geq 0$ is a given constant.

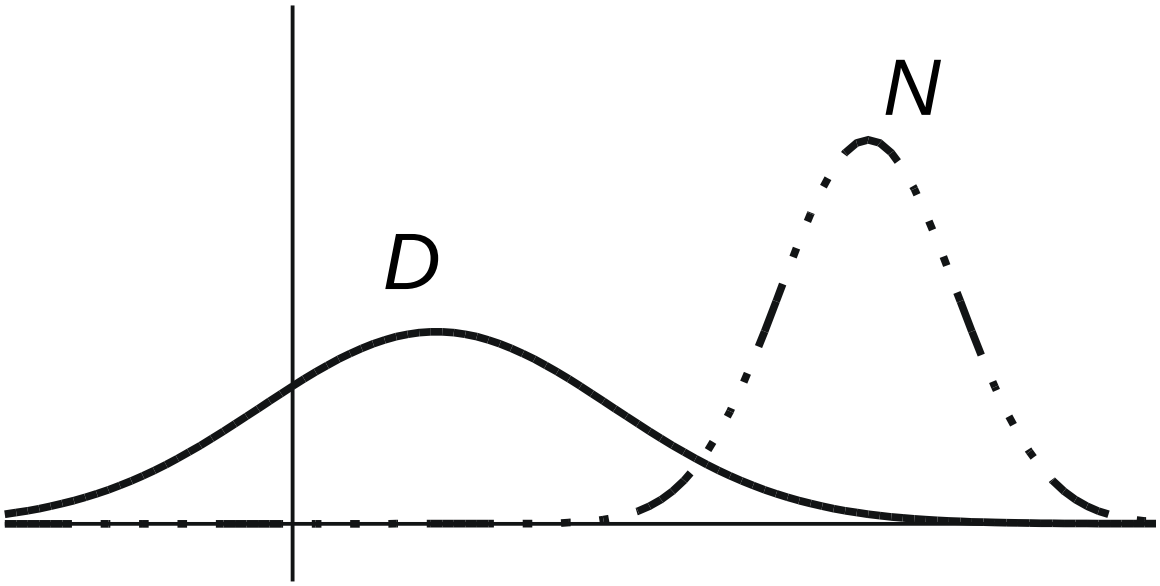

FIGURE 3 Different distribution laws for classes D and N

Bayes algorithm. One can apply this algorithm in the case when each component of vectors $\mathbf{w}$ may take only finite number of different values. If it is not so then discretization (see Section 3.1 below) should be made before application of the algorithm.

If the vectors of the set W are realizations of a certain random vector value then according to Bayes formula

$$P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D)\, P(\mathbf{w} \in D) = P(\mathbf{w} \in D | \mathbf{w} = \mathbf{w}^i)\, P(\mathbf{w} = \mathbf{w}^i) \qquad (1)$$

It follows from (1) that

$$P_D^i = P(\mathbf{w} \in D \mid \mathbf{w} = \mathbf{w}^i) = \frac{P(\mathbf{w} = \mathbf{w}^i \mid \mathbf{w} \in D) P(\mathbf{w} \in D)}{P(\mathbf{w} = \mathbf{w}^i)}.$$

Similarly

$$P_D^i = P(\mathbf{w} \in N \mid \mathbf{w} = \mathbf{w}^i) = \frac{P(\mathbf{w} = \mathbf{w}^i \mid \mathbf{w} \in N) P(\mathbf{w} \in N)}{P(\mathbf{w} = \mathbf{w}^i)}.$$

Estimations of probabilities in the right side of these relations are given by following approximate formulae, in which the samples D_0 and N_0 are used:

$$P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D) \approx P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D_0),$$

$$P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in N) \approx P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in N_0),$$

$$P(\mathbf{w} = \mathbf{w}^i) \approx P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D_0)\, P(\mathbf{w} \in D) + P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in N_0)\, P(\mathbf{w} \in N).$$

Probability $P(\mathbf{w} \in D)$ is a parameter of the algorithm and has to be given,
$P(\mathbf{w} \in N) = 1 - P(\mathbf{w} \in D)$.

Note that if m is the number of components in vectors $\mathbf{w}$ and k_j is the number of values that component w_j may take then the number of probabilities $P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D_0)$ (or $P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in N_0)$) that should be estimated is $k_1 \cdot k_2 \cdot \ldots \cdot k_m$. This number may be rather large even more than the number of objects in the set D_0 (or N_0) that makes impossible calculating probabilities $P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D_0)$ (or $P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in N_0)$).

The situation is simpler then the components of the vectors may be considered as independent random values. In this case

$$P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in D_0) = P(w_1 = w_1^i | \mathbf{w} \in D_0) \cdot P(w_2 = w_2^i | \mathbf{w} \in D_0) \cdot \ldots \cdot P(w_m = w_m^i | \mathbf{w} \in D_0)$$

and

$$P(\mathbf{w} = \mathbf{w}^i | \mathbf{w} \in N_0) = P(w_1 = w_1^i | \mathbf{w} \in N_0) \cdot P(w_2 = w_2^i | \mathbf{w} \in N_0) \cdot \ldots \cdot P(w_m = w_m^i | \mathbf{w} \in N_0).$$

Therefore only $2k_j$ conditional probabilities should be calculated for each component on the condition that a vector belongs to the set D_0 and on the condition that it belongs to the set N_0.

2.2 Geometrical Algorithms

In these algorithms surfaces in the space w_1, w_2,..., w_m are constructed to separate classes D and N (see Fig. 4).

Algorithm Hyperplane. This is an example of a geometrical algorithm.

The hyperplane $P(\mathbf{w}) = a_0 + a_1 w_1 + a_2 w_2 + \ldots + a_m w_m = 0$, where $a_1^2 + a_2^2 + \ldots + a_m^2 = 1$ is constructed in the space w_1, w_2,..., w_m to separate the sets D_0 and N_0 by the best way. It means that some function on the hyperplane has to have extremum value.

The example of the function is

$$J(a_0, a_1, \ldots, a_m) = \sum_{i=1}^{n_1} P(\mathbf{w}^i) - \sum_{i=1}^{n_2} P(\mathbf{v}^i) \Rightarrow \max.$$

Here $\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^{n_1}$ are objects of D_0, $\mathbf{v}^1, \mathbf{v}^2, \ldots, \mathbf{v}^{n_2}$ are objects of N_0.

The recognition rule is formulated as follows:

$$\mathbf{w}^i \in D, \text{ if } P(\mathbf{w}^i) \geq \varepsilon,$$
$$\mathbf{w}^i \in N, \text{ if } P(\mathbf{w}^i) < -\varepsilon,$$
$$\mathbf{w}^i \in U, \text{ if } -\varepsilon \leq P(\mathbf{w}^i) < \varepsilon,$$

where $\varepsilon \geq 0$ is a given constant.

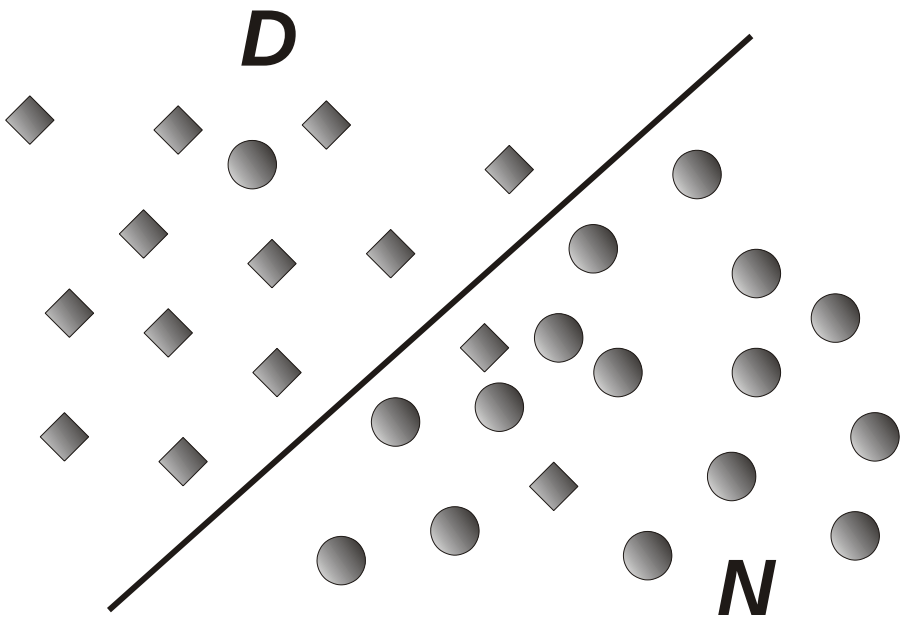

FIGURE 4 Separation of objects from classes *D* (rhombs) and *N* (circles) in two-dimensional space by a straight line.

2.3 Logical Algorithms

In these algorithms characteristic traits of classes *D* and *N* are searched using the sets D_0 and N_0. Traits are Boolean functions on w_1, w_2,..., w_m. The object $\mathbf{w}^i$ has the trait, if the value of the corresponding function, calculated for it, is ***true***, and does not have the trait, if it is ***false***. A trait is a characteristic trait of the class *D*, if the objects of the set D_0 have this trait more often than the objects of the set N_0. A trait is a characteristic trait of the class *N*, if the objects of the set N_0 have this trait more often than objects of the set D_0.

Using the searched characteristic traits the recognition rule is formulated as follows:

$$\mathbf{w}^i \in D, \text{ if } n_D^i - n_N^i \geq \Delta + \varepsilon,$$
$$\mathbf{w}^i \in N, \text{ if } n_D^i - n_N^i < \Delta - \varepsilon,$$
$$\mathbf{w}^i \in U, \text{ if } \Delta - \varepsilon \leq n_D^i - n_N^i < \Delta + \varepsilon.$$

Here n_D^i and n_N^i are the numbers of characteristic traits of classes *D* and *N*, which the object $\mathbf{w}^i$ has, Δ and $\varepsilon \geq 0$ are given constants.

Logical algorithms are useful to apply in cases then the numbers of objects in sets D_0 and N_0 are small.

As a rule logical algorithms are applied to vectors with binary components. An example of logical algorithm is the algorithm CORA-3. It is applied to geophysical problems, in particular to the problems of recognition of earthquake-prone areas and intermediate-term prediction of earthquakes. The detailed description of this algorithm can be found in *Gelfand et al.* (1976) and is given below.

III. PRELIMINARY DATA PROCESSING

As it was mentioned above some pattern recognition algorithms (e.g., CORA-3) do classify the vectors with binary components. Therefore, if the set *W* initially consists of vectors with real components (functions) then prior to an algorithm application, the coding of objects in the form of vectors with binary components has to be carried out. For this purpose, the characteristics are discretized, i.e. ranges of their values are represented as the union of disjoint parts. Then each of these parts is given accordingly by the value of a component of a binary vector or by the combination of values of its several components.

After discretization the data become robust. For example, if a range of some function is divided into three parts then only three gradations for this function ("small", "medium", "large") are used after the discretization instead of its exact value. Do not regret the loss of information. This makes results of recognition stable to variations of data.

3.1 Discretization

Let us consider some component (function) w_j of vectors (objects), which form the set *W*. Let the range of values of the function is limited with quantities x_0^j and x_f^j ($x_0^j < x_f^j$). The procedure of discretization for the function w_j consists of dividing the range into k_j intervals by thresholds of discretization (Fig. 5):

$$x_1^j, x_2^j, ..., x_{k_j-1}^j \ (x_0^j < x_1^j < x_2^j < ... < x_{k_j-1}^j < x_f^j).$$

Assume that the value w_j^i of the function numbered *j* of the object numbered *i* belongs to the interval numbered *s*, if $x_{s-1}^j < w_j^i \le x_s^j$, where $x_{k_j+1}^j = x_f^j$. After discretization we replace the exact value of the function by the interval, which contains this value.

Usually we divide the range of function values into two intervals ("small" and "large" values) or into three intervals ("small", "medium" and "large" values).

Thresholds of discretization can be introduced manually on the basis of various considerations for the nature of the given function.

The other way to determine the thresholds is to compute them so as to make the numbers of objects with the function values within each interval (x_{s-1}^j, x_s^j), s = 1, 2, ..., k_j, being roughly equal to each other. In this case one should specify the number of intervals k_j only. Then the thresholds of discretization may be calculated by using a special algorithm. All objects together or only objects of D_0 and N_0 may be considered. This type of discretization is called here and below as ***objective*** or ***automatic***.

Our purpose is to find such intervals where values of the function w_j for objects from one class occur more often than for objects from another class.

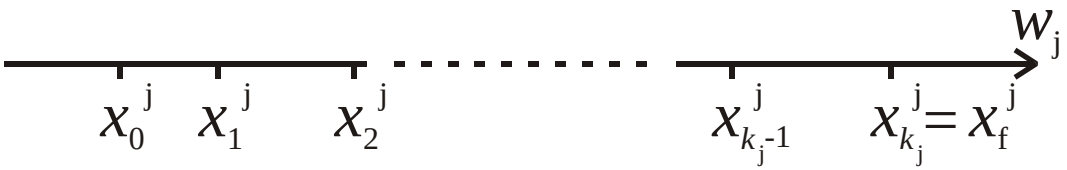

FIGURE 5 Discretization of the function w_j.

How informative is the function w_j in a given discretization can be characterized as follows.

Let us compute for each interval $(x_{s-1}^j,\ x_s^j)$ the numbers P_s^D and P_s^N (s = 1, 2, ..., k_j), which give for the sets D_0 and N_0 respectively the percent of objects, for which the value of the function w_j falls within the interval numbered s.

Let us denote $P_{\max} = \max\limits_{1\le s\le k_j}\left|P_s^D - P_s^N\right|$.

In other words P_s^D and P_s^N are empirical histograms of the function w_j for the sets D_0 and N_0, and $P_{\max}$ is the maximal difference of these histograms.

The larger is $P_{\max}$, the more informative is the function w_j.

Functions for which $P_{\max} < 10\%$ are usually excluded.

Another criterion of the quality of a discretization is ***monotonous dependence*** of P_s^D and P_s^N on s. Let $k_j = 3$. Let us denote:

$$M_D = \frac{\left|P_2^D - P_1^D\right| + \left|P_3^D - P_2^D\right|}{\left|P_3^D - P_1^D\right|},$$

$$M_N = \frac{\left|P_2^N - P_1^N\right| + \left|P_3^N - P_2^N\right|}{\left|P^N{}_3 - P_1^N\right|}.$$

If P_s^D changes monotonously with s, $M_D = 1$; the larger is M_D, more jerky is P_s^D. This is clear from Fig. 6. Similar statements are true for M_N, P_s^N.

The smaller are M_D and M_N, the better is the discretization of the function w_j. Functions with both M_D, $M_N \ge 3$ are usually excluded.

Samples D_0 and N_0 are often marginally small, so that their observed difference may be random. Therefore the relation between functions P_s^D and P_s^N after discretization should be not absurd according to the problem under consideration, though they may be unexpected indeed.

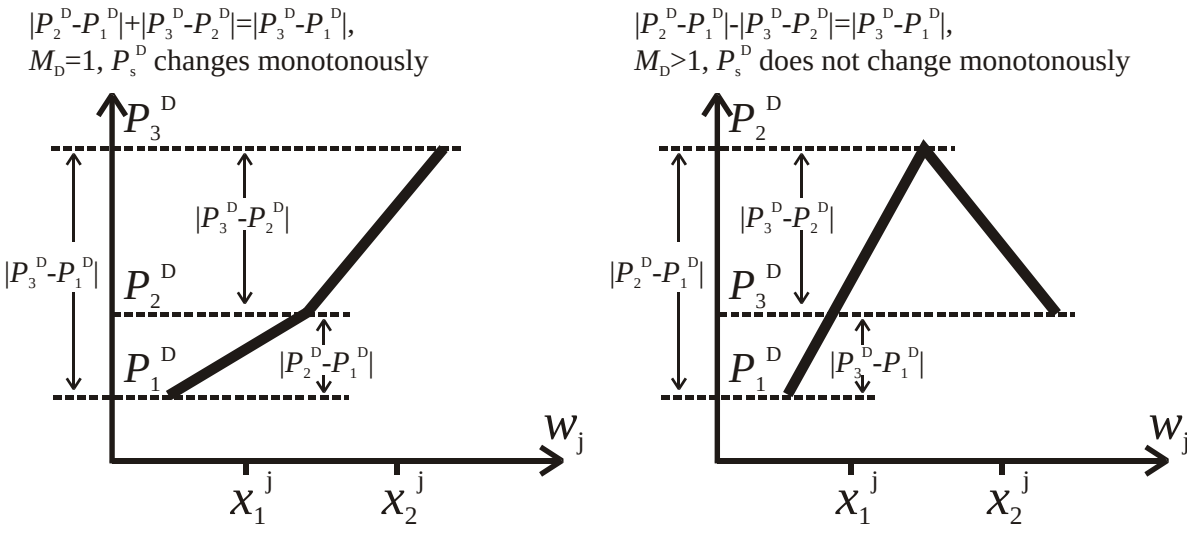

FIGURE 6 Monotonous and non-monotonous changing of P_s^D

3.2 Coding

With discretization thresholds determined, vectors $\mathbf{w}^i$ are coded into binary vectors. Only the functions selected at the stage of discretization are considered for coding. At the stage of coding l_j components of binary vectors are determined for the function w_j. Number l_j depends on the number of thresholds as well as on the type of coding procedure applied to the function w_j.

The following two types of coding are used.

1. ***I*** ("impulse") type. In this case $l_j = k_j$, i.e. the number of binary vector components allocated for the coding of the function w_j is equal to the number of intervals into which the range of its values is divided after discretization.

Let us denote as $\omega_1, \omega_2, ..., \omega_{lj}$ the values of binary vector components, which code the function w_j. If the value w_j^i of the function w_j for the object numbered i falls within the s-th interval of its discretization, i.e. $x_{s-1}^j < w_j^i \le x_s^j$, then we set

$\omega_1 = \omega_2 = ... = \omega_{s-1} = 0, \omega_s = 1, \omega_{s+1} = 0 = ... = \omega_{lj} = 0.$

2. ***S*** ("stair") type. In this case $l_j = k_j - 1$, i.e. the number of binary vector components, allocated for the coding of a function, is equal to the number of the thresholds of discretization. If the value w_j^i for the object numbered i falls within the s-th interval of its discretization, then we set

$\omega_1 = \omega_2 = ... = \omega_{s-1} = 0, \omega_s = \omega_{s+1} = ... = \omega_{lj} = 1.$

The case when the codes of the function w_j are constructed for $k_j = 3$ is considered below.

If the value w_j^i belongs to the first interval ($x_0^j < w_j^i \le x_1^j$) ***I*** type coding has the form: 100. ***S*** type coding for the same value w_j^i has the form: 11.

For the second interval ($x_1^j < w_j^i \le x_2^j$) the codes are 010 (***I*** type) and 01 (***S*** type).

For the third interval ($x_2^j < w_j^i \le x_3^j$) they are 001 and 00 respectively.

Discretization and coding procedures transform the set of vectors $W = \{ \mathbf{w}^i \}$, $i = 1, 2, ..., n$, which correspond to all objects, into a set of vectors with l binary components. Here $l = \sum' l_j$, where summation is implemented only over the functions left after discretization.

Thus, discretization and coding transform the initial problem in the form of the classification within the finite set of l-dimensional vectors with binary components. These vectors are also called objects of recognition.

IV. ALGORITHM CORA-3

Algorithm CORA-3 (*Bongard*, 1967) operates in two stages:

- selection of characteristic traits (*learning*);
- *voting*.

4.1 Learning

In the learning stage, the algorithm determines characteristic traits for classes *D* and *N* using vectors that from sets D_0 and N_0.

Traits. A matrix **A**,

$$\mathbf{A} = \left\| \begin{matrix} i_1 & i_2 & i_3 \\ \delta_1 & \delta_2 & \delta_3 \end{matrix} \right\|,$$

defines a trait, where i_1, i_2, i_3, $1 \le i_1 \le i_2 \le i_3 \le l$ are the numbers of binary vector components and δ_1, δ_2, δ_3 are their binary values. We say that a binary vector (an object) $\omega^i = (\omega_1^i, \omega_2^i, ..., \omega_l^i)$ has the trait **A** (or the Boolean function corresponding to this trait has a value ***true***) if $\omega_{i_1}^i = \delta_1, \quad \omega_{i_2}^i = \delta_2, \quad \omega_{i_3}^i = \delta_3$.

Characteristic traits. Let $W' \subseteq W$. Denote the number of vectors $\omega^i \in W'$ that have trait **A** by $K(W', \mathbf{A})$.

The algorithm has four free parameters $k_1, \bar{k}_1, k_2, \bar{k}_2$, which are nonnegative integers used to define characteristic traits of the two classes.

Trait **A** is a characteristic trait of class *D* if

$K(D_0, \mathbf{A}) \ge k_1$ and $K(N_0, \mathbf{A}) \le \bar{k}_1$.

Trait A is a characteristic trait of class *N* if

$K(N_0, \mathbf{A}) \ge k_2$ and $K(D_0, \mathbf{A}) \le \bar{k}_2$.

Parameters k_1 and k_2 are called selection thresholds for characteristic traits of classes *D* and *N* respectively. Parameters $\bar{k}_1$ and $\bar{k}_2$ are called contradiction thresholds for characteristic traits of classes *D* and *N*.

Equivalent, weaker, and stronger traits. The number of characteristic traits may be rather large. Some of them occur on the same vectors from training sets. The algorithm distinguishes such cases and does not include all characteristic traits in the final list.

Specifically, denote by $\Omega(\mathbf{A})$ a subset of set *W* such that $\omega^i \in \Omega(\mathbf{A})$ has trait **A**. Let, $\mathbf{A}_1$ and $\mathbf{A}_2$ be two characteristic traits of class *D*. Trait $\mathbf{A}_1$ is *weaker* than trait $\mathbf{A}_2$ (or $\mathbf{A}_2$ is *stronger* than $\mathbf{A}_1$), if

$\Omega(\mathbf{A}_1) \cap D_0 \subset \Omega(\mathbf{A}_2) \cap D_0$ and $(\Omega(\mathbf{A}_2) \cap D_0) \setminus (\Omega(\mathbf{A}_1) \cap D_0) \ne \varnothing$.

This condition means that all vectors from D_0 that have $\mathbf{A}_1$ also possess $\mathbf{A}_2$; at the same time there is at least one vector from D_0, which has trait $\mathbf{A}_2$, and does not have $\mathbf{A}_1$.

A similar definition is valid for characteristic traits of class *N*. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two characteristic traits of class *N*. Then the $\mathbf{A}_1$ is weaker than trait $\mathbf{A}_2$ (or $\mathbf{A}_2$ is stronger than $\mathbf{A}_1$) if

$\Omega(\mathbf{A}_1) \cap N_0 \subset \Omega(\mathbf{A}_2) \cap N_0$ and $(\Omega(\mathbf{A}_2) \cap N_0) \setminus (\Omega(\mathbf{A}_1) \cap N_0) \ne \varnothing$.

Two characteristic traits $\mathbf{A}_1$ and $\mathbf{A}_2$ of class *D* are called *equivalent* if they are found on the same vectors of set D_0, i.e.,

$\Omega(\mathbf{A}_1) \cap D_0 = \Omega(\mathbf{A}_2) \cap D_0$.

Similarly, characteristics traits $\mathbf{A}_1$ and $\mathbf{A}_2$ of class N are called equivalent if

$\Omega(\mathbf{A}_1) \cap N_0 = \Omega(\mathbf{A}_2) \cap N_0$.

The algorithm excludes from the lists of characteristic traits those that are weaker or equivalent to a selected trait.

Thus, the learning stage results in the final list of q_D and q_N characteristic traits of classes D and N. respectively. Any member of this list does not have weaker or equivalent members.

4.2 Voting and Classification

In the second stage the algorithm performs voting and classification using the final list of characteristic traits. For each vector $\omega^i \in W$, it calculates the number n_D^i of characteristic traits of class D, which the vector possesses, the number n_N^i of those of class N, and the difference $\Delta_i = n_D^i - n_N^i$ called voting.

The classification is defined as follows.

Class D (set D) is formed from the vectors ω^i, for which $\Delta_i \geq \Delta$. The vectors, for which $\Delta_i < \Delta$, are included in class N (set N).

Here Δ is a parameter of the algorithm as well as k_1, $\bar{k}_1$, k_2, and $\bar{k}_2$.

This recognition rule corresponds to $\varepsilon = 0$ in the description of logical algorithms given above.

4.3 Algorithm CLUSTERS

Algorithm CLUSTERS is the modification of algorithm CORA-3 (*Gelfand et al.*, 1976). It is applied in the case when set D_0 consists of S subsets (subclasses):

$D_0 = D_0^1 \cup D_0^2 \cup ... \cup D_0^S$,

and it is known a priori that at least one element of each subclass belongs to class D but some elements of set D_0 may belong to class N.

The learning stage of algorithm CLUSTERS differs from that of CORA-3 in the following.

First, by definition, a subclass has a trait if it contains at least one vector with this trait. Trait **A** is a characteristic trait of class D, if

$K^S(D_0, \mathbf{A}) \geq k_1$ and $K(N_0, \mathbf{A}) \leq \bar{k}_1$.

Here $K^S(D_0, \mathbf{A})$ is the number of subclasses that have the trait **A**.

Second, the definition of the weaker and equivalent traits for characteristic traits of class D is different. A characteristic trait $\mathbf{A}_1$ of class D is weaker than a characteristic trait $\mathbf{A}_2$ of the same class if any subclass that has trait $\mathbf{A}_1$ also has $\mathbf{A}_2$ and there is at least one subclass, which has trait $\mathbf{A}_2$ but does not have trait $\mathbf{A}_1$. Traits $\mathbf{A}_1$ and $\mathbf{A}_2$ are equivalent if they are found in the same subclasses.

Algorithm CLUSTERS forms the sets of characteristic traits of classes D and N like CORA-3.

The stage of voting and classification is the same as in algorithm CORA-3.

V. ALGORITHM HAMMING

Another algorithm applied to geophysical problems is algorithm HAMMING (*Gvishiani and Kossobokov*, 1981). There are also other possible applications of this algorithm (e.g., *Keilis-Borok and Lichtman*, 1981).

This algorithm operates also in two stages: learning and voting.

5.1 Learning

In the first stage (learning), the algorithm computes for each component ω_k (k = 1, 2, ..., l) of binary vectors the following values:

$q_D(k|0)$ - the number of objects of the set D_0, which have $\omega_k = 0$,

$q_D(k|1)$ - the number of objects of the set D_0, which have $\omega_k = 1$,

$q_N(k|0)$ - the number of objects of the set N_0, which have $\omega_k = 0$,

$q_N(k|1)$ - the number of objects of the set N_0, which have $\omega_k = 1$.

Then the relative number of vectors, for which this component equals to 1, is determined for the set D_0:

$$\alpha_D(k|1) = \frac{q_D(k|1)}{q_D(k|0) + q_D(k|1)}$$

and for the set N_0:

$$\alpha_N(k|1) = \frac{q_N(k|1)}{q_N(k|0) + q_N(k|1)} \ .$$

A binary vector $\mathbf{K} = (\kappa_1, \kappa_2, ..., \kappa_l)$ called the ***kernel of class D***, is determined as follows:

$$k_i = \begin{cases} 1, \text{ if } \alpha_D(i) \geq \alpha_N(i) \\ 0, \text{ if } \alpha_D(i) < \alpha_N(i) \end{cases}$$

The calculation of the kernel $\mathbf{K}$, whose components are more typical of set D_0 than of N_0 completes the first stage.

NOTE: It may be more reliable to eliminate the components, for which

$|\alpha_D(k|1) - \alpha_N(k|1)| < \varepsilon$, where ε is a small positive constant.

5.2 Voting and Classification

In the second stage, the algorithm computes Hamming's distances

$$\rho_i = \sum_{k=1}^{l} \left| \omega_k^i - \kappa_k \right|$$

from each vector $\omega^i \in W$ to the kernel of class D.

The classification is defined as follows.

Class *D* (set *D*) is formed from vectors ω^i, for which $\rho_i \le R$.

The vectors, for which $\rho_i > R$, are included in class *N* (set *N*).

Here *R* is a parameter of the algorithm.

Algorithm HAMMING-1 is generalization of HAMMING. It operates with the generalized Hamming's distance

$$\rho_i = \sum_{k=1}^{l} \left| \omega_k^i - \kappa_k \right| \xi_k .$$

Weights $\xi_k > 0$ (k = 1, 2, ..., l) are parameters of the algorithm. They can be assigned arbitrarily or computed from objective considerations that reduce the danger of self-deception; for example, by formula:

$$\xi_k = \frac{|\alpha_D(k|1) - \alpha_N(k|1)|}{\max\limits_k |\alpha_D(k|1) - \alpha_N(k|1)|}$$

where maximum is taken over all components used in the given run of the algorithm.

VI. EVALUATION OF THE CLASSIFICATION RELIABILITY

Reliability of results of recognition is evaluated by several methods including control tests, statistical analysis of the established classification and other techniques. These tests are necessary to be sure in the obtained results. It is especially important in the case of small samples D_0 and N_0. The tests illustrate - how reliable are the results of the pattern recognition. However they do not provide a proof in the strict statistical sense if the training material is small.

The following simplest tests are useful.

1. To save the part of objects from W_0 for recognition only, not using it in learning.
2. To check the conditions: $D_0 \subset D$, $N_0 \subset N$.

NOTE: Sometimes these conditions are not valid because sets D_0 and N_0 are not "clear" enough. For example, in the case of recognition of earthquake-prone areas objects of D_0 are structures where epicenters of earthquakes with $M \geq M_0$ are known and objects of N_0 are structures where epicenters of such earthquakes are not known. Objects of N_0 may belong to the class D, because in some areas earthquakes with $M \geq M_0$ may be possible, though yet unknown. Objects of D_0 may belong to the class N due to the errors in the catalog (in epicenters and/or magnitude).

The examples of some other tests are listed below. These tests include some variation of the objects, used components of vectors, numerical parameters etc. The test is positive if the results of recognition are stable to these variations. Since the danger of self-deception is not completely eliminated by these tests the design and implementation of new tests should be pursued.

6.1 Using a Result of Classification as a Training Set (RTS test)

This test is an attempt to repeat the established classification $W = D \cup N$, using the resultant sets D and N as the new training sets instead of D_0 and N_0. We usually consider this test as successful if not more than 5% of the total number of objects are classified in the test differently comparing with their initial classification. The "physical" idea of the test is rather obvious and natural: if our classification is correct then such changing of training material should not change the result of classification.

Note that algorithm CORA-3 allows easy repetition of initial classification if one takes $\bar{k}_1 = \bar{k}_2 = 0$ and sufficiently small k_1 and k_2. Therefore, it is advisable to perform this test with nonzero thresholds $\bar{k}_1$ and $\bar{k}_2$. For example, $\bar{k}_1 = \bar{k}_2 = 1$, or $\bar{k}_1 = \bar{k}_2 = 2$, or the same as in the initial classification. In the case of $\bar{k}_1 = \bar{k}_2 = 0$ the substantial information is carried with maximum values of k_1 and k_2, under which the initial classification can be repeated.

In the case of any algorithm used to obtain the initial classification, it's advisable to repeat it in making the test by using HAMMING algorithm. We consider success of RTS test as the necessary condition for the classification obtained to pretend to be the problem solution. In this sense RTS test is obligatory to check the reliability of the classification.

6.2 Stability Testing (ST tests)

These tests generalize RTS test. Their goal is to obtain the initial classification $W = D \cup N$, using the various subsets $D_0' \subseteq D$, $N_0' \subseteq N$ as D_0 and N_0 training sets. The test is considered successful if the initial classification is rather stable while we change training material. Usually we accept the result if not more than 10% of the total number of objects change their classification in the result of the test. The choice of D_0' and N_0' used as training sets in ST test

can be different. For instance, in the case of recognition of earthquake-prone areas the region at hand can be divided into two parts, and subsets D_0' and N_0' then formed from objects of the sets D and N objects with preimages belong to one part. The other way of selecting D_0' and N_0' can be based on voting results in the initial classification. If algorithm HAMMING (or HAMMING-1) is used, the objects $\mathbf{w}^i \in D$ close to the kernel $\mathbf{K}$ can be assigned to D_0', and those far from it are assigned to N_0'. When algorithm CORA-3 (or CLUSTERS) is used, the objects $\mathbf{w}^i \in D$ with larger values of Δ_i can be assigned to D_0', whereas those with small Δ_i form N_0'.

Successful results of different ST tests are appealing indirect arguments favoring the validity of an established classification. At the same time, a success in a single test with an arbitrary choice of D_0' and N_0' is by no means a proof of reliability.

6.3 Sliding Control (SC test)

This test is designed for establishing classifications on the basis of the training sets $(D_0 \backslash \mathbf{w}^i)$ and $(N_0 \backslash \mathbf{w}^{i+n_1})$, i = 1, 2, ..., max(n_1,n_2). The idea of SC test is very clear. We just want to check weather classification of the objects belonging to the training set is stable while they are excluded from the training set. The first variant discards the objects $\mathbf{w}^1 \in D_0$ and $\mathbf{w}^{1+n_1} \in N_0$, the second variant resets them but discards the objects $\mathbf{w}^2 \in D_0$ and $\mathbf{w}^{2+n_1} \in N_0$, etc. If one of sets D_0 or N_0 (with a smaller number of objects) has already all its objects discarded once, we proceed only with the other set. In case of algorithm CLUSTERS the whole subclasses are excluded in turn from the set D_0.

Formal criteria of success of the test is small value of ratio $\frac{m_D}{|D_0|}$ or $\frac{m_D + m_N}{|D_0| + |N_0|}$. Here m_D and m_N show how many objects of D_0 and N_0 respectively change classification after they were excluded from learning. We usually consider SC test as successful if not above 20% of objects in each of D_0 and N_0 sets change their classification while neglecting.

This test is very similar to the well-known "jackknife" procedure, under which each variant discards only one object, first from D_0, and then from N_0. On the other hand SC is preferable because it needs executing less variants of classification.

6.4 Voting by Equivalent Traits (VET test)

This test is applied only if classification is obtained by CORA-3 (or CLUSTERS) algorithm. In both cases the result of classification depends on the choice of traits picked up from equivalence groups. The VET test aims at evaluating the classification stability under such a choice.

Let object $\mathbf{w}^i$ possesses u_{Dj}^i traits, which are equivalent to j-th trait of class D, and u_{Nj}^i traits, which are equivalent to j-th trait of class N. We define on the bases of numbers u_{Dj}^i and u_{Nj}^i the numbers of "votes" in favor of classes D and N respectively as follows.

$$u_D^i = \sum_{j=1}^{p_D} \frac{u_{Dj}^i}{p_D^j}, \quad u_N^i = \sum_{j=1}^{p_N} \frac{u_{Nj}^i}{p_N^j} .$$

Here p_D^j is the total number of traits equivalent to j-th trait of class D, p_N^j - the number of traits equivalent to j-th trait of class N. In calculation of both numbers p_D^j and p_N^j j-th trait itself is obviously included. In the test the set D is formed from the objects, which satisfy the condition $u_D^i - u_N^i \geq \Delta$ and the rest of objects forms the set N.

The results of the VET test are claimed successful if it is possible to find Δ such that the total change in classification is less than 5% of the total number of recognition objects. We consider a success of the VET test as a necessary precondition for claiming the validity of the resultant classification obtained with CORA-3 or CLUSTERS.

6.5 Randomization of Data

These tests (*Gvishiani and Kossobokov*, 1981) are used to estimate the probability of an erroneous classification and its nonrandomness in the absence of a control sample.

The sequence of intermixed problems is considered in these tests. An intermixed problem is formulated on the basis of initial one by a random choice of n_1 objects from given n objects of the set W and also by a random choice n_2 objects from the rest of n - n_1 objects of the set W. These two new random training sets we symbolize as D_0' and N_0'. Coding of the objects in the form of binary vectors remains the same for an intermixed problem as it is in the real one. In other words it means that we preserve the relationship between the characteristics, which organic one to the set W as a whole. The total number $C_n^{n_1} C_{n-n_1}^{n_2} = n!/[n_1! n_2! (n - n_1 - n_2)!]$ of intermixed problems may be defined.

A pattern recognition algorithm is applied to each intermixed problem, and the classification $W = D \cup N$ based upon the training sets D_0' and N_0' is obtained in the given intermixed problem. The condition that $|D|$ is not greater than the number of objects in the set D obtained in the initial classification is imposed on the classification in the intermixed problem.

Assume that F of intermixed problems have been formed and f_1 among them succeeded to include $D_0' \subseteq D$. Then f_1/F ratio may be used as the measure of the result to be non-random. If the values of f_1/F are small it obviously means that, it is complicated to obtain a random result of the same quality as the real one. In this sense the small values of f_1/F speak for the fact that the real result obtained is non-random. On the other hand it cannot of course be used as a necessary condition to proceed with the classification.

Gvishiani and Kossobokov (1981) showed that under some natural additional requirements classifications in intermixed problems offer to define the upper estimate of classification error probability for the original problem. This upper estimate is calculated by the formula

$$\bar{p} = |\overline{N}| / n - \bar{\nu}_D / n_1.$$

Here $|\overline{N}|$ is the average number of objects allocated to class N in the intermixed problems, $\overline{\nu_D}$ - the average number of objects from sets D_0' allocated to N in the intermixed problems.

Naturally, a small value of $\bar{p}$ is the argument favoring the validity of classification obtained for the original problem. If the estimation results in a large value ($\bar{p}$ > 0.5), it is advisable to return to the original problem. Such a situation may indicate, for instance, an insufficient size of D_0. On the other hand, one should remember that $\bar{p}$ gives only the upper estimation of the error probability, though its value is usually much less.

6.6 Result Replication Tests

These tests are the attempts to replicate the obtained result by altering the solution procedure starting with some intermediate stage. The application of another pattern recognition algorithm is used in the simplest example of such experiment. For example, classification was established by performing CORA-3 algorithm, then, using that same coding of objects, an attempt is made to repeat the classification by applying HAMMING algorithm. This test is

usually considered as satisfactory one if not more than 20% of objects change their classification.

When application of a simpler algorithm results in repeating almost entirely the initial classification, its validation rises, of course. On the other hand replication of the classification by another algorithm cannot be considered, of course, as the necessary condition for the result to be valid.

The set of used components of binary vectors may be changed. In particular this may include elimination of each used component in turn.

An attempt may be also made to repeat the classification altering discretization thresholds for the functions describing the objects. Corresponding changes in coding of the objects should be also made. New functions may be included in the description of the objects. Then by replication of all subsequent stages of the problem consideration, a new classification is established and its comparison with the initial is made.

REFERENCES

Bongard,M.M. (1967). Classification Problem, Nauka, Moscow (in Russian).

Gelfand,I.M., Sh.A.Guberman, V.I.Keilis-Borok, L.Knopoff, F.Press, I.Ya.Ranzman, I.M.Rotwain, and A.M.Sadovsky (1976). Pattern recognition applied to earthquake epicenters in California. *Phys. Earth Planet. Inter.*, **11**: 227-283.

Gvishiani,A.D. and V.G.Kossobokov (1981). On foundations of the pattern recognition results applied to earthquake-prone areas. *Izvestiya Acad. Sci. USSR. Physics of the Earth*, 2: 21-36 (in Russian).

Keilis-Borok,V.I. and A.J.Lichtman (1981). Pattern recognition applied to presidential elections in the United States, 1860-1980: role of integral social, economic and political traits. *Proceedings of US National Ac. Sci.*, 78, 11: 7230-7234.

Keilis-Borok,V.I. and I.M.Rotwain (1990). Diagnosis of Time of Increased Probability of strong earthquakes in different regions of the world: algorithm CN. *Phys. Earth Planet. Inter.*, **61**: 57-72.