Advanced School on Understanding and Prediction of Earthquakes and other Extreme Events in Complex Systems

26 September - 8 October 2011

User's Guide for Programs CODM and PRAL

A.Soloviev

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

THE PACKAGE FOR PATTERN RECOGNITION

I. COMMON NOTATIONS

The authors used these programs mainly for recognition of earthquake-prone areas (Gelfand et al., 1976; Gvishiani and Kossobokov, 1981) and for research in earthquake prediction (Sadovsky, 1986; Keilis-Borok and Rotwain, 1990). However there is a wide variety of other possible applications (for example, Keilis-Borok and Lichtman, 1981).

The authors should emphasize that all the methods of earthquake prediction studies suggested are yet insufficiently tested. The actual forward prediction can be made so far on experimental basis only and the users are advised to disclaim any liability for the consequences of the actions, based on their predictions. Accordingly the authors disclaim the liability for any consequences of the use of these programs. Let us remind also, that the concrete prediction of a strong earthquake has to be released only by a qualified scientific body to the proper authorities (see for example the code, established under the auspices of UNESCO and IUGG).

Following are:
(i) The content of the package
(ii) For each program:

- the purpose of the program;
- the description of the dialogue;
- the example of printout.

Content of the package

The package includes the following programs written in C for PC compatible computers:

CODM (and its variant **CODMF**, which uses output **PROFILE** of the program **FUNC** as input file)

- discretization of the functions;
- coding the objects as binary vectors.

PRAL

- application of the pattern recognition algorithm **CORA-3** and its modification **CLUSTERS**;
- application of the pattern recognition algorithm **HAMMING**.

The algorithms are applied to vectors with binary components.

Initial data - objects of recognition specified as vectors with real or binary components.
The programs work in an interactive mode. The necessary input data are requested by using window interface.

We suggest that the user is acquainted with algorithms. The descriptions of **CORA-3** may be found in Gelfand et al. (1976). **HAMMING** is described by Gvishiani and Kossobokov (1981) and Keilis-Borok and Lichtman (1981).

WARNING. The estimation of significance of pattern recognition results requires numerical tests. Examples of logical tests can be found in Gelfand et al. (1976), of statistical tests - in Gvishiani and Kossobokov (1981).

To users.
The package is modified and expanded from time to time according to the experience of its applications. Any suggestions or information on results obtained will be kindly appreciated. Please send both to
Dr.A.A.Soloviev,
International Institute of Earthquake Prediction Theory and Mathematical Geophysics,
Russian Academy of Sciences,
84/32 Profsouznaya st., Moscow 117997, Russian Fedration
Phone: +7 495 333 45 13, Fax: +7 495 333 41 24
E-mail: soloviev@mitp.ru

II. PROGRAM CODM

2.1 Purpose

The program is used for:

- indication of the class (and subclass) for the objects;
- making histograms;
- binary coding.

There are three classes: the first and the second classes are used for learning, the third one - only for voting. If necessary, subclasses in the first class can be formed.

Histograms of functions can be made separately for each class of objects. Thresholds for histograms can be entered in or calculated automatically; p - 1 thresholds divide the range of the function into p intervals. For automatic calculation of thresholds you have to specify p. In this case the program calculates thresholds to place almost the same number of objects into each interval.

There is also a possibility to make a table ("table-histogram") showing the distribution of the objects according to the values of the function. The range of the function is divided by thresholds with a constant step. For each interval the table lists the names of the objects with the values of the function within this interval.

The table with two-dimensional distribution of objects according to the values of two functions can be constructed. By using the thresholds for discretization of the functions the objects can be coded as vectors with binary components.

There are two methods of coding: ***I***-method and ***S***-method. Let $x_1 < x_2 < \ldots < x_{p-1}$ be the thresholds for discretization, x_0 and x_p- the bounds of the function range. If the value of the function is x, and $x_{j-1} < x \leq x_j$, $(1 \leq j \leq p)$, then the function value is coded as

$0\ \ 0 \ldots\ 0\ \ 1\ \ 0\ \ \ldots\ 0\ \ 0$ - in ***I***-method,
$0\ \ 0 \ldots\ 0\ \ 1\ \ 1\ \ \ldots\ 1$ - in ***S***-method.
$1\ \ 2 \ldots j\text{-}1\ \ j\ \ j\text{+}1 \ldots p\text{-}1\ \ p$

2.2 Files

The program **CODM** reads the values of the functions of the objects from input file in ASCII code. The extension of the name of this file should be **.PAT**. This file has to have the following format:

- *the first line*: the number of functions and the number of objects;
- *the next line or several lines*: the names of functions (5 symbols each);
- *the next line or several lines*: the names of objects (5 symbols each);
- *the next lines*: values of functions of objects (an object begins from a new line). The values, which are greater than 99999 are interpreted as undefined.

The following is an example of such file (The file contains information about values of 2 functions with names **f** and **g** for 15 objects with names M1, M2, M3, M4, M5, M6,M7, M8, M9, M10, M11, M12, M13, M14, M15):

```
 2 15
    f    g
   M1   M2   M3   M4   M5   M6   M7   M8   M9  M10  M11  M12  M13
  M14  M15
 0 2
 17.5 3
 50 2
 35 3
 0 2
 0 2
 15 2
 25 3
 0 2
 0 4
 15 2
 24 2
 27.5 2
 30 2
 0 3
```

There is a variant of the program with name **CODMF**, which uses output **PROFILE** of the program **FUNC** with values of functions as input file. The extension of the name of this file should be **.PAT**.

The program may store the following data on a disk in **PROFILE**: the name of the input file; index of classes for objects; composition of subclasses of the first class; thresholds of discretization and the method of coding for each function. The extension of the name of this file should be **.COD**. This information may be used in further runs of the program.

The program may create a file on a disk with coding of objects, identification of classes and subclasses, names of objects and functions. The extension of the name of this file should be **.RAT**. This file is used by the program **PRAL**.

The program may create a ASCII format text file on a disk with histograms and coding for printout. The name of this file is **COD.PRI**. The number of columns in the file does not exceed 120.

2.3 Dialog

To input data the program uses the interface organized as a set of menus with accessible windows.

First menu

The first menu is selection of the mode. To select the necessary mode you have to move the cursor to the proper window and press **ENTER**. There are three possible modes:

- discretization (only discretization is made);
- coding (only coding is made);
- discretization and coding (discretization and coding are made).

By pressing **F10** it is possible to terminate the program.

Second menu

The second menu contains the window where the name of the input **PROFILE** has to be specified. The name has to have the extension .**COD**. **Blank** means that input **PROFILE** will not be used. The function key **F9** may be used to search the file in the current directory.

By pressing **F10** it is possible to terminate the program.

Third menu

The third menu contains the following windows.

#	Window Name	Function and Comments
1	Input file	To specify ***the name of the file with values of functions of objects***. The name has to have extension .**PAT**. The function key **F9** may be used to search the file in the current directory.
2	Discr.	To specify ***the number of classes, which will be used in discretization***. This window is used only if discretization is made.
3	Codir.	To specify ***the number of classes, the objects from which will be coded***. This window is used only if coding is made.
4	Classes	To indicate or change ***class for each object***. **CURRENT** - current definition of classes is used. **1 FOR ALL** - all objects will belong to the 1st class. **2 FOR ALL** - all objects will belong to the 2nd class, etc. The window can be opened by pressing **F2** to screen the names of objects and to change their classes.
5	Type	To set ***whether coding will be typed***. **ON** - coding will be typed. **OFF** - coding will not be typed. This window is used only if coding is made.
6	Print	To set ***whether printout will be created***. **ON** - the file **COD.PRI** will be opened. If coding is made the printout of coding will be written in the file **COD.PRI**. **OFF** - the file **COD.PRI** will not be created.

7	Subclasses	To introduce ***subclasses in the first class***. **ON** - subclasses in the first class are specified. In this case the window can be opened by pressing **F2** to define or to change subclasses. **OFF** - subclasses are not specified. This window is used only if coding is made.
8	File with coding	To specify ***the name of the output file with coding***. The name has to have the extension .**RAT**. **Blank** means that a file with coding will not be created. This window is used only if coding is made.
9	Output Profile	To specify ***the name of the output PROFILE***. The name has to have extension .**COD**. **Blank** means that output **PROFILE** will not be created.
10	Control	To show ***consistency of input data*** already indicated. **OK** - data are consistent and if the cursor is moved to this window ***the fourth menu*** can be reached by pressing **ENTER**. **???** - data are not consistent.

By pressing **ESC** it is possible to return to the first menu.
By pressing **F10** it is possible to terminate the program.

Fourth menu

The fourth menu shows the list of functions with thresholds for discretization and methods of coding. By moving the cursor it is possible to choose thresholds or methods of coding to be corrected. For each function by pressing **F2** it is possible to pass to ***the fifth menu***. When **ESC** is pressed the program passes to coding or to the end of work according to the mode selected.

By pressing **F10** it is possible to terminate the program.

Fifth menu

The fifth menu contains four operations, which may be made for the function selected.

- **Diagram of discretization** - shows a histogram (numbers and percent of objects from each class for each interval of discretization).
- **Automatic discretization** - calculates automatically the thresholds for discretization. The number of intervals has to be specified.
- **Table-histogram** - shows the table-histogram for the function. The step of the table has to be specified.
- **Two-dimensional table** - shows two-dimensional distribution of objects according to the values of two functions: the function under consideration and any other function. The name of the other function, minima, maxima and steps for the functions have to be specified.

To execute the operation it is necessary to make specifications, to move the cursor to the proper window and, to press **ENTER**. In making of tables and diagrams the right limits of intervals are included in the proper intervals.

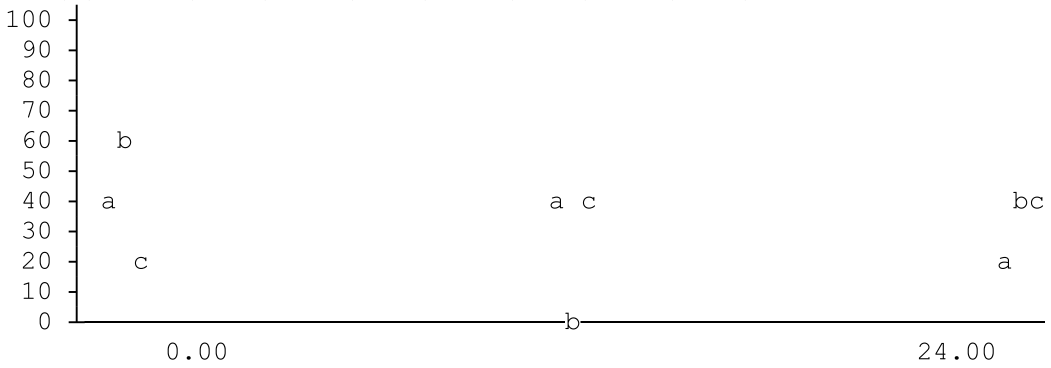
By pressing **ESC** it is possible to return to the fourth menu.

By pressing **F10** it is possible to terminate the program.

2.4 Example of Printout

```
Input profile - ex25.cod
File with values of functions - ex25.pat
   number of objects = 15, number of functions = 2
Mode: discretization and coding
Number of classes for coding = 3; subclasses - off
Number of classes for discretization = 3
Output profile - ex25.cod

 thresholds for 1 function f obtained by a-method
                0.00    24.00
 class:                                  undefined
  1(a):   2( 40%)   2( 40%)   1( 20%)   0(  0%)
  2(b):   3( 60%)   0(  0%)   2( 40%)   0(  0%)
  3(c):   1( 20%)   2( 40%)   2( 40%)   0(  0%)
```



```
f: min=0.00 max=50.00 step=5.00
g: min=2.00 max=4.00 step=0.20

      2.0    2.8    3.8    4.0
 50.0 +------+------+------+
      |-  M3 |      |      |
 45.0 +------+------+------+
      |      |*  M4 |      |
 30.0 +------+------+------+
      |*  M13|      |      |
      |-  M14|      |      |
 25.0 +------+------+------+
      |*  M12|+  M8 |      |
 20.0 +------+------+------+
      |      |+  M2 |      |
 15.0 +------+------+------+
      |+  M7 |      |      |
      |*  M11|      |      |
 10.0 +------+------+------+
      |+  M1 |-  M15|+  M10|
      |-  M5 |      |      |
      |*  M6 |      |      |
      |-  M9 |      |      |
 -0.0 +------+------+------+
thresholds for 2 function g obtained by h-method
                 2.00
  class:                      undefined
   1(a):   2( 40%)   3( 60%)   0(  0%)
   2(b):   4( 80%)   1( 20%)   0(  0%)
   3(c):   4( 80%)   1( 20%)   0(  0%)
 100 -|
  90 -|
  80 -|              bc
  70 -|
  60 -|                                           a
  50 -|
  40 -|             a
  30 -|
  20 -|                                            bc
  10 -|
   0 -+-------------------------------------------------------------
                                  2.00

 2 function g : step=0.500

    +-------------------+-------------------+-------------------+
    | 1 class           | 2 class           | 3 class           |
    +-------------------+-------------------+-------------------+
    |    M1    M7       |    M3    M5    M9 |    M6   M11   M12 |
    |                   |   M14             |   M13             |
2.00+-------------------+-------------------+-------------------+
    |    M2    M8       |   M15             |    M4             |
3.00+-------------------+-------------------+-------------------+
    |   M10             |                   |                   |
4.00+-------------------+-------------------+-------------------+

length of vectors = 4
       discretization and coding of functions:
           method
 function of coding        values of thresholds
   f          s         0.00   24.00
   g          i         2.00

     vectors of class 1(D0)

               f|  g|
                |   |
                |   |
                |   |
                |   |
               2   4
   1.     M1  1 1|1 0|
   2.     M2  0 1|0 1|
   3.     M7  0 1|1 0|
   4.     M8  0 0|0 1|
   5.    M10  1 1|0 1|

vectors of class 2(N0)

   1.     M3  0 0|1 0|
   2.     M5  1 1|1 0|
   3.     M9  1 1|1 0|
   4.    M14  0 0|1 0|
   5.    M15  1 1|0 1|
     vectors of class 3(X)

   1.    M4  0 0|0 1|
   2.    M6  1 1|1 0|
   3.   M11  0 1|1 0|
   4.   M12  0 1|1 0|
   5.   M13  0 0|1 0|
```

2.5 Comments to the Example of Printout

The printout contains the following:

- The histogram for the function **f**. The thresholds were calculated automatically. Symbol "**a**" indicates points of the histogram for learning objects of the first class, symbol "**b**" - for learning objects of the second class, symbol "**c**" - for objects of the third class. Number of intervals to divide the function's range equals to **3**.
- Two-dimensional table for the functions **f** and **g**.
- The histogram for the function **g**. Number of intervals to divide the function range into equals to **2**. The threshold **2.0** was entered in.
- The table-histogram for the function **g**.
- The coding of the objects. The function **f** was coded by **s-method**, the function **g** - by **i-method**.

III. PROGRAM PRAL

3.1 Purpose

The program is used for application of the pattern recognition algorithm **CORA-3** (Gelfand et al., 1976), its modification **CLUSTERS** (Gelfand et al., 1976), and the pattern recognition algorithm **HAMMING** (Gvishiani and Kossobokov, 1981; Keilis-Borok and Lichtman, 1981) to objects, which are binary vectors. The algorithm **CORA-3** and its modification **CLUSTERS** are described in detail in (Gelfand et al., 1976). The description of the algorithm **HAMMING** is given below.

Algorithm Hamming

The learning in this algorithm consists of calculation of ***a kernel***. Two values $\alpha_1(i)$ and $\alpha_2(i)$ are calculated for each used component (i is the number of the component)

$$\alpha_1(i) = \frac{n_1(i)+1}{N_1+2}, \quad \alpha_2(i) = \frac{n_2(i)+1}{N_2+2}.$$

Here N_1 is the number of objects of the first class in the learning, $n_1(i)$ is the number of those of them, which possess 1 in the i-th component; N_2 and $n_2(i)$ are similar numbers for objects of the second class. The kernel is the vector $\mathbf{k} = (k_1, k_2, ..., k_l)$, where

$$k_i = \begin{cases} 1, \text{ if } \alpha_1(i) \geq \alpha_2(i), \\ 0, \text{ if } \alpha_1(i) < \alpha_2(i), \end{cases}$$

l is the number of used components.

Voting is calculation of the Hamming's distance r_j between each object and the kernel

$r_j = w_1 \mid x_1^j - k_1 \mid + w_2 \mid x_2^j - k_2 \mid + ... + w_l \mid x_l^j - \mathrm{k}_l \mid$.

Here *j* is the number of the object, x_i^j is the value of the *i*-th used component of the *j*-th vector, w_i is the given weight for the *i*-th used component. Weights are real numbers between 0 and 1 and multiples of 0.1.

An object is classified as the object of the first class if r_j, calculated for it, is not greater than a chosen threshold Δ. The objects, for which r_j is greater than Δ are classified as the objects of the second class.

3.2 Modes of Computation

During one run of the program several variants can be calculated. Variants can have the following differences:

- the algorithm selected;
- the composition of learning sets of the first and second classes;
- the components of vectors, which are used;
- the values of thresholds for characteristic traits of the first and the second classes (if **CORA-3** or **CLUSTERS** is applied);
- the values of weights (if **HAMMING** is applied);
- the control experiment executed;
- the kind of output.

The control experiments are:

- Test excluding (**T-E**) of objects;
- **T-E** of functions;
- **Equivalent traits** (if **CORA-3** or **CLUSTERS** is applied).

T-E of objects is additional calculation of $n_{\text{T-E}}$ variants, where $n_{\text{T-E}}$ =max (N_1 ,N_2), N_1 and N_2 are numbers of objects in the learning sets of the first and the second classes respectively. In the first of these variants the first object from the learning set of the first class and the first object from the learning set of the second class are excluded from the learning; in the second variant the pair of the first objects is returned and the pair of the second objects is excluded, and so on. After excluding and returning the last object of the lesser learning set the objects from the greater learning set only are excluded one by one. When the program deals with subclasses, instead of separate objects of the first class the subclasses are excluded one by one in **T-E** of objects.

T-E of functions is an additional calculation of *m* variants, where *m* is the number of functions used in the variant. In each of these variants the functions of the main variant are excluded one at a time.

Equivalent traits is an experiment, in which the ε-equivalent, ε-covering, or ε-subordinating traits for the characteristic traits obtained in the variant are selected. Let ***A*** be a characteristic trait included in the final list of traits in the variant, ***B*** - a characteristic trait of the same class selected in the experiment, *M* - the learning sample of the corresponding class.

Denote the numbers of objects from the set *M*, which have the trait ***A***, the trait ***B***, the both traits ***A*** and ***B***, and at least one of the traits ***A*** or ***B*** by $\mu(\boldsymbol{A})$, $\mu(\boldsymbol{B})$, $\mu(\boldsymbol{A}\cap\boldsymbol{B})$, and $\mu(\boldsymbol{A}\cup\boldsymbol{B})$ respectively.
If $(\mu(\boldsymbol{A}\cup\boldsymbol{B}) - \mu(\boldsymbol{A}\cap\boldsymbol{B}))/ \mu(\boldsymbol{A}\cup\boldsymbol{B}) \leq \varepsilon$, than the trait ***B*** is a **ε-equivalent** trait for the trait ***A***.
If $(\mu(\boldsymbol{A}) - \mu(\boldsymbol{A}\cap\boldsymbol{B}))/\mu(\boldsymbol{A}) \leq \varepsilon$, than the trait ***B*** is a **ε -covering** trait for the trait ***A***.
If $(\mu(\boldsymbol{B}) - \mu(\boldsymbol{A}\cap\boldsymbol{B}))/\mu(\boldsymbol{B}) \leq \varepsilon$, than the trait ***B*** is a **ε -subordinating** trait for the trait ***A***.
Here ε is a real number, $0 \leq \varepsilon < 1$.
Thus the group of ε-equivalent, ε-covering, or ε -subordinating traits is selected for the each characteristic trait obtained in the variant (the trait itself is included in the group). The voting of objects is performed in the experiment by using these groups of traits. For the object with the number *j* the votes for the first (U_j) and for the second (V_j) classes are calculated by the formulas

$$U_j= u_1^j/p_1+ u_2^j/p_2+ ... + u_{t1}^j/p_{t1},$$
$$V_j = v_1^j/s_1+ v_2^j/s_2+ ... + v_{t2}^j/s_{t2}.$$

Here t1 and t2 are the numbers of characteristic traits obtained in the variant for the first and the second class respectively, p_i (i = 1, 2, ... , t1) and s_i (i = 1, 2, ...,t2) are the numbers of traits in the groups obtained in the experiment, u_i^j (i = 1, 2, ..., t1) and v_i^j (i = 1, 2, ..., t2) are the numbers of traits from the corresponding groups, which the object has. The votes U_j and V_j are used for classification as the numbers of characteristic traits of the first and of the second class, which the object has.

3.3 Files

The program uses the file with the coding of objects and other information, which is created by the program **CODM**. The extension of the name of this file should be **.RAT**. The program may write information on conditions of each variant in a file on a disk (**PROFILE**) and read information from this file. The extension of the name of this file should be **.KAT** (for **CORA** algorithm) or **.HAT** (for **HAMMING** algorithm).

The selected characteristic traits may be written in the file on a disk to use them for classification of new objects. The extension of the name of this file is **.FAT**.

The program may create on a disk a new file with the coding of objects and identification of classes for objects, which corresponds to the classification obtained. The extension of the name of this file is **.RAT**. This file may be used in the control experiment, in which the obtained classes are used as a new learning set. When the problem of earthquake prone areas is considered this experiment is called Earthquake future" (Gelfand et al., 1976).

The program may create on a disk a ASCII format text file for printing with the name **PRA.PRI**. The number of columns in this file does not exceed 120.

3.4 Dialog

To input data the program uses the interface organized as a set of menus with accessible windows.

First menu

The first menu is selection of the algorithm. To select the algorithm you have to move the cursor to the proper window and press **ENTER**. If the algorithm **CORA-3** is selected and subclasses are defined in the file with the coding of objects then the algorithm **CLUSTERS** is applied.

By pressing **F10** it is possible to terminate the program.

Second menu

The second menu contains the window where the name of the input **PROFILE** has to be indicated. If the algorithm **CORA-3** or **CLUSTERS** is applied then the name has to have the extension **.KAT**. If the algorithm **HAMMING** is applied then the extension has to be .**HAT**. **Blank** means that an input **PROFILE** will not be used. The function key **F9** can be used to search the file in the current directory.

By pressing **F10** it is possible to terminate the program.

Third menu

The third menu contains the following windows.

#	Window Name	Function and Comments
1	File with coding	To specify ***the name of the file with the coding of objects***. This file is created by the program **CODM**. The name has to have extension .**RAT**. The function key **F9** can be used to search the file in the current directory.
2	Type	To set ***whether the coding of objects will be typed***. **ON** - the coding will be typed. **OFF** - the coding will not be typed.
3	Print	To set whether the printout will contain the coding of objects. **ON** - the coding of objects will be written in the file **PRA.PRI**. **OFF** - the coding of objects will not be written in the file **PRA.PRI**.
4	Control	To show ***consistency of input data*** already indicated. **OK** - data are consistent and if the cursor is moved to this window ***the fourth menu*** can be reached by pressing **ENTER**. The coding of objects will be typed if **ON** is in the window 2. **???** - data are not consistent.

By pressing **ESC** it is possible to return to ***the first menu***.
By pressing **F10** it is possible to terminate the program.

Fourth menu

The fourth menu contains the following windows.

#	Window Name	Function and Comments
1	Traits from	To specify ***how the traits will be obtained***. **Blank (learning)** means that traits will be obtained as a result of learning. **The name of the file** (with the extension **.FAT**) means that the traits will be read from this file. The function key F9 can be used to search the file in the current directory. This window is not used if the algorithm **HAMMING** is applied.
2	k1 k1t k2 k2t	To specify ***the values of selection and contradiction thresholds*** for traits of the first class (k1, k1t) and the second class (k2, k2t). This window is not used if the algorithm **HAMMING** is applied or traits are read from the file.
3	Objects of 1st class	To specify ***the composition of the learning sample of the first class.*** **ALL** - all objects of the first class will take part in the learning. **WITH EXCLUSIONS -** some objects of the first class may be excluded from the learning. The window may be opened by pressing **F2** to indicate these objects. This window is not used if the algorithm **CORA-3** or **CLUSTERS** is applied and at the same time traits are read from the file.
4	Objects of 2nd class	To specify ***the composition of the learning sample of the second class***. **ALL** - all objects of the second class will take part in the learning. **WITH EXCLUSIONS -** some objects of the second class may be excluded from the learning. The window may be opened by pressing **F2** to indicate these objects. This window is not used if the algorithm **CORA-3** or **CLUSTERS** is applied and at the same time traits are read from the file.
5	Components of vectors	To specify ***what components of vectors will be used in the learning***. **ALL** - all components of vectors will be used in the learning. **WITH EXCLUSIONS -** some components of vectors will not be used in the learning. The window may be opened by pressing **F2** to indicate these components. This window is not used if the algorithm **CORA-3** or **CLUSTERS** is applied and at the same time traits are read from the file.

6	Output Profile	To specify ***the name of the output PROFILE***. If the algorithm **CORA-3** or **CLUSTERS** is applied then the name has to have the extension **.KAT**. If the algorithm **HAMMING** is applied then the extension has to be **.HAT**. **Blank** means that output **PROFILE** will not be created.
7	Control	To show ***consistency of input data*** already indicated. **OK** - data are consistent and if the cursor is moved to this window the learning is executed by pressing **ENTER**. If the algorithm **CORA-3** or **CLUSTERS** is applied the numbers of occurrences of the components in the traits will be typed on the screen. ***The fifth menu*** will be reached after the learning. **???** - data are not consistent.

By pressing **ESC** it is possible to return to ***the third menu***.
By pressing **F10** it is possible to terminate the program.

Fifth menu

The fifth menu contains the following windows.

#	Window Name	Function and Comments
1	Delta	To specify ***the value of the parameter Δ***, which is used for classification of objects. If the algorithm **CORA-3** or **CLUSTERS** is applied then Δ is an integer number used as a threshold for a difference between numbers of characteristic traits of the first and second class which an object has. If the algorithm **HAMMING** is applied then Δ is a real number used as a threshold for a Hamming's distance between an object and the kernel. By pressing **F2** it is possible to show on the the screen the classification obtained with the specified value of Δ.
2	Definition of weights	To specify ***the values of weights*** for calculation of Hamming's distances between objects and the kernel. **ALL EQUAL TO 1.0** - all weights are set to be equal to 1. **CURRENT** - weights will have their current values. The window can be opened by pressing **F2** to change the values of weights. **AUTOMATIC** - weights are calculated by the formula: $w_i = 0.1 \text{ ENTIRE}\left(\frac{\lvert \alpha_1(i) - \alpha_2(i) \rvert}{\max_i \lvert \alpha_1(i) - \alpha_2(i) \rvert} + 0.5 \right).$ This window is not used if the algorithm **CORA-3** or **CLUSTERS** is applied.

3	Kernel and weights	To set ***whether the printout will contain the information about the kernel*** obtained in the variant and the weights used. **ON** - the information about the kernel and the weights will be written in the file **PRA.PRI**. **OFF** - the information about the kernel and the weights will not be written in the file **PRA.PRI**. By pressing **F2** it is possible to show on the screen the kernel and the weights. This window is not used if the algorithm **CORA-3** or **CLUSTERS** is applied.
4	Traits	To set ***whether the printout will contain the image of the traits*** obtained in the variant. **ON** - the image of the traits will be written in the file **PRA.PRI**. **OFF** - the image of the traits will not be written in the file **PRA.PRI**. By pressing **F2** it is possible to show on the screen the image of the traits. This window is not used if the algorithm **HAMMING** is applied.
5	Lattices	To set ***whether the printout will contain the lattices of the traits*** obtained in the variant. **ON** - the lattices of the traits will be written in the file **PRA.PRI**. **OFF** - the lattices of the traits will not be written in the file **PRA.PRI**. By pressing **F2** it is possible to show on the screen the lattices of the traits. This window is not used if the algorithm **HAMMING** is applied.
6	Table of voting	To set ***whether the printout will contain the 2-dimensional table of voting*** for the variant. **ON** - the table will be written in the file **PRA.PRI**. **OFF** - the table will not be written in the file **PRA.PRI**. By pressing **F2** it is possible to show on the screen the 2-dimensional table of voting. This window is not used if the algorithm **HAMMING** is applied.
7	Voting	To set ***whether the printout will contain the results of voting*** obtained in the variant. **ON** - the results of voting will be written in the file **PRA.PRI**. **OFF** - the results of voting will not be written in the file **PRA.PRI**. By pressing **F2** it is possible to show on the screen the results of voting obtained in the last variants.
8	Protocol	To set ***whether the printout will contain the conditions of the variant***. **ON** - the conditions of the variant will be written in the file **PRA.PRI**. **OFF** - the conditions of the variant will not be written in the file **PRA.PRI**. This window is not used if **ON** is at least in one of the windows **3-7** (in this case the conditions of the variant will be written in the file **PRA.PRI**).

9	File for traits	To specify ***the name of the file, in which the traits obtained will be written***. The name has to have the extension **.FAT**. **Blank** means that this file will not be created. This window is not used if the algorithm **HAMMING** is applied.
10	File for coding of classification	To specify ***the name of the file, in which the coding of objects and identification of classes for objects, which corresponds to the classification obtained, will be written***. The name has to have the extension **.RAT**. **Blank** means that this file will not be created.
11	Control experiment	To specify ***the control experiment, which will be executed*** in the variant. **NO** - no control experiments will be executed. **T-E OF OBJECTS** - test excluding of objects will be executed. **T-E OF FUNCTIONS** - test excluding of functions will be executed. **EQUIVALENT TRAITS** - the experiment “**Equivalent traits**” will be executed. The window may be opened by pressing **F2** to get into ***the sixth menu*** to specify the conditions of the experiment. This experiment is not executed if the algorithm **HAMMING** is applied.
12	Next variant	To specify ***whether some more variants will be executed***. **YES** - one more variant will be calculated. **NO** - no more variants will be calculated. This window is not used if ***a control experiment*** is executed (the content of the window **11** is other than **NO**).
13	Stop after subvariant	To set for a control experiment which consists of several subvariants ***whether the program will be interrupted after each subvariant***. **YES** - the program will be interrupted after each subvariant. **NO** - the program will be interrupted after all subvariants of the experiment are executed. This window is used only if the control experiment “**T-E of objects**” or “**T-E of functions**” is executed (**T-E OF OBJECTS** or **T-E OF FUNCTIONS** is in the window **11**).
14	Control	To show ***consistency of input data*** already indicated. **OK** - data are consistent and if the cursor is moved to this window by pressing **ENTER**, one of the following actions will be made: if one more variant is calculated (**YES** is in the window **12**) the fourth menu will be returned; if a control experiment is to be executed (the content of the window **11** is other than **NO**) it will be executed; if no new variants and control experiments are to be executed (**NO** is in the windows **11** and **12**) the program will be stopped. **???** - data are not consistent.

By pressing **ESC** it is possible to return to ***the fourth menu***.
By pressing **F10** it is possible to terminate the program.

Sixth menu

The sixth menu contains the following windows.

#	Window Name	Function and Comments
1	Traits to select	To specify ***what kind of traits will be selected in the experiment "Equivalent traits"***. **E-EQUIVALENT** - ε-equivalent traits for the characteristic traits obtained in the variant will be selected. **E-COVERING** - ε-covering traits for the characteristic traits obtained in the variant will be selected. **E-SUBORDINATING** - ε-subordinating traits for the characteristic traits obtained in the variant will be selected. **E-SUBORDINATING & E-COVERING** - both ε-covering and ε-subordinating traits for the characteristic traits obtained in the variant will be selected.
2	Value of E	To specify ***the value of the parameter ε***.
3	k1 k1t k2 k2t	To specify ***the values of selection and contradiction thresholds*** for selection of the traits of the first class (k1, k1t) and the second class (k2, k2t) in the experiment.
4	Objects of 1st class	To specify ***the composition of the learning sample of the first class*** for the experiment. **ALL** - all objects of the first class will be included. **WITH EXCLUSIONS** - some objects of the first class may be excluded from the learning. The window can be opened by pressing **F2** to indicate these objects.
5	Objects of 2nd class	To specify ***the composition of the learning sample of the second class*** for the experiment. **ALL** - all objects of the second class will be included. **WITH EXCLUSIONS** - some objects of the second class may be excluded from the learning. The window may be opened by pressing **F2** to indicate these objects.
6	Components of vectors	To specify ***which components of vectors will be used*** in the experiment. **ALL** - all components of vectors will be used. **WITH EXCLUSIONS** - some components of vectors will not be used. The window may be opened by pressing **F2** to indicate these components.

By pressing **ESC** it is possible to return to ***the fifth menu***.
By pressing **F10** it is possible to terminate the program.

3.5 Example of Printout

```
    vectors of class 1(D0)

               |   |
               |   |
               |   |
               |   |
              f|  g|
              2   4
  1.    M1  1 1|1 0|
  2.    M2  0 1|0 1|
  3.    M7  0 1|1 0|
  4.    M8  0 0|0 1|
  5.   M10  1 1|0 1|
   control: 2 4 2 3

    vectors of class 2(N0)

  1.    M3  0 0|1 0|
  2.    M5  1 1|1 0|
  3.    M9  1 1|1 0|
  4.   M14  0 0|1 0|
  5.   M15  1 1|0 1|
   control: 3 3 4 1

    vectors of class 3(X)

  1.    M4  0 0|0 1|
  2.    M6  1 1|1 0|
  3.   M11  0 1|1 0|
  4.   M12  0 1|1 0|
  5.   M13  0 0|1 0|
   control: 1 3 4 1


   1 variant

Algorithm: CORA-3
Data from file: test.rat
k1=2, k1t=1, k2=2, k2t=1
Objects of 1 class which do not take part in learning:
Objects of 2 class which do not take part in learning:
Numbers of components which are not used:
Mode of work: learning
Delta=0
Conditions of variant have been written in profile: test.kat
                          |   |
                          |   |
                          |   |
                          |   |
                         f|  g|
                       1   3
                       1 1 1 1


2 characteristic traits of class 1 (D-traits)

 1. 7; 4, 4, 4   3: 1     |  1|
 2. 1; 1, 1, 2   2: 0  0 1|   |
2 characteristic traits of class 2 (N-traits)

 1. 0; 2, 2, 2   1: 2    0|   |
 2. 6; 1, 1, 4   1: 2  1  |  0|


Lattices of traits (objects from learning set which have a trait
are marked by 1)

2 characteristic traits of class 1 (D-traits)


                       11111   11111
 1. 7; 4, 4, 4   3: 1  -1-11   ----1
 2. 1; 1, 1, 2   2: 0  -11--   -----

2 characteristic traits of class 2 (N-traits)


                       11111   11111
 1. 0; 2, 2, 2   1: 2  ---1-   1--1-
 2. 6; 1, 1, 4   1: 2  1----   -11--

   number   |
  of votes  | number of votes for 1 class
for 2 class |   0    |   1    |   2    |
       ----------------------------------
         1  | +   M1 | +   M8 |        |
            | -   M3 |     M4 |        |
            | -   M5 |        |        |
            | -   M9 |        |        |
            | -  M14 |        |        |
            |     M6 |        |        |
            |    M13 |        |        |
       ----------------------------------
         0  |        | +   M7 | +   M2 |
            |        | +  M10 |        |
            |        | -  M15 |        |
            |        |    M11 |        |
            |        |    M12 |        |
       ----------------------------------

                        delta: -5 -4 -3 -2 -1  0  1  2  3  4  5
numbers of objects in class D: 15 15 15 15 15  8  6  1  0  0  0





2 variant

Algorithm: HAMMING
Data from file: test.rat
Objects of 1 class which do not take part in learning:
Objects of 2 class which do not take part in learning:
Numbers of components which are not used:
Weights of components which are used:
   1.0, 1.0, 1.0, 1.0
Mode of work: learning, t-e of functions
Delta= 2.00
```

Conditions of variant have been written in profile: test.hat

names of functions		f		g
number of comp.	1	2	3	4
% of 1 in 1 cl.	43	71	43	57
% of 1 in 2 cl.	57	57	71	29
kernel	0	1	0	1
weights of comp.	1.0	1.0	1.0	1.0

2 variant (1 subvariant)

names of functions		f		g
number of comp.	1	2	3	4
% of 1 in 1 cl.			43	57
% of 1 in 2 cl.			71	29
kernel			0	1
weights of comp.			1.0	1.0

2 variant (2 subvariant)

names of functions		f		g
number of comp.	1	2	3	4
% of 1 in 1 cl.	43	71		
% of 1 in 2 cl.	57	57		
kernel	0	1		
weights of comp.	1.0	1.0		

```
   variant      1      2     2-1    2-2
    k1:k2     2: 2  - : -  - : -  - : -
   k1t:k2t    1: 1  - : -  - : -  - : -
    p1:p2     2: 2   2.00;  1.00;  1.00;
  del.;nDc    0;  8    8      5     15

    voting for vectors of class 1(D0)

  1.    M1    0: 1   3.00   2.00   1.00+
  2.    M2    2: 0+  0.00+  0.00+  0.00+
  3.    M7    1: 0+  2.00+  2.00   0.00+
  4.    M8    1: 1+  1.00+  0.00+  1.00+
5.   M10    1: 0+  1.00+  0.00+  1.00+

  voting for vectors of class 2(N0)

1.    M3    0: 1   3.00   2.00   1.00+
2.    M5    0: 1   3.00   2.00   1.00+
3.    M9    0: 1   3.00   2.00   1.00+
4.   M14    0: 1   3.00   2.00   1.00+
5.   M15    1: 0+  1.00+  0.00+  1.00+

  voting for vectors of class 3(X)

1.    M4    1: 1+  1.00+  0.00+  1.00+
2.    M6    0: 1   3.00   2.00   1.00+
3.   M11    1: 0+  2.00+  2.00   0.00+
4.   M12    1: 0+  2.00+  2.00   0.00+
5.   M13    0: 1   3.00   2.00   1.00+
```

3.6 Comments to the Example of Printout

The printout contains the following:

- The coding of the objects.
- The conditions of the first variant (protocol); the algorithm used is **CORA-3**.
- The images of the traits of the first and the second classes. For each trait the following information is printed in one line:
    - ✓ the number of the trait;
    - ✓ number $4\delta_1+ 2\delta_2+ \delta_3$, where δ_1, δ_2, and δ_3 are the values of three components of vectors, which form the trait;
    - ✓ the numbers of the components, which form the trait;
    - ✓ the number of objects from the first class, which have the trait;
    - ✓ after "**:**" number of objects from the second class, which have the trait;
    - ✓ "**image**" of the trait.
- The lattices of the traits. The beginning of the line, which corresponds to each trait is the same as in the case of the images of the traits. For each trait in the corresponding line the following information is also printed:
    - ✓ "**lattice**" of the trait for the objects from the first class (the *i*-th element of the lattice is "**1**", if the *i*-th object of the first class has the trait, otherwise it is "**-**");
    - ✓ "**lattice**" of the trait for the objects from the second class.

  Above the lattices for objects of the first class one line is printed: the *i*-th element of this line is "**1**", if the *i*-th object of the first class takes part in the learning, otherwise it is "**0**". Above the lattices for the objects of the second class the similar line is also printed; it shows what objects of the second class take part in the learning.
- The 2-dimensional table of voting for the variant. The names of the objects from the first class are marked by '+' and the objects of the second class - by "**-**".
- The conditions of the second variant (protocol); the algorithm used is **HAMMING**; the experiment **T-E of functions** is made.
- The kernels for the first variant and for the subvariants of the T-E of functions. The values of $\alpha_1(i)\cdot 100\%$, $\alpha_2(i)\cdot 100\%$ and weights are printed with the kernels.
- The results of voting. In the title the number of the variant and the number of the subvariant (if an experiment is made) are printed. If the algorithm **CORA-3** is applied then the values of thresholds for trait selection in this variant, numbers of the traits selected for the first (**p1**) and the second (**p2**) class, the value of the threshold Δ (**del.**) and

the number (**nDc**) of objects, which are classified as the objects of the first class are also printed in the title for each variant. If the algorithm **HAMMING** is applied then the value of Δ and the number **nDc** are printed in the last two lines of the title of the variant or the subvariant. If the algorithm **CORA-3** is applied then two columns are printed under the title: in the left column there are numbers of the traits of the first class, which the objects have; in the right column - numbers of the traits of the second class, which the objects have. If the algorithm **HAMMING** is applied then the Hamming's distances from the corresponding kernels to objects are printed under the title. If the object is classified as the object of the first class the results of voting or the value of the Hamming's distance are marked by "+". If some objects from the first or from the second classes do not take part in the learning in the variant or the subvariant then the results of voting or the values of the Hamming's distance for them are marked by "*".

REFERENCES

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.Kosobokov. 1981. On foundations of the pattern recognition results applied to earthquake-prone areas. *Proceedings of Ac. 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.

Sadovsky,M.A. (ed.), 1986. *Long-term Earthquake Prediction*. Moscow, IFZ AN SSSR – MSSSS (in Russian). English translation: Algorithms of Long-term Earthquakes' Prediction. CERESIS, Lima, Peru, 1986.