INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 586 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONE: 2240-1
CABLE: CENTRATOM - TELEX 460392 - I

H4.SMR/303 - 2

WORKSHOP
GLOBAL GEOPHYSICAL INFORMATICS WITH APPLICATIONS TO RESEARCH IN EARTHQUAKE PREDICTIONS AND REDUCTION OF SEISMIC RISK

(15 November - 16 December 1988)

THE PACKAGE OF PATTERN RECOGNITION PROGRAMS. USER'S GUIDE

A. A. SOLOVIEV

Institute of Physics of the Earth
Academy of Sciences of the U.S.S.R.
Bolshaya Gruzinskaya 10
123 242 Moscow
U.S.S.R.

PR: The package of programs - to apply algorithms of pattern recognition.

The authors used it mainly for recognition of earthquake-prone areas [1, 2] and the research in the problem of earthquake prediction [3]. However there is a wide variety of other applications (for example [4]).

Concerning applications of the package for earthquake prediction studies the authors have to emphasize that all methods of prediction, so far suggested, are yet insufficiently tested indeed. 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).

Initial data - objects of recognition specified as vectors with real or binary components.

All programs work in an interactive mode. The necessary input data are requested by consecutive questions flashed on display.

Following are:

(i) The general list of programs.

(ii) For each program:

- the purpose of the program;
- the list of questions of the dialogue with possible answers and the responce of the program to each answer;
- an example of printout.

General list of programs.

1. CODM Discretizes parameters and codes objects as binary vectors.

2. CORA Applies the pattern recognition algorithm "CORA-3" or its modification "subclasses" to vectors with binary components.

3. HAM Applies the pattern recognition algorithm "HAMMING's RULE" to vectors with binary components.

4. VOT Applies the recognition rule obtained by "CORA-3" algorithm to classify new objects by voting.

Before using the programs CORA and HAM, the user has to get acquainted with algorithms. Their descriptions can be found: for CORA - in [1] ; for HAM -in [2, 4].

WARNING. The estimation of significance of pattern recognition results requires numerical tests. Examples of logical tests can be found in [1], of statistical tests - in [2].

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 Department of Computational Geophysics, Institute of the Physics of the Earth, Ac. Sci. USSR, B. Gruzinskaya, 10, Moscow 123810, USSR.

Program CODM.

PURPOSE. The program is used for:

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

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

Histograms of parameters can be constructed separately for each class of objects. The program compares the histograms of a parameter for all classes. In particular, if the hystograms for the first and the second classes are sufficiently different, the parameter is promising for pattern recognition. To generate the histogram the thresholds for discretization of the parameter has to be specified. These thresholds can be entered in or calculated automatically; (p-1) thresholds divide the range of the parameter into p intervals. For automatic calculation of thresholds you have to specify p. The thresholds calculated automatically place almost the same number of objects into each interval.

There is also a possibility to construct a table ("table-histogram") with distribution of objects according to values of the parameter. The range of the parameter is divided by thresholds with a constant step, i.e. into equal intervals. For each interval the table lists the names of objects with the values of the parameter within this interval.

Objects can be coded as vectors with binary components (using the thresholds for discretization of parameters). There are two methods of coding: i-method and s-method. Let x(1) < x(2) < ... < x(p-1) be the thresholds for discretization, x(0) and x(p) be the bounds of the parameter's range. If the value of the parameter is x, and x(j-1) < x <= x(j), (1<= j <= p), then the parameter value is coded as

```
0  0 ...  0   1   0  ...  0   0  - in i-method,
0  0 ...  0   1   1  ...  1      - in s-method.
1  2 ... j-1  j  j+1 ... p-1  p
```

The program CODM reads information on values of parameters of objects from file (REMO FILE), created by the program PAR. It also can use information about the sets used for learning from the other file, created by the same program.

The program can store on a disk in PROFILE the following information: the name of the used REMO FILE; the class of each object; subclasses of the first class; the thresholds of discretization and the method of coding for each parameter.

This information can be used in further runs of the program.

The program can create a file (REM1 FILE) on a disk with coding of objects, identification of classes and subclasses, names of objects and parameters. This file is used in programs CORA and HAM.

To print histograms and coding the program can create a file on a disk. The name of this file is cod.pri. It can be printed by a suitable command of an operating system.

QUESTIONS OF THE DIALOGUE.

Question.	Answer and responce of the program.
1. profile?	y - the information from the PROFILE with the flashed name will be used; n - no information from any PROFILE will be used; e - goes to stop; the name of the PROFILE with information to be used (the extension of the name has to be .cod).
2. remO file?	y - the REMO FILE with the flashed name will be used; e - goes to stop; the name of the REMO FILE to be used (the extension of the name has to be .pat).
3. number of classes: N. new number?	new number of classes - the program repeats q. 3 with the flashed new number of classes (N); / - goes further.
4. number of classes?	number of classes.
5. file with information about classes?	y - information about learning sets from the PROFILE with the flashed name will be used; n - new information about learning sets will be entered from the terminal; the name of the file, created by the program PAR with information about learning sets (the extension of the name has to be .oat).

6. new class for the object: NAME? : the number of the class for the object with the flashed name (NAME) (number of classes + 1, if the object will not be used).

7. new classes for all objects? :
y - for each object the new class number has to be indicated;
n - the number of the class will be changed for no objects;
the name of the object which class will be changed.

8. next correction? :
the name of the object which class number will be changed;
n - the number of the class will be changed for no object.

9. do you want to work with subclasses in 1 class? :
y - subclasses in the first class will be formed;
n - there will be no subclasses in the first class.

10. number of subclasses? :
number of subclasses - the program repeats q. 10 with the flashed number of subclasses;
/ - goes further.

11. number of objects in N subclass? :
number of objects in the subclass with the flashed number (N) - the program repeats q. 11 with the flashed number of objects;
/ - goes further.

12. name of M object of N subclass? : the name of next object of the subclass with the flashed number (N).

13. do you want to change N subclass? :
y - the composition of the subclass with the flashed number (N) will be changed;
n - the subclass with the flashed number will not be changed.

14. do you want to change subclasses? :
y - all subclasses will be changed, goes to q. 10;
n - the subclasses flashed before q. 14 will not be changed.

15. with what parameter do you want to work? :
the name of the parameter for which discretization or coding will be determined;
n - the end of discretization and coding of parameters, goes to q. 22.

16. N parameter NAME? :
a - the thresholds of discretization for the parameter with the flashed number (N) and name (NAME) will be calculated automatically, goes to q. 17;
h - the thresholds of discretization for this parameter will be entered in, goes to q. 18;
t - the table-histogram will be constructed for this parameter, goes to q. 27;
c - this parameter will be coded by using its previous discretization, goes to q. 21;
n - goes to q. 15.

17. number of intervals? : number of intervals to divide the parameter's range.

18. values of thresholds? : the values of the thresholds for division of the parameter's range and / after the last value.

19. is distribution good? :
y - goes to q. 20;
n - goes to q. 16.

20. do you want to print distribution? :
y - information on this distribution will be written in the file for printing;
n - information on this distribution will not be written in the file for printing.

21. do you want to code this parameter? :
1 - the parameter will be coded by 1-method;
s - the parameter will be coded by s-method;
n - the parameter will not be coded.

22. any corrections? :
the name of parameter - goes to q. 16;
n - goes to q. 23.

23. do you want to write information in profile? : y - the PROFILE with the flashed name will be created;
: n - a new PROFILE will not be created;
: e - goes to stop;
: the name of the PROFILE to be created (the extension of the name has to be .cod).

24. do you want to print coding? : y - coding of the objects will be written in the file for printing;
: n - coding will be not writing in the file for printing.

25. do you want to create file with coding? : the name of REM1 FILE which will be created (the extension of the name has to be .rat);
: n - REM1 FILE will not be created.

26. do you want to type coding? : y - the coding of the objects will be flashed;
: n - the coding will not be flashed.

27. step of table? : the value of the step between thresholds of the table-histogram.

28. number of objects in tail? : the number of objects in the first and the last intervals of the table-histogram.

29. what do you want to do with this table? : n - goes to q. 16;
: p - the table will be written in the file for printing;
: v - construction of new table with new step or number of objects in tail, goes to q. 27.

EXAMPLE OF PRINTOUT.

```
 thresholds for  1 parameter:      f
obtained by a-method; rem0 file: ex25.pat
                 7.50       24.50
 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.%)


100 :
 90 :
 80 :
 70 :
 60 :  b
 50 :
 40 :a                        a c                         bc
 30 :
 20 :   c                                                a
 10 :
  0 ---------------------------b---------------------------
       7.500                                          24.50

 thresholds for  2 parameter:      g
obtained by h-method; rem0 file: ex25.pat
                 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 -------------------------------------------------------
                            2.000
```

```
2 parameter:      g; step=   0.50; igr= 0; rem0 file: ex26.pat
       ----------------------------------------
       :  1 class     :  2 class     :  3 class     :
 2.00----------------------------------------
       :     M1     M7 :     M3     M5 :     M6    M11 :
       :     M1     M7 :     M9    M14 :    M12    M13 :
 2.50----------------------------------------
       :     M2     M8 :    M15        :     M4        :
 3.00----------------------------------------
       :    M10        :               :               :
 4.00----------------------------------------

  thresholds for parameters with min and max
   1 parameter:      f; method of coding:s
    -5.000        7.600        24.50        50.00
   2 parameter:      g; method of coding:1
     1.800        2.000        4.001

                                 vectors of class 1


                   :  :
                   :  :
                   :  :
                   :  :
                  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

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 for recognition

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:
```

COMMENTS TO EXAMPLE OF PRINTOUT. The printout contains the following:

--The histogram for the parameter f. The thresholds were calculated automatically. Symbol "a" idicates points of the hystogram 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 parameter's range equals to 3.

--The histogram for the parameter g. Number of intervals to divide the parameter's range equals to 2. The threshold 2. was entered in.

--The table-histogram for the parameter g. Step of the table equals to 0.5 . Number of objects in tail (igr) equals to 0 .

--The coding of the objects. The parameter f was coded by s-method, the parameter g - by 1-method.

Values of parameters were the same as in the table in the exercise 25.

Program CORA.

PURPOSE. The program is used for application of the pattern recognition algorithm "CORA-3" [1] or its modification "SUBCLASSES" [1] to objects which are binary vectors.

During one run of the program several variants can be calculated. Variants can have the following differences: values of thresholds for characteristic traits of the first and the second classes; composition of the learning sets of the first and second classes; components of vectors which are used; mode of work ; kind of output.

The modes of work are:

- only learning;
- learning with test excluding (T-E) of objects;
- learning with T-E of parameters;

T-E of objects is additional calculation of n12 variants, where n12=max(n1,n2), n1 and n2 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 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 parameters is an additional calculation of m variants, where m is the number of parameters used in the variant. In each of these variants the parameters of the main variant are excluded one at a time.

If T-E of objects is made, the first is the main variant and then n12 T-E variants are computed.

If T-E of parameters is made, the first is the main variant and then m T-E variants are computed.

The program uses the file (REM1 FILE) with the coding of objects and other information which is created by the program CODM. The program can write information on conditions of each variant in a file on a disk (PROFILE) and read information from this file.

The selected characteristic traits can be written in the file on a disk to use them for classification of new objects by the program VOT.

The program can create on a disk a file for printing with the name cor.pri. This file can be printed by a suitable command of an operating system.

QUESTIONS OF THE DIALOGUE.

Question.	Answer and responce of the program.
1. profile?	y - information from the PROFILE with the flashed name will be used; n - information from any PROFILE will not be used; the name of the PROFILE which will be used (the extension of the name has to be .kat).

2. this file is absent; repeate?	y - goes to q. 1;
	c - goes to q. 3;
	n - goes to stop.

3. name of rem1 file?	y - the REM1 FILE with the flashed name will be used; n - goes to stop; the name of the REM1 FILE which will be used (the extension of the name have to be .rat).
4. max number of traits (mnt)?	the maximum number of traits which can be selected for each class (it is necessary for memory distribution).
5. do you want to change mnt?	y - goes to q. 4; n - goes to q. 6.

6. any change?

: y - goes to q. 7;
: n - goes to q. 17;
: e - goes to stop;
: w - runs the variant with conditions which were flashed before q. 6;
: 1 - goes to q. 7;
: 2 - goes to q. 9;
: 3 - goes to q. 11;
: 4 - goes to q. 12;
: 5 - goes to q. 13;
: 6 - goes to q. 14;
: 7 - goes to q. 15;
: 8 - goes to q. 16;
: 9 - goes to q. 17.

7. 1: thresholds for learning?

: values of selection and contradiction thresholds for traits of the first class (K1, K1t) and the second class (K2, K2t).

8. change learning sets of objects or parameters?

: y - goes to q. 9;
: n - goes to q. 12;
: w - runs the variant;
: e or an integer from 1 to 9 - see the answers to q. 6.

9. 2: numbers of objects of 1 class which will not take part in learning?

: 0/ - all objects of the first class will take part in the learning;
: the numbers of objects of the first class which will not take part in the learning in this variant, and / after the last number.

10. 2: numbers of objects of 2 class which will not take part in learning?

: 0/ - all objects of the second class will take part in the learning;
: the numbers of objects of the second class which will not take part in the learning in this variant, and / after the last number.

11. 3: numbers of components which will not be used?

: 0/ - all components of vectors will be used in this variant;
: the numbers of the components which will not be used in this variant, and / after the last number.

12. 4: out: coding?

: y - the output for this variant will contain the coding of objects;
: n - the output for this variant will not contain the coding;
: w - runs the variant;
: e or an integer from 1 to 9 - see the answers to q. 6.

13. 5: out: table of voting?

: y - the output for this variant will contain 2-dimensional table of voting;
: n - the output for this variant will not contain 2-dimensional table of voting;
: w - runs the variant;
: e or an integer from 1 to 9 - see the answers to q. 6.

14. 6: out: lattices of traits; traits?

: a1,a2 - if a1=y, the output for this variant will contain the lattices of the traits; if a1=n, the output will not contain the lattices of the traits; if a2=y, the output will contain the images of the traits; if a2=n, the output will not contain the images of the traits.

15. 7: mode: t-e of objects t-e of parameters?

: a1,a2 - if a1=y, T-E of objects will be made for this variant; if a1=n, T-E of objects will not be made; if a2=y, T-E of parameters will be made; if a2=n, T-E of parameters will not be made.

16. 8: type,print results?

: a1,a2 - if a1=y, the output for this variant will be flashed; if a1=n, the output will not be flashed; if a2=y, the output for this variant will be printed; if a2=n, the output will not be printed.

17. 9: profile?
: y - the conditions of this variant will be written in the PROFILE with the flashed name;
: n - the conditions of this variant will not be written in a PROFILE;
: the name of the PROFILE in which the conditions of this variant will be written (the extension of the name has to be .kat).
:

18. any corrections?
: y - goes to q. 7;
: n - runs the variant;
: w - runs the variant;
: e or an integer from 1 to 9 - see the answers to q. 6.
:

19. O. K.?
: d - goes to the voting; a threshold for classification (delta) can be chosen (the objects for which the difference of votes for the first and the second classes is greater or equal to delta are classified as the objects of the first class);
: y - goes to the voting; delta will be equal to 0;
: n - goes to redefinition of thresholds for traits selection (q. 7);
: e - this variant is ignored, goes to q. 23;
: w - the selected characteristic traits will be written in the file on a disk.
:COMMENT. This question is asked after selection of traits; before it the numbers of selected traits of the first class (p1) and of the second class (p2) are flashed. The number of components in binary vectors (l), the numbers of objects in the learning sets of the first (n1) and the second (n2) class and of the examinating objects (n3), the values of thresholds k1, k1t, k2, k2t, the value of mnt and the successional number of the variant (in) are also flashed.
:

20. do you want to write traits on disk?
: n - characteristic traits will not be written on disk;
: the name of the file for writing of characteristic traits (the extension of the name has to be .fat).
:

21. delta?
: the value of the threshold delta.
:COMMENT. questions 21 and 22 are asked only in the case, when the answer to q. 19 is d.
:

22. change delta?
: y - goes to q. 21;
: n - choice of delta is finished, goes further.
:

23. next variant?
: y - goes to q. 6;
: n - goes to stop after the output of voting results.
:

EXAMPLE OF PRINTOUT.

```
data from ex25.rat
max number of traits(mnt): 20
1: k1= 2,k1t= 1,k2= 2,k2t= 1
2: numbers of objects of 1 class, which will not take part in learning-
   numbers of objects of 2 class, which will not take part in learning-
3: numbers of components which will not be used-
4-6: output -4: coding; 5: table of voting; 6: traits and lattices of
                                                 traits;

7: mode of work: learning;
8: type-y; print-y
9: profile-ex25.kat
                                vectors of class 1


                 :   :
                 :   :
                 :   :
                 :   :
              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:

     contr: 2 4 2 3
```

```
                            vectors of class 2

 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:

    contr: 3 3 4 1

                            vectors for recognition

 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:

    contr: 1 3 4 1

     1 variant

                         2 traits of class 1

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

                         2 traits of class 2

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

```
                     f   g
                       2   4
                     1 1 1 1
class 1
      1.     3:   1     : 1:
      2.     2:   0  0 1:   :
class 2
      1.     1:   2    0:   :
      2.     1:   2  1  : 0:
   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:      :
       ----:---------------------

      15  15  15  15  15   8   6   1   0   0   0
  var. no.     1
   k1:k2      2: 2
  k1t:k2t     1: 1
    p1:p2     2: 2
 del.;nic    0;  6

                                voting (class 1)

1.      M1       0: 1
2.      M2       2: 0+
3.      M7       1: 0+
4.      M8       1: 1+
5.     M10       1: 0+
                                voting (class 2)

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

                                voting for new objects

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

COMMENTS TO EXAMPLE OF PRINTOUT. The printout contains the following:

--The conditions of the variant (answers to questions 7-17).

--The coding of the objects.

--The lattices of the traits of the first and the second classes. For each trait in one line the following information is printed:

- the number of the trait;
- number 4*d1+2*d2+d3, where d1, d2, d3 are the values of three components of vectors which form the trait;
- the numbers of the components which form the trait;
- number of objects from the first class which have the trait;
- after ":" number of objects from the second class which have the trait;

- the "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 "-");
- the lattice of the trait for the objects from the second class.

Above the lattices for objects of the first class two lines are printed; the i-th element of the first line is "1", if the i-th object of the first class takes part in the learning otherwise it is "0"; in the second line symbol "1" marks objects of the first class which are the last objects in subclasses. Above the lattices for the objects of the second class one line is printed; it shows what objects take part in the learning.

--The images of the traits.

--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 results of voting. In the title the number of the variant, the values of thresholds for traits selection in this variant, numbers of the traits selected for the first (p1) and the second (p2) class, the value of the the threshold delta and number (nic) of objects which are classified as the objects of the first class are printed for each variant. Under the title two columns for each variant are printed; 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 object is classified as the object of the first class the results of voting are marked by "+". If some object from the first or from the second classes did not take part in the learning the results of voting are marked by "*".

Program HAM.

PURPOSE. The program is used for application of the pattern recognition algorithm "HAMMING's RULE" [2, 4] to objects which are vectors with binary components.

The learning in this algorithm consists of calculation of a kernel. Two values c1(i) and c2(i) are calculated for each used component (i is the number of the component)

$$c1(i)=(n1(i)+1)/(n1+2)\ ;\ c2(i)=(n2(i)+1)/(n2+2)\ .$$

Here n1 is the number of objects of the first class in the learning, n1(i) shows how many of them posess 1 in the i-th component; n2 and n2(i) are similar numbers for objects of the second class. The kernel is the vector

$$k=< k(1),k(2),\dots,k(l) >,$$

$$k(i) = 1 \text{ if } c1(i)=>c2(i);\ k(i) = 0 \text{ if } c1(i)<c2(i)\ .$$

Here l is the number of used components.

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

$$r(j)=w(1)*abs(x(j,1)-k(1))+w(2)*abs(x(j,2)-k(2))+\dots+$$

$$+w(l)*abs(x(j,l)-k(l)).$$

Here j is the number of the object, w(i) is the given weight for the i-th used component which is a real number between 0 and 1 which is a multiple of 0.1, x(j,i) is the value of the i-th used component of the j-th vector.

The objects are classified as the objects of the first class if r(j) is less or equal to chosen threshold delta.

The organization of calculations in the program is the same as in the program CORA. The program reads the coding of objects and other necessary information from REM1 FILE, created by the program CODM. The program can write information on conditions of a variant in its PROFILE and read this information. The program can create on a disk a file with the name ham.pri for printing. This file can be printed by a suitable command of an operating system.

QUESTIONS OF THE DIALOGUE.

Question.	Answer and responce of the program.
1. profile?	y - information from the PROFILE with the flashed name will be read; n - no information from any PROFILE will be read; the name of the PROFILE which will be read (the extension of the name has to be .hat).

2. this file is absent; repeate?	y - goes to q. 1;
	c - goes to q. 3;
	n - goes to stop.

3. name of remi file?	y - the REM1 FILE with the flashed name will be used; n - goes to stop; the name of the REM1 FILE which will be used (the extension of the name has to be .rat).

4. any change?	y - goes to q. 5;
	n - goes to q. 12;
	e - goes to stop;
	w - runs the variant with conditions which are flashed before q. 4;
	1 - goes to q. 6;
	2 - goes to q. 8;
	3 - goes to q. 9;
	4 - goes to q. 10;
	5 - goes to q. 11;
	6 - goes to q. 12.

5. change learning sets of objects or parameters?	y - goes to q. 6;
	n - goes to q. 9;
	w - runs the variant;
	e or an integer from 1 to 6 - see the answers to q. 4.

6. 1: numbers of objects of 1 class which will not take part in learning?	0/ - all learning objects of the first class will take part in the learning; the numbers of objects of the first class which will not take part in the learning in this variant, and / after the last number.

7. 1: numbers of objects of 2 class which will not take part in learning?

: 0/ - all objects of the second class will take part in the learning;
: the numbers of objects of the second class which will not take part in the learning in this variant, and / after the last number.

8. 2: numbers of components which will not be used?

: 0/ - all components of vectors will be used in this variant;
: the numbers of the components which will be not used in this variant, and / after the last number.

9. 3: out: coding?

: y - the output for this variant will contain the coding of objects;
: n - the output for this variant will not contain the coding;
: w - runs the variant;
: e or an integer from 1 to 6 - see the answers to q. 4.

10. 4: mode: t-e of objects; t-e of parameters?

: a1,a2 - if a1=y, T-E of objects will be made for this variant; if a1=n, T-E of of objects will not be made; if a2=y, T-E of parameters will be made; if a2=n, T-E of parameters will not be made.

11. 5: type,print results?

: a1,a2 - if a1=y, the output for this variant will be flashed; if a1=n, the output will not be flashed; if a2=y, the output for this variant will be printed; if a2=n, the output will not be printed.

12. 6: profile?

: y - the conditions of this variant will be written in the PROFILE with the flashed name;
: n - the conditions of this variant will not be written in a PROFILE;
: the name of the PROFILE in which the conditions of this variant will be written (the extension of the name has to be .hat).

13. any corrections?

: y - goes to q. 5;
: n - runs the variant;
: w - runs the variant;
: e or an integer from 1 to 6 - see the answers to q. 4.

14. do you want to change weights?

: e - all weights will be equal to 1;
: a - weights will be calculated by the formula: w(i)=0.1*
entier(abs(c1(i)-c2(i))/(0.1
*max(abs(c1(i)-c2(i))))+0.5) ;
i
: h - new values of weights have to be punched in, goes to q. 15;
: n - the flashed values of weights will used in the voting.
:COMMENT. Before this question the values of c1(i)*100%, c2(i)*100%, the components of the kernel and old values of weights are flashed.

15. new values of weights for components used?

: the values of weights, but only for components used in this variant.

16. delta?

: the value of the threshold for classification - delta.

17. change delta?

: y - goes to q. 16;
: n - choice of delta is made, goes further.

18. next variant?

: y - goes to q. 19;
: n - goes to stop after output of the voting results.

19. number of question for correction?

: an integer from 1 to 6 - see the answers to q. 4.

EXAMPLE OF PRINTOUT.

```
data from ex25.rat
1: numbers of objects of 1th class which will not take part in learning-
   numbers of objects of 2th class which will not take part in learning-
2: numbers of components which will not be used-
3: output - coding
4: mode of work learning; t-e of parameters
5: type-y; print-y
6: profile-ex25.hat
                             vectors of class 1


               :   :
               :   :
               :   :
               :   :
            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:

     contr: 2 4 2 3
                             vectors of class 2

  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:

     contr: 3 3 4 1


                             vectors for recognition

  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:

     contr: 1 3 4 1
```

```
      1 variant

       names    :       :       :
        of      :       :       :
    parameters  :       :       :
                :       :       :
                : 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

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

      3 variant

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

```
 variant       1       2       3
   delta     2.00    1.00    1.00
    nic        8       5      15

                             voting (class 1)

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

                             voting (class 2)

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

                             voting for new objects

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

COMMENTS TO EXAMPLE OF PRINTOUT. The printout contains the following:
--The conditions of the first variant (answers to questions 6-12); the mode of work is learning and T-E of parameters.
--The coding of the objects.
--The kernels for the first variant and for the variants in the T-E of parameters. The values of c1(1)*100%, c2(1)*100% and weights are printed with the kernels.
--The voting results which contain the Hamming's distances from the corresponding kernels to objects. The title contains the number of the variant, the value of the threshold delta and number (nic) of the objects which are classified as the objects of the first class (distances for these objects are marked by "+"). The distances for objects from the first and second classes which do not take part in the learning for this variant are marked by "*".

Program VOT.

PURPOSE. The program is used for voting, applied to new objects. Voting is made by characteristic traits obtained by program CORA and written on a disk.

The program uses the file (REM1 FILE) with the coding of new objects and other information which is created by the program CODM. It also uses the file created by the program CORA with characteristic traits.

The program can create on a disk a file for printing with the name vot.pri. This file can be printed by a suitable command of an operating system.

QUESTIONS OF THE DIALOGUE.

Question.	Answer and responce of the program.
1. type, print results?	a1, a2 - if a1=y, the voting will be flashed; if a1=n, the voting will not be flashed; if a2=y the voting will be written in the file for printing (vot.pri); if a2=n the voting will not be written in the file for printing.
2. name of file with traits?	e - goes to stop; the name of the file created by the program CORA with characteristic traits (the extension of the name has to be .fat).
3. name of file with objects?	e - goes to stop; the name of the REM1 FILE with coding of objects to be voted (the extension of the name has to be .rat).
4. numbers of variants, which traits you want to use for voting?	the list of sequantial numbers of sets of charateristic traits, which will be used for voting with / after the last number (the list has to contain not more than 18 numbers); / - goes to stop.

EXAMPLE OF PRINTOUT.

```
 voting of objects from file: ex25.rat
features from file: ex25.fat
var. no.      1
 k1:k2       2: 2
k1t:k2t      1: 1
 p1:p2       2: 2

                                 voting (class 1)

 1.     M1   0: 1
 2.     M2   2: 0
 3.     M7   1: 0
 4.     M8   1: 1
 5.    M10   1: 0

                                 voting (class 2)

 6.     M3   0: 1
 7.     M5   0: 1
 8.     M9   0: 1
 9.    M14   0: 1
10.    M15   1: 0

                                 voting for new objects

11.     M4   1: 1
12.     M6   0: 1
13.    M11   1: 0
14.    M12   1: 0
15.    M13   0: 1
```

COMMENTS TO EXAMPLE OF PRINTOUT. The coding of objects and the traits are the same as in the example of printout of the program CORA represented above.

References.

1. Gelfand, I. M., Guberman, Sh. A., Keilis-Borok, V. I., Knopoff, L., Press, F., Ranzman, I. Ya., Rotwain, I. M. and Sadovsky, A. M., 1976. Pattern recognition applied to earthquake epicenters in California. Phys. Earth Planet. Inter., 11: 227-283.
2. Gvishiani, A. D., Kosobokov, V. G., 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.
3. Gabrielov, A. M., Dmitrieva, O. E., Keilis-Borok, V. I. et al. Algorithms of long-term earthquakes' prediction. Preprint. Institute of the Physics of the Earth Ac. Sci. USSR. Moscow, 1986.
4. Keilis-Borok, V. I., Lichtman, A. J., 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., v. 78, 11: 7230-7234.