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

26 September - 8 October 2011

Exercises on Pattern Recognition

I. WRITTEN EXERCISES

1.1 ONE-DIMENSIONAL DISTRIBUTIONS

Consider a function x varying in interval $[x_0, x_T]$. Two sets of values of x are given: one for objects of the first class (**D**-objects) and another for objects of the second class (**N**-objects). Is this function useful for the discrimination between *D* and *N*? To answer this question, one may compare the distributions (histograms) of the values of x for **D**- and **N**-objects. Here distribution is a table of numbers n_i, where n_i is the number of values of x from the interval $(x_0 + (i\text{-}1)\Delta x, x_0 + i\Delta x]$, $i = 0, 1, ..., L$, and Δx is a numerical parameter.

Exercise 1: The values of two functions P1 and P2 for 10 **D**-objects and for 10 **N**-objects are given in Table 1. Find n_i for these functions separately for **D**- and **N**-objects. Let x_0 be equal to the minimal observed value of a function, for both **D**- and **N**-objects, and take $\Delta P_1 = 1$ and $\Delta P_2 = 10$.

TABLE 1

	P1	P2
D1	5.8	81
D2	7.4	112
D3	8.6	93
D4	8.5	108
D5	6.6	154
D6	9.3	97
D7	7.4	112
D8	6.7	103
D9	4.1	132
D10	7.2	100

	P1	P2
N1	6.3	98
N2	2.2	82
N3	2.9	129
N4	5.8	104
N5	1.4	71
N6	3.3	68
N7	4.5	82
N8	0.6	65
N9	3.2	77
N10	2.7	96

1.2 INFORMATIVE FUNCTIONS

Consider two distributions of values of the same function x: one distribution for **D**- and another for **N**-objects. Function x is informative for discrimination between **D**- and **N**-objects if the difference between these distributions is sufficiently large.

Let us denote: $P_x(\varepsilon,\Delta) = (1 - \varepsilon)n_D(\Delta)/n_D - \varepsilon n_N(\Delta)/n_N$. Here $n_D(\Delta)/n_D$ and $n_N(\Delta)/n_N$ are empirical cumulative distribution functions of x for **D**- and **N**-objects. In other words, n_D is the total number **D**-objects, for which the values of x are determined, $n_D(\Delta)$ is the number of these objects having $x > \Delta$; n_N and $n_N(\Delta)$ are the corresponding numbers for **N**-objects. The value of Δ varies within the same limits as x, and $0 \le \varepsilon \le 1$ determines the relative costs of failure to predict and false alarm ($\varepsilon = 1/2$, if the costs are equal).

How informative is x may be characterized by the maximal difference between these distribution functions, $\max_{\Delta} |P_x(1/2,\Delta)|$. Function x is the more informative, the nearer the absolute value of $P_x(1/2,\Delta)$ is to 1/2 at some suitable value of Δ.

Exercise 2: For x = P1 and P2 from Table 1, find $P_x(1/2,\Delta)$ at $\Delta = 7$ and 99, respectively.

Exercise 3: For the same functions find which Δ maximizes $|P_x(1/2,\Delta)|$. Which function (P1 or P2) is more informative judging by $\max_{\Delta} |P_x(1/2,\Delta)|$?

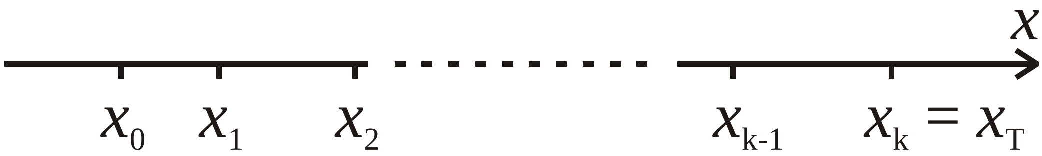
Exercise 4: For P1 find the value $\Delta(\varepsilon)$ maximizing $|P_x(\varepsilon,\Delta)|$: *a*) $\varepsilon = 0.25$, *b*) $\varepsilon = 0.75$.

Exercise 5: Make the previous ***Exercise 4*** for P2.

1.3 DISCRETIZATION

The used learning samples of the first and second classes and the set of the objects which are not used in the learning will be denote by D_0, N_0, and X respectively.

The values of each function x lie within certain range (x_0, x_T). We divide this range into k intervals by points x_i, i = 1, 2, ..., k-1. The value of x belongs to the i-th interval if $x_{i-1} < x \leq x_i$.



In a process of discretization we substitute the exact value of the function by the interval, which contains this value. Usually we divide the range into two intervals ("***large***" and "***small***" values) or into three intervals ("***large***", "***medium***" and "***small***" values). Our purpose is to find such intervals where moments of one class occur more often than moments of another class.

Objective (automatic) discretization: Each interval has about equal number of all the objects together (or of the moments from D_0 and N_0 together).

Exercise 6: The values of two functions, ***f*** and ***g***, are given in the Table 2 below. They are observed on D_0, N_0 and X. Find the intervals for objective discretization. For ***f*** take k = 3 and all objects. For ***g*** take k = 2 and objects from D_0 and N_0 only.

TABLE 2

# of moment	Class of of moment	Values of functions	
		f	***g***
1	D_0	0.	2
2	D_0	17.5	3
3	N_0	50.	2
4	X	35.	3
5	N_0	0.	2
6	X	0.	2
7	D_0	15.	2
8	D_0	25.	3
9	N_0	0.	2
10	D_0	0.	4
11	X	15.	2
12	X	24.	2
13	X	27.5	2
14	N_0	30.	2
15	N_0	0.	3

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

1. Denote P_i^D to be the number of objects from D_0 within the i-th interval of x, in percentage of total number of objects in D_0, and P_i^N to be the similar number for objects from N_0.

Let $P_{\max} = \max\limits_{1 \le i \le k} \left| P_i^D - P_i^N \right|$.

In other words, P_i^D and P_i^N are empirical histograms of the value of function for objects from D_0 and N_0; $P_{\max}$ is the maximal difference of these histograms.

The larger is $P_{\max}$, the more informative is the function. Functions for which $P_{\max} < 20\%$ are usually excluded.

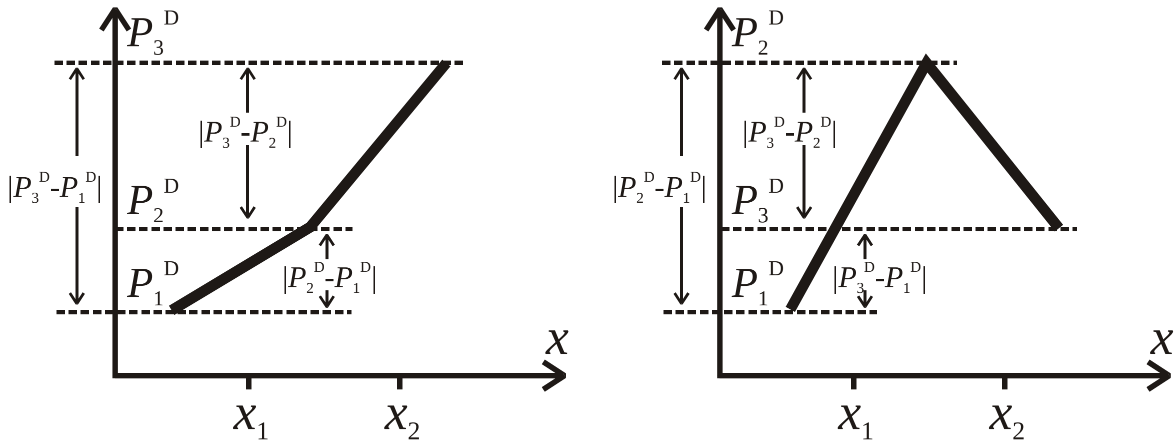
2. Let $k_j = 3$. Let us denote:

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

If P_i^D changes monotonously with i, then $M_D = 1$; the larger is M_D, more jerky is P_i^D. This is clear from the figures below. Similar statements are true for M_N, P_i^N.

$|P_2^D-P_1^D|+|P_3^D-P_2^D|=|P_3^D-P_1^D|$, $M_D=1$, P_i^D changes monotonously

$|P_2^D-P_1^D|-|P_3^D-P_2^D|=|P_3^D-P_1^D|$, $M_D>1$, P_i^D does not change monotonously



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

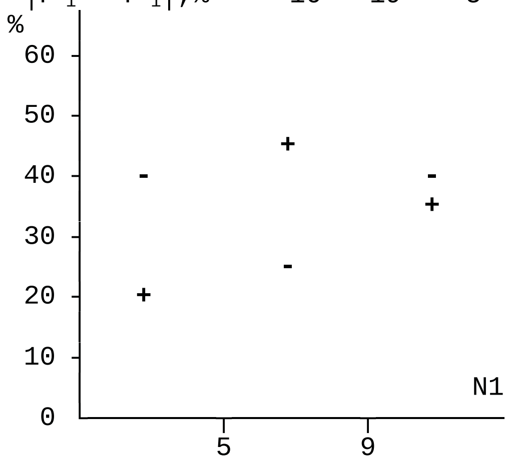
3. Samples D_0 and N_0 are often marginally small, so that their observed difference may be random. Therefore the relation between functions P_i^D and P_i^N after discretization should be not absurd from considered problem point of view, though they may be unexpected indeed.

Exercise 7: Count P_i^D and P_i^N for functions ***f*** and ***g*** from the Table 2 above. For ***f*** take $k = 3$, $f_1 = 5$, $f_2 = 30$; for ***g*** take $k = 2$, $g_1 = 2$.

Exercise 8: Figure below shows the values of P_i^D and P_i^N for seven functions. Arrange these functions in order of decreasing information, which they carry. What functions would you like to exclude?

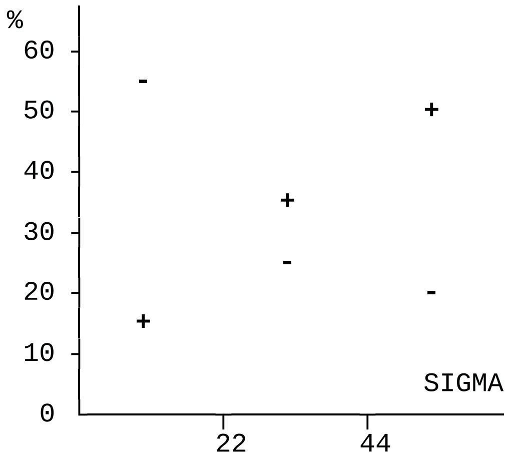
1. N1

P^D_i, %	22	44	33
P^N_i, %	38	25	38
$\lvert P^D_i - P^N_i \rvert$, %	16	19	5



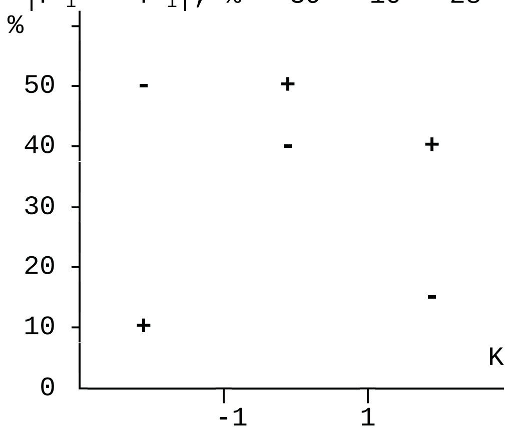
2. SIGMA

P^D_i, %	15	37	48
P^N_i, %	54	25	21
$\lvert P^D_i - P^N_i \rvert$, %	39	12	27



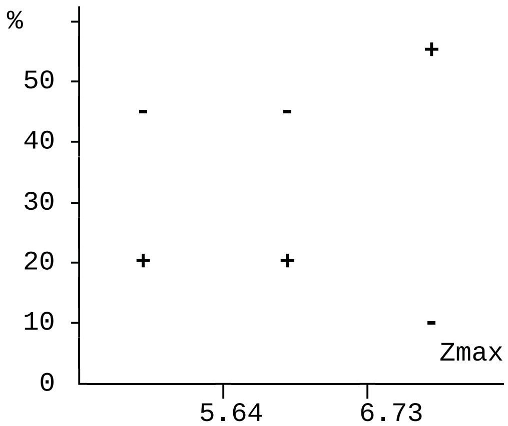
3. K

P^D_i, %	11	48	41
P^N_i, %	50	38	13
$\lvert P^D_i - P^N_i \rvert$, %	39	10	28



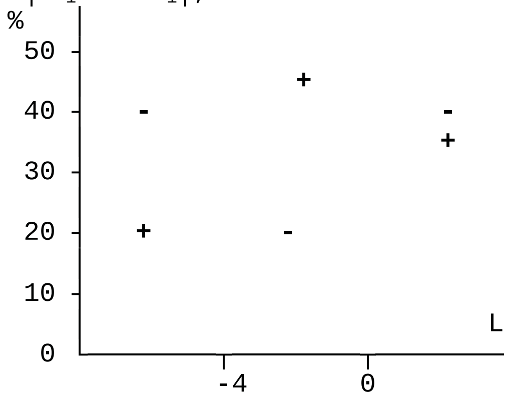
4. Zmax

P^D_i, %	22	22	56
P^N_i, %	46	46	8
$\lvert P^D_i - P^N_i \rvert$, %	24	24	48



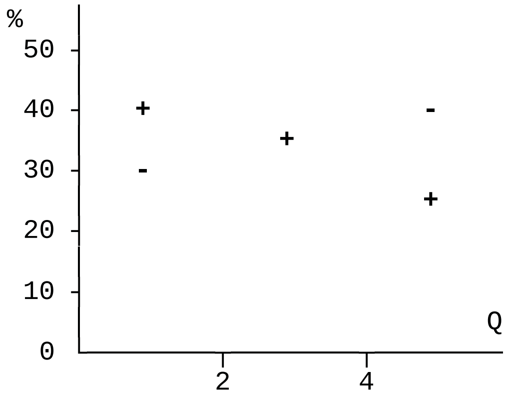
5. L

P^D_i , %	22	44	33
P^N_i , %	42	21	38
$\lvert P^D_i - P^N_i \rvert$, %	20	23	5



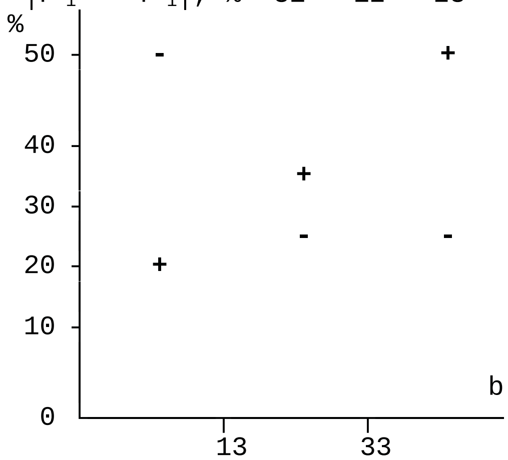
6. Q

P^D_i , %	41	33	26
P^N_i , %	29	33	38
$\lvert P^D_i - P^N_i \rvert$, %	12	0	12



7. Bmax

P^D_i, %	19	33	48
P^N_i , %	50	25	25
$\lvert P^D_i - P^N_i \rvert$, %	31	12	23



1.4 CODING

The value of the function belongs to the *l*-th interval, if $x_{l-1} < x \le x_l$. In this case two ways of coding are the following:

$i =$	1	2	...	l-1	l	l+1	...	k-1	k	
I-coding	0	0	...	0	1	0	...	0	0	(k digits),
S-coding	0	0	...	0	1	1	...	1		(k-1 digits).

The coding correspond to simple answers to the following questions about the value of *x*:

I - coding: Does *x* belong to $x_{l-1} < x \le x_l$? (1 - yes, 0 - no),
S - coding: Is $x \le x_l$? (1 - yes, 0 - no).

Exercise 9: The following Table shows the values of three functions: ***a***, ***b***, ***c***.

# of object	Values of functions		
	a	***b***	***c***
1	-10	3	2.5
2	5	7	-4
3	-2	6	-3
4	0	17	10
5	4	2	1

Discretization should be made as follows:

Function	*k*	x_l
a	2	0.5
b	3	4.5, 10
c	3	0, 5

Write the code of each object:
1) with ***I***-coding for function ***a***, ***S***-coding for functions ***b*** and ***c***;
2) with ***S***-coding for ***a***, ***I***-coding for ***b*** and ***S***-coding for ***c***.

1.5 DEFINITION OF THE TRAIT IN ALGORITHM CORA-3

The trait is represented by the 3*2 matrix

$$\mathbf{A} = \begin{Vmatrix} i_1 & i_2 & i_3 \\ \delta_1 & \delta_2 & \delta_3 \end{Vmatrix}$$

where i_1, i_2, i_3, are integers, $1 \le i_1 \le i_2 \le i_3 \le L$, L is the length of the binary code of the objects, δ_j is 0 or 1.

Object, which is the binary vector $\omega^i = (\omega_1^i, \omega_2^i, ..., \omega_L^i)$, has the trait **A** if

$$\omega_{i_1}^i = \delta_1, \quad \omega_{i_2}^i = \delta_2, \quad \omega_{i_3}^i = \delta_3.$$

For example if an object has the trait

$$\mathbf{A} = \begin{Vmatrix} 1 & 3 & 4 \\ 0 & 1 & 0 \end{Vmatrix}$$

then it means that the first and the fourth digits in the code of the object are 0 and the third digit is 1, and if an object has the trait

$$\mathbf{A} = \begin{Vmatrix} 2 & 2 & 2 \\ 0 & 0 & 0 \end{Vmatrix}$$

then it means that the second digit in the code of the object is 0.

Exercise 10: Consider objects (0 1 0 1 1) and (1 1 0 1 1). Find, whether they have the trait

$$\mathbf{A} = \begin{Vmatrix} 1 & 2 & 4 \\ 0 & 1 & 1 \end{Vmatrix}$$

===

1.6. CHARACTERISTIC TRAITS

Denote by $K(D_0, \mathbf{A})$ the number of objects of the set D_0, which have the trait **A**, and by $K(N_0, \mathbf{A})$ the number of objects of set N_0, which have the trait A.

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

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

The trait A is a characteristic trait of class N if

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

Here k_1, k_2, $\bar{k}_1$, and $\bar{k}_2$ are parameters of the algorithm. Characteristic traits of the first and second classes are also called D-traits and N-traits respectively.

Exercise 11: Given are the objects of two learning set:

Set D_0	Set N_0
1 1 1 0	0 0 1 0
0 1 1 0	0 1 1 1
1 1 1 1	0 0 0 0
1 1 0 0	0 0 0 1
1 1 0 1	0 1 0 0
	1 0 0 0

Find all D- and N-traits for $k_1 = 3$, $\bar{k}_1 = 1$, $k_2 = 3$, and $\bar{k}_2 = 0$.

1.7 EQUIVALENT AND WEAKER TRAITS

Consider two D-traits.
Denote:

S_1 is the subset of objects of set D_0, which have the first trait,
S_2 is the subset of objects of set D_0, which have the second trait.

The traits are ***equivalent*** if S_1 and S_2 coincide. The first trait is ***weaker*** than the second if S_1 is a strict subset of S_2. Definition of equivalent and weaker traits for N-traits is similar but D_0 is replaced by N_0.

Exercise 12: Consider all characteristic traits obtained in ***Exercise 11*** from Section 1.6. Eliminate all weaker traits. Leave only one trait from each group of equivalent traits.

==

1.7 CHARACTERISTIC TRAITS FOR SUBCLASSES (ALGORITHM "CLUSTERS")

In some problems the learning set D_0 consists of subclasses and it is known that in each subclass there is at least one object of the first class. Other objects of subclasses may belong to the second class. In this case the algorithm **CLUSTERS**, which is the modification of the algorithm **CORA-3** is applied.

Characteristic traits of the first class are defined in the algorithm **CLUSTERS** as follows.
A subclass has a trait if some object from it has this trait. Denote by $K^S(D_0, \mathbf{A})$ the number of subclasses, which have the trait **A**. **A** is a characteristic trait of the first class (D-trait) if

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

The definition of characteristic traits of the second class in the algorithm **CLUSTERS** is the same as in the algorithm **CORA-3** (see Section 1.6).

Exercise 13: Let objects of set D_0 given in **Exercise 11** of Section 1.6 divided into subclasses as follows.

```
Set D0   ––  1st subclass ––––
             1 1 1 0
             0 1 1 0
         ––  2nd subclass ––––
             1 1 1 1
         ––  3rd subclass ––––
             1 1 0 0
             1 1 0 1
```

Find subclasses, which have the following traits:

$$\mathbf{A} = \begin{Vmatrix} 1 & 1 & 4 \\ 0 & 0 & 0 \end{Vmatrix}, \quad \mathbf{B} = \begin{Vmatrix} 2 & 3 & 4 \\ 1 & 0 & 0 \end{Vmatrix}.$$

Exercise 14: Take the set D_0 given above and the set N_0 from ***Exercise 11*** of Section 1.6 and find all D-traits for $k_1 = 2$, $\bar{k}_1 = 1$.

1.9 EQUIVALENT AND WEAKER TRAITS FOR SUBCLASSES (ALGORITHM CLUSTERS)

Consider two *D*-traits.
Denote:

S^s_1 as the set of subclasses, which has the first trait,

S^s_2 is the set of subclasses, which has the second trait.

The traits are *S*-equivalent if S^s_1 coincide with S^s_2. The first trait is *S*-weaker than the second if $S^s_1$is a strict subset of S^s_2.

Exercise 15: Consider all characteristic traits from ***Exercise 14*** of Section 1.7. Eliminate all *S*-weaker traits. Leave only one trait from each group of *S*-equivalent traits.

==

1.10 VOTING AND RECOGNITION

Each object has some number n_D of *D*-traits and some number n_N of *N*-traits. The object is recognized as:

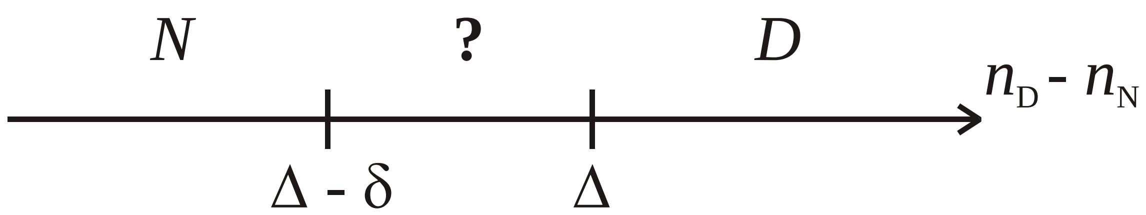
-object of the first class (*D*-object) if $n_D - n_N \geq \Delta$,

- object of the second class (*N*-object) if $n_D - n_N < \Delta$.

Here Δ is a parameter of the algorithm.

Exercise 16: Consider the characteristic traits left in ***Exercise 12*** of Section 1.7 after elimination of equivalent and weaker ones. Divide the objects from ***Exercise 11*** of Section 1.6 into classes *D* and *N* assuming $\Delta = 0$.

Note: It may be more reliable to assign to *N* only the object with $n_D - n_N \leq \Delta - \delta$, and to leave unassigned the objects with $\Delta > n_D - n_N > \Delta - \delta$.



1.11 HAMMING ALGORITHM: KERNEL OF THE FIRST CLASS

Determination of the kernel

Each object is a binary vector, as in previous exercises. Kernel is a binary vector of the same length; each component of this vector is "typical" for the first class. To define the kernel two values $\alpha_1(i)$ and $\alpha_2(i)$ are calculated for each used component (i is the number of the component)

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

Here N_1 is the number of objects used in the learning sample of the first class, $n_1(i)$ 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 the learning sample 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.

Exercise 17: Find the kernel for the following learning material:

Set D_0	Set N_0
0 1 1 0 1	1 0 1 1 0
1 0 0 1 0	1 1 0 0 1
0 1 0 0 1	1 0 1 0 0
0 1 0 1 1	0 0 0 0 0
	1 0 0 0 0

Note: It may be more reliable to eliminate the components, for which $|\alpha_1(i) - \alpha_2(i)| < \varepsilon$, where ε is a small constant.

1.12 HAMMING ALGORITHM: VOTING AND RECOGNITION

Hamming's distance from an object to the kernel of class D is

$$r = \sum_{i=1}^{L} w_i |\omega_i - k_i|.$$

Here w_i are the weights of components.

An object is recognized as an object of the first class (D-object), if $r \leq R$ or as an object of the second class (N-object), if $r > R$.

Exercise 18: Compute the Hamming's distance between the kernel and all objects from ***Exercise 16*** of Section 1.11 with all $w_i = 1$.

Find the minimal value of R, which will assign to the first class all objects of the learning sample D_0. With this R divide the objects of the learning sample N_0 into D- and N-objects.

II. ANSWERS FOR WRITTEN EXERCISES

Exercise 1:

P1

i	0	1	2	3	4	5	6	7	8	9
D	-	-	-	-	1	-	2	4	2	1
N	1	1	1	4	1	-	2	-	-	-
$x_0 + (i\text{-}1)\Delta x$	-0.4	0.6	1.6	2.6	3.6	4.6	5.6	6.6	7.6	8.6

P2

i	0	1	2	3	4	5	6	7	8	9
D	-	-	1	1	3	3	-	1	-	1
N	1	2	3	-	3	-	-	1	-	-
$x_0 + (i\text{-}1)\Delta x$	55	65	75	85	95	105	115	125	135	145

Exercise 2: For x = P1 $P_x(1/2, 7) = 0.3$. For x = P2 $P_x(1/2, 99) = 0.25$.

Exercise 3: For x = P1 max $|P_x(1/2,\Delta)| = 0.4$ at $6.3 \le \Delta < 6.6$.
For x = P2 max $|P_x(1/2,\Delta)| = 0.25$ at $82 \le \Delta < 93$ or $96 \le \Delta < 97$ or $98 \le \Delta < 100$.
P1 is more informative.

Exercise 4: For P1: *a*) $3.3 \le \Delta(0.25) < 4.1$; *b*) $6.3 \le \Delta(0.75) < 6.6$

Exercise 5: For P2: *a*) $77 \le \Delta(0.25) < 81$; *b*) $104 \le \Delta(0.75) < 108$ or $129 \le \Delta(0.75) < 132$.

Exercise 6: For $\boldsymbol{f}$: $f_1 = 0$, $24 \le f_2 < 25$; for $\boldsymbol{g}$: $2 \le g_1 < 3$.

Exercise 7: For $\boldsymbol{f}$: $P_1^D = 40\%$, $P_2^D = 60\%$, $P_3^D = 0\%$, $P_1^N = 60\%$, $P_2^N = 20\%$, $P_3^N = 20\%$;
for $\boldsymbol{g}$: $P_1^D = 40\%$, $P_2^D = 60\%$, $P_1^N = 80\%$, $P_2^N = 20\%$.

Exercise 8: Zmax, SIGMA, K, Bmax, L, N1, Q. It is reasonable to exclude functions L, N1, Q.

Exercise 9:

1) 1 0 1 1 0 1	2) 1 1 0 0 0 1
0 1 0 1 1 1	0 0 1 0 1 1
1 0 0 1 1 1	1 0 1 0 1 1
1 0 0 0 0 0	1 0 0 1 0 0
0 1 1 1 0 1	0 1 0 0 0 1

Exercise 10: (0 1 0 1 1) has, (1 1 0 1 1) has not.

Exercise 11:

D-traits: $\begin{Vmatrix}1&1&1\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&1&2\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&2&2\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&2&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&2&4\\1&1&0\end{Vmatrix}$, $\begin{Vmatrix}2&3&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&4&4\\1&0&0\end{Vmatrix}$

N-traits: $\begin{Vmatrix}1&1&2\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}1&1&3\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}1&2&2\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}1&3&3\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&2&2\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&2&3\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&2&4\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&3&3\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&4&4\\0&0&0\end{Vmatrix}$

Exercise 12:

D-traits: $\begin{Vmatrix}1&1&1\\1&1&1\end{Vmatrix}$ or $\begin{Vmatrix}1&1&2\\1&1&1\end{Vmatrix}$ or $\begin{Vmatrix}1&2&2\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&2&3\\1&1&1\end{Vmatrix}$ or $\begin{Vmatrix}2&3&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&2&4\\1&1&0\end{Vmatrix}$ or $\begin{Vmatrix}2&4&4\\1&0&0\end{Vmatrix}$

N-traits: $\begin{Vmatrix}1&1&3\\0&0&0\end{Vmatrix}$ or $\begin{Vmatrix}1&3&3\\0&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&2&2\\0&0&0\end{Vmatrix}$

Exercise 13: The 1st subclass has trait **A** and the 3rd has trait **B**.

Exercise 14:

$\begin{Vmatrix}1&1&1\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&1&2\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&1&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&1&4\\1&1&0\end{Vmatrix}$, $\begin{Vmatrix}1&1&4\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&2&2\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&2&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&2&4\\1&1&0\end{Vmatrix}$, $\begin{Vmatrix}1&2&4\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}1&3&3\\1&1&1\end{Vmatrix}$,

$\begin{Vmatrix}1&4&4\\1&0&0\end{Vmatrix}$, $\begin{Vmatrix}1&4&4\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&2&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&2&4\\1&1&0\end{Vmatrix}$, $\begin{Vmatrix}2&2&4\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&3&3\\1&1&1\end{Vmatrix}$, $\begin{Vmatrix}2&4&4\\1&0&0\end{Vmatrix}$, $\begin{Vmatrix}2&4&4\\1&1&1\end{Vmatrix}$

Exercise 15:

$\begin{Vmatrix}1&1&1\\1&1&1\end{Vmatrix}$ or $\begin{Vmatrix}1&1&2\\1&1&1\end{Vmatrix}$ or $\begin{Vmatrix}1&2&2\\1&1&1\end{Vmatrix}$

Exercise 16:

$n_{\mathrm{D}} : n_{\mathrm{N}}$

D_0

3 : 0 *D*
2 : 0 *D*
2 : 0 *D*
2 : 0 *D*
1 : 0 *D*

N_0

0 : 1 *N*
1 : 0 *D*
0 : 2 *N*
0 : 2 *N*
1 : 1 *D*
1 : 1 *D* or 0 : 1 *N*

Exercise 17: **k** = (0, 1, 0, 1, 1)

Exercise 18:

r

D_0

2 D
3 D
1 D
0 D

N_0

4 N
2 D
5 N
3 D
4 N

$R = 3$

III. COMPUTER EXERCISES WITH THE PROGRAMS CODM, CODMF, AND PRAL

EXERCISE 1

```
Task: To make and to print table-histograms for functions P1 and P2
      (Table 1 from Section 1.1 of the written exercises). Compare the
      obtained tables with the result of Exercise 1 of Section 1.1 of
      the written exercises.
Program to use: CODM.
Input files: profile - EXE1.COD,
             file with values of functions - EXE1.PAT.
Input data: mode of work - discretization,
            number of classes - 2,
            classes of objects from file EXE1.COD,
            print - on,
            do not create output profile,
            steps of tables:
                 for function P1 - 1,
                 for function P2 - 10.
```

EXERCISE 2

```
Task: To make discretization and to print the obtained histogram for
      the function f. Create output profile. Compare the obtained
      thresholds with the result of Exercise 6 of Section 1.3 of the
      written exercises.
Program to use: CODM.
Input files: profile - no,
             file with values of functions - EXE2.PAT.
Input data: mode of work - discretization,
            number of classes - 3,
            classes of objects according to the Table 2 from
            Exercise 6 (Section III of the written exercises),
            print - on,
            create output profile with name EXE2.COD,
            automatic discretization for the function f with
            3 intervals.
```

EXERCISE 3

```
Task: To make discretization and to print the obtained histogram for
      the function g. Compare the obtained thresholds with the
      result of Exercise 6 of Section 1.3 of the written exercises.
Program to use: CODM.
Input files: profile - EXE2.COD,
             file with values of functions - EXE2.PAT.
Input data: mode of work - discretization,
            number of classes - 2,
            classes of objects from file EXE2.COD,
            print - on,
            do not create output profile,
            automatic discretization for the function g with
            2 intervals.
```

EXERCISE 4

Task: To make discretization and to print the obtained histograms for the functions **f** and **g**. Compare the obtained histograms with the result of Exercise 7 of Section 1.3 of the written exercises.

Program to use: **CODM**.

Input files: profile - EXE2.COD,
file with values of functions - EXE2.PAT.

Input data: mode of work - discretization,
number of classes - 3,
classes of objects from file EXE2.COD,
print - on,
do not create output profile,
thresholds of discretization:
for the function f: 5, 30,
for the function g: 2 .

EXERCISE 5

Task: To make discretization for the functions **a**, **b**, **c** and to code the objects (Table from Exercise 9 of Section 1.4 of the written exercises). Print the obtained coding and compare it with the result of Exercise 9.

Program to use: **CODM**.

Input files: profile - no,
file with values of functions - EXE5.PAT.

Input data: mode of work - discretization and coding,
number of classes for discretization - 3,
number of classes for coding - 3,
all objects belong to the third class,
type - on, print - on,
subclasses - off,
create output profile with name EXE5.COD,
thresholds of discretization:
for the function a: 0.5,
for the function b: 4.5, 10,
for the function c: 0, 5,
methods of coding:
for the function a - I,
for the function b - S,
for the function c - S.

EXERCISE 6

```
Task: To code the objects (Table from Exercise 9 of Section 1.4 of
      the written exercises). Print the obtained coding and compare
      it with the result of Exercise 9.
Program to use: CODM.
Input files: profile - EXE5.COD,
             file with values of functions - EXE5.PAT.
Input data: mode of work - coding,
            number of classes for coding - 3,
            classes of objects from file EXE5.COD,
            type - on, print - on,
            subclasses - off,
            do not create output profile,
            thresholds of discretization from file EXE5.COD,
            methods of coding:
                 for the function a - S,
                 for the function b - I,
                 the function c is not coded.
```

EXERCISE 7

```
Task: To classify the objects (Table from Exercise 11 of Section 1.4
      of the written exercises) by using the pattern recognition
      algorithm CORA-3. Compare the result with the results of
      Exercises 11, 12 and 16 of Sections 1.6, 1.7, and 1.10 of the
      written exercises.
Program to use: PRAL.
Algorithm: CORA-3.
Input files: profile - no,
             file with coding of object - EXE7.RAT.
Input data: k1=3, k1t=1, k2=3, k2t=0,
            all objects and components are used,
            to print: coding, traits, lattices of
                      traits, table of voting,
            value of delta - 0,
            only one variant, no control experiments.
```

EXERCISE 8

```
Task: To classify the objects (Table from Exercise 11 of Section 1.6
      of the written exercises) with subclasses in the first class
      (Table from Exercise 13 of Section 1.8 of the written
      exercises) by using the pattern recognition algorithm
      CLUSTERS. Compare the result with the results of Exercises 14
      and 15 of Sections 1.8 and 1.9 of the written exercises.
Program to use: PRAL.
Algorithm: CORA-3.
Input files: profile - no,
             file with coding of object - EXE8.RAT.
Input data: k1=2, k1t=1, k2=3, k2t=0,
            all objects and components are used,
            to print: coding, traits, lattices of
                      traits, table of voting,
            value of delta - 0,
            only one variant, no control experiments.
```

EXERCISE 9

```
Task: To classify the objects (Table from Exercise 17 of Section 1.11
      of the written exercises) by using the pattern recognition
      algorithm HAMMING. Compare the results with the results
      of Exercises 17 and 18 of Sections 1.11 and 1.12 of the written
      exercises.
Program to use: PRAL.
Algorithm: HAMMING.
Input files: profile - no,
             file with coding of object - EXE9.RAT.
Input data: all objects and components are used,
            to print: coding, kernel,
            all weights are equal to 1,
            value of delta - 3,
            only one variant, no control experiments.
```

EXERCISE 10

```
Task: To carry out discretization and coding for values of functions
      on seismic flow calculated for the objects of the first region
      of the Southern California in Exercise 15 of the computer exercises
      with the programs for analysis of earthquake catalogs. Use program
      CODMF (a version of program CODM), which reads values of functions of
      objects from a file created by program FUNC. Compare the results with
      the printout given at the end of these exercises.
Program to use: CODMF.
Input files: profile - no.
             file with values of functions - EX15.PAT (created in
             Exercise 15 of the computer exercises with the programs
             for analysis of earthquake catalogs).
Input data: mode of work - discretization and coding,
            number of classes for discretization - 2,
            number of classes for coding - 3,
            classes of objects according to their time order:
            2 1 1 1 1 1 1 3 3 3 2 1 1 1 3 3 1 1 1 3 3 1 1 1 3 3
            3 2 2 2 2 2 2 2 1 1 1 3 1 1 1 3 3 3 2 2 2 1 1 1 1 3
            3 3 2,
            type - on, print - on,
            subclasses - off,
            create output file with coding with name EXE10.RAT,
            create output profile with name EXE10.COD,
            carry out automatic discretization for all functions
            except of SIGTH with 2 intervals for the function
            N2 and 3 intervals for other functions,
            construct diagrams of discretization and print them,
                       cod all functions except of SIGTH by S-method.
```

EXERCISE 11

```
Task: By using the thresholds for discretization from profile
      EXE10.COD to create the file with coding of objects of the
      first region of the Southern California with 2 months step.
      Values of functions on seismic flow for these objects were
      calculated in Exercise 16 of the computer exercises with the
      programs for analysis of earthquake catalogs. Use program CODMF
      (a version of program CODM), which reads values of functions of
      objects from a file created by program FUNC.
Program to use: CODMF.
Input files: profile - EXE10.COD (created in Exercise 10),
             file with values of functions - EX16.PAT (created in
             Exercise 16 of the computer exercises with the
             programs for analysis of earthquake catalogs).
Input data: mode of work - coding,
            number of classes for coding - 3,
            all objects belong to the third class,
            type - on, print - off,
            subclasses - off,
            create output file with coding with name EXE11.RAT,
            create output profile with name EXE11.COD,
            cod all functions except of SIGTH by S-method using
            thresholds for discretization from the input profile.
```

IV. RESULTING PRINTOUT FOR COMPUTER EXERCISE 10

```
Input profile - no
File with values of functions - ex15.pat
   number of objects=55; number of functions=10
Mode: discretization and coding
Number of classes for coding = 3; subclasses - off
Number of classes for discretization = 2
Output profile - exe10.cod

 thresholds for 1 function N2 obtained by a-method
                0.00
 class:                      undefined
  1(a):   8( 32%)  17( 68%)   0(  0%)
  2(b):  12( 92%)   1(  8%)   0(  0%)
100 -|
 90 -|              b
 80 -|
 70 -|                                           a
 60 -|
 50 -|
 40 -|
 30 -|             a
 20 -|
 10 -|                                            b
  0 -+------------------------------------------------------------
                                  0.00

 thresholds for 2 function K obtained by a-method
               -1.00     1.00
 class:                                  undefined
  1(a):   4( 16%)  13( 52%)   8( 32%)   0(  0%)
  2(b):   6( 46%)   6( 46%)   1(  8%)   0(  0%)
100 -|
 90 -|
 80 -|
 70 -|
 60 -|
 50 -| b                           ab
 40 -|
 30 -|                                                         a
 20 -|a
 10 -|                                                          b
  0 -+------------------------------------------------------------
         -1.00                                             1.00
```

```
 thresholds for 3 function G obtained by a-method
                0.50     0.71
 class:                                 undefined
  1(a):  12( 48%)   9( 36%)   4( 16%)   0(  0%)
  2(b):   3( 23%)   2( 15%)   8( 62%)   0(  0%)
100 |
 90 |
 80 |
 70 |
 60 |                                                           b
 50 |                              a
 40 | a
 30 |
 20 |  b                            b                          a
 10 |
  0 +------------------------------------------------------------
          0.50                                             0.71

 thresholds for 4 function SIGMA obtained by a-method
               49.00    81.00
 class:                                 undefined
  1(a):   5( 20%)   8( 32%)  12( 48%)   0(  0%)
  2(b):   8( 62%)   4( 31%)   1(  8%)   0(  0%)
100 |
 90 |
 80 |
 70 |
 60 |  b
 50 |
 40 |                              a                           a
 30 |                               b
 20 | a
 10 |                                                           b
  0 +------------------------------------------------------------
         49.00                                            81.00

 thresholds for 5 function Smax obtained by a-method
                9.65    19.64
 class:                                 undefined
  1(a):   7( 28%)   8( 32%)  10( 40%)   0(  0%)
  2(b):   6( 46%)   4( 31%)   3( 23%)   0(  0%)
100 |
 90 |
 80 |
 70 |
 60 |
 50 |  b
 40 |                                                          a
 30 | a                            ab
 20 |                                                           b
 10 |
  0 +------------------------------------------------------------
          9.65                                            19.64
```

```
thresholds for 6 function Zmax obtained by a-method
                4.18     5.03
 class:                                 undefined
  1(a):   5( 20%)   7( 28%)  13( 52%)   0(  0%)
  2(b):   8( 62%)   4( 31%)   1(  8%)   0(  0%)
100 -|
 90 -|
 80 -|
 70 -|
 60 -| b
 50 -|
 40 -|                            a                           a
 30 -|                             b
 20 -|a
 10 -|                                                         b
  0 -+-----------------------------------------------------------
          4.18                                             5.03
```

```
 thresholds for 7 function N3 obtained by a-method
                3.00     5.00
 class:                                 undefined
  1(a):   7( 35%)   8( 40%)   5( 25%)   5( 20%)
  2(b):   6( 50%)   1(  8%)   5( 42%)   1(  8%)
100 -|
 90 -|
 80 -|
 70 -|
 60 -|
 50 -| b
 40 -|a                           a                            b
 30 -|                                                        a
 20 -|
 10 -|                             b
  0 -+-----------------------------------------------------------
          3.00                                             5.00
```

```
 thresholds for 8 function q obtained by a-method
                0.00    11.00
 class:                                 undefined
  1(a):   7( 39%)   3( 17%)   8( 44%)   7( 28%)
  2(b):   4( 36%)   5( 45%)   2( 18%)   2( 15%)
100 -|
 90 -|
 80 -|
 70 -|
 60 -|
 50 -|                             b
 40 -|ab                                                      a
 30 -|
 20 -|                            a                            b
 10 -|
  0 -+-----------------------------------------------------------
          0.00                                            11.00
```

```
 thresholds for 10 function Bmax obtained by a-method
              17.00    55.00
 class:                                        undefined
  1(a):   4( 16%)  11( 44%)  10( 40%)   0(  0%)
  2(b):   8( 62%)   2( 15%)   3( 23%)   0(  0%)
100 -
 90 -
 80 -
 70 -
 60 -  b
 50 -
 40 -                             a                            a
 30 -
 20 - a                            b                            b
 10 -
  0 -------------------------------------------------------------
         17.00                                            55.00

   length of vectors = 17
      discretization and coding of functions:
          method
function of coding        values of thresholds
  N2         s         0.00
  K          s        -1.00    1.00
  G          s         0.50    0.71
  SIGMA      s        49.00   81.00
  Smax       s         9.65   19.64
  Zmax       s         4.18    5.03
  N3         s         3.00    5.00
  q          s         0.00   11.00
  Bmax       s        17.00   55.00
      discretization has not been made for functions:
SIGTH,
coding has been written in file with name: ex16.rat
```

vectors of class 1(D0)

		N2		K		G		SIGMA		Smax		Zmax		N3		q		Bmax
			2		4		6		8		10		12		14		16	
1.	1385	1	1	1	0	0	1	1	0	1	0	1	-	-	-	-	0	1
2.	1395	1	0	1	0	1	0	1	0	1	0	1	-	-	-	-	0	1
3.	1405	1	0	0	0	1	0	1	1	1	0	1	-	-	-	-	1	1
4.	140o	0	0	0	0	1	0	1	1	1	0	1	-	-	-	-	0	1
5.	141o	0	0	0	1	1	0	0	0	0	0	0	-	-	-	-	0	1
6.	142o	0	0	1	1	1	0	0	0	0	0	0	0	1	-	-	0	1
7.	146d	0	0	0	1	1	0	0	0	0	0	0	0	1	-	-	0	1
8.	147d	0	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	0
9.	148d	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
10.	1507	0	0	1	1	1	0	1	0	1	0	0	0	0	1	1	0	1
11.	1517	0	1	1	1	1	0	0	0	1	0	0	0	0	1	1	0	1
12.	1527	1	0	1	1	1	0	1	0	1	0	1	0	1	1	1	0	1
13.	1542	0	0	1	1	1	0	0	0	1	0	0	0	1	1	1	0	0
14.	1552	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
15.	1562	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
16.	1664	1	0	1	0	1	1	1	0	1	1	1	1	1	0	0	1	1
17.	1674	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1
18.	1684	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	1
19.	1692	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0
20.	1702	0	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0	0
21.	1712	0	0	0	0	1	0	1	1	1	0	1	1	1	0	0	0	0
22.	177o	1	0	1	0	0	1	1	1	1	1	1	0	1	0	1	0	0
23.	178o	1	0	1	0	0	1	1	1	1	1	1	0	1	0	1	0	1
24.	179o	1	0	1	0	0	1	1	1	1	1	1	0	1	0	0	0	1
25.	1805	0	0	0	0	1	0	1	0	1	1	1	1	1	0	0	0	0

vectors of class 2(N0)

1.	1381	1	1	1	0	0	1	1	0	1	0	1	-	-	-	-	0	1
2.	1461	1	0	1	1	1	0	1	0	1	0	1	1	1	-	-	1	1
3.	1601	1	0	1	0	0	1	1	0	0	0	1	0	0	1	1	0	1
4.	1611	1	1	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1
5.	1621	1	0	1	0	1	1	1	1	1	1	1	0	0	1	1	1	1
6.	1631	1	0	0	0	1	1	1	1	1	1	1	0	0	1	1	1	1
7.	1641	1	1	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1
8.	1651	1	0	1	0	0	0	1	1	1	1	1	1	1	0	1	1	1
9.	1661	1	0	1	0	0	0	1	1	1	1	1	1	1	0	0	1	1
10.	1751	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1
11.	1761	1	1	1	0	0	1	1	0	1	1	1	1	1	0	1	0	0
12.	1771	1	1	1	0	0	1	1	1	1	1	1	0	1	0	1	0	0
13.	1841	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0	0

vectors of class 3(X)

1.	1431	0	0 1	1 1	0 0	0 1	0 0	0 1	- -	0 1
2.	1441	0	1 1	1 1	0 0	0 1	0 0	1 1	- -	0 1
3.	1451	0	1 1	1 1	1 1	0 1	1 1	1 1	- -	1 1
4.	1491	0	1 1	1 1	0 0	0 0	0 0	0 0	0 1	0 0
5.	1501	0	0 1	1 1	0 0	0 1	0 0	0 0	1 1	0 0
6.	1531	0	0 1	1 1	0 0	0 1	0 0	0 1	1 1	0 0
7.	1541	0	0 1	1 1	0 0	0 1	0 0	0 1	1 1	0 0
8.	1571	0	1 1	1 1	0 0	0 0	0 0	0 0	1 1	0 0
9.	1581	0	1 1	0 1	1 1	1 1	1 1	0 0	1 1	0 1
10.	1591	0	1 1	0 1	1 1	0 0	0 1	0 0	1 1	0 1
11.	1691	0	1 1	1 1	0 0	0 0	0 0	1 1	0 0	0 0
12.	1721	0	1 1	0 1	0 1	1 1	0 0	1 1	0 0	0 0
13.	1731	0	1 1	1 1	1 1	1 1	1 1	1 1	0 0	0 0
14.	1741	0	1 1	1 1	1 1	0 1	1 1	1 1	0 0	0 0
15.	1811	0	0 0	1 1	0 0	0 0	0 0	1 1	0 0	0 0
16.	1821	0	0 0	1 1	0 0	0 0	0 0	1 1	0 0	0 0
17.	1831	0	1 1	1 1	0 0	0 0	0 0	1 1	0 0	0 0