Кравцова Александра Николаевна, группа 12-1

Лабораторная работа № 4

**Вариант № 8**

Распознавание образов, описываемых бинарными признаками

**Цель работы**

Синтезировать алгоритмы распознавания образов, описываемых бинарными признаками. Исследовать синтезированные алгоритмы распознавания с точки зрения ожидаемых потерь и ошибок.

**Задание**

Написать код, реализующий алгоритм распознавания образов, заданных бинарными изображениями.  
Провести имитационное моделирование алгоритма, в ходе которого рассчитать значения вероятности ошибок распознавания для трех различных случаев априорных вероятностей гипотез:

**Предварительные данные:**

* Образы - **ФИО**
* Значение вероятности искажения символа **= 0.4**

**Код программы (Желтым выделены отличные от шаблона фрагменты):**

%Файл pr54\_rec\_bin. Синтез и анализ алгоритмов распознавания образов по дискретным

%признакам (на примере распознавания бинарных изображений)

% ПРИМЕР ДЛЯ ТРЁХ КЛАССОВ С ВЫВОДОМ МАТРИЦ ОШИБОК

clear all; close all;

%1.Задание исходных данных

n=35; %количество признаков (исходя из размера изображений)

M=3; % ЗДЕСЬ М меняется на 3

s=zeros(n,M); %количество классов и эталонные описания

%1.1. Задание эталонов классов (ЗДЕСЬ добавляется описание третей буквы)

letterK = [1 0 0 0 1 ...

1 0 0 1 0 ...

1 0 1 0 0 ...

1 1 0 0 0 ...

1 0 1 0 0 ...

1 0 0 1 0 ...

1 0 0 0 1 ]';

letterA = [0 0 1 0 0 ...

0 1 0 1 0 ...

1 0 0 0 1 ...

1 1 1 1 1 ...

1 0 0 0 1 ...

1 0 0 0 1 ...

1 0 0 0 1 ]';

letterN = [1 0 0 0 1 ... % Добавляется третья буква

1 0 0 0 1 ...

1 1 0 0 1 ...

1 0 1 0 1 ...

1 0 0 1 1 ...

1 0 0 0 1 ...

1 0 0 0 1 ]';

% отображение исходного вида букв

figure;

subplot(1,3,1), imshow(reshape(1 - letterK,5,7)')

subplot(1,3,2), imshow(reshape(1 - letterA,5,7)')

subplot(1,3,3), imshow(reshape(1 - letterN,5,7)')

s(:,1)=letterK;

s(:,2)=letterA;

s(:,3)=letterN; % Добавляется третья буква

%1.2. Задание параметров эксперимента (ЗДЕСЬ добавляется третье значение в pw)

pw=[0.5 0.5 0.5]; %априорные вероятности гипотез

np=sum(pw); pw=pw/np;%исключение некорректного задания априорных вероятностей

N = 20; %количество шагов изменения варьируемого параметра - pi

K = 1000;%количество реализаций

pi=zeros(1,N); %массив вероятностей искажения символа

% ЗДЕСЬ Удаляется пункт 1.3.

% 1.4. Матрицы ошибок (ЗДЕСЬ добавляется теоретическая матрица ошибок)

Pc\_=zeros(M);%экспериментальная матрица вероятностей ошибок

Pt = zeros(M);% МАТРИЦЫ теперь двумерные

% ЗДЕСЬ убирается объявление цикла (тело цикла остаётся)

%Цикл по значениям вероятности искажения элементов символов

% for t = 1:N,

% pi(t)=(1/N)\*(t-1); pI=pi(t); %вероятность искажения элемента (пикселя)

pI = 0.4; % ЗДЕСЬ задается фиксированное значение вероятности искажения (pI)

%2.Синтез решающего правила и расчет теоретических вероятностей ошибок

if pI==0, pI=0.0001; end; %регуляризация разделяющей функции

if pI==0.5, pI=0.4999; end;

pI\_=1-pI;

s\_=1-s;%получение инвертированных изображений

G1=zeros(1,n); G2=zeros(1,n);

% ДОБАВЛЯЕТСЯ двойной цикл и пункты 1.1. - 1.3. уходят внутрь него

% В п.1.1-1.2 индексы при s() и при pw() равные 1 меняются на ii,

% равные 2 меняются на jj

% П. 1.1. и 1.2. надо скопировать в п.3.2.

% Попарное сравнение классов

for ii = 1 : M - 1

for jj = ii + 1 : M

% 2.1. Вычисление порога принятия решений

ns=sum(abs(s(:,ii)-s(:,jj)));%общее количество несовпадающих элементов

l0\_ = log(pw(jj)/pw(ii)); %порог принятия решения

L0 = log(pw(jj)/pw(ii)) / (2\*log(pI\_)-2\*log(pI)) + ns/2;

L0r = floor(L0);

% 2.2.Вычисление коэффициентов разделяющей функции (этот пункт отсюда можно убрать)

for k=1:n,

G1(1,k)=log((s(k,ii)\*pI\_+s\_(k,ii)\*pI)/(s(k,jj)\*pI\_+s\_(k,jj)\*pI));

G2(1,k)=log((s(k,ii)\*pI+s\_(k,ii)\*pI\_)/(s(k,jj)\*pI+s\_(k,jj)\*pI\_));

end;

% 2.3. Определение вероятностей перепутывания

% Здесь вместо p12th(t) будет Pt(ii,jj), вместо p21th(t) - Pt(jj,ii)

if pI < 0.5, %расчет вероятностей ошибок

Pt(ii,jj) = binocdf(L0r, ns, 1-pI);

Pt(jj,ii) = 1 - binocdf(L0r,ns,pI);

else

Pt(ii,jj) = 1 - binocdf(L0r,ns,1-pI);

Pt(jj,ii) = binocdf(L0r,ns,pI);

end;

end

end

% ЗДЕСЬ добавляется строчка с вычислением диагональных элементов теор. матрицы ошибок

% Вычисление вероятностей правильного распознавания (диагональные элементы)

Pt = Pt + diag(ones(3, 1) - sum(Pt, 2));

figure;

%3.Тестирование алгоритма методом статистических испытаний

for kk=1:K,%цикл по числу реализаций

for i=1:M,%цикл по классам

% 3.1. Моделирование искажения

x = s(:,i);

r = rand(n,1);

ir = find(r < pI);

x(ir) = 1-x(ir);%искажение элементов – инверсия в случайных точках

x\_ = 1-x;

% 3.2. Классификация искаженного образа (попарное сравнение классов)

% ЗДЕСЬ добавляется инициализация массива для хранения

% результатов попарных сравнений классов; п.3.2 помещается

% внутрь двойного цикла вместе с п.2.1 и 2.2.

iais = []; % результаты попарных сравнений (индексы классов)

for ii = 1 : M - 1

for jj = ii + 1 : M

% Копия 2.1. Вычисление порога принятия решений

ns=sum(abs(s(:,ii)-s(:,jj)));%общее количество несовпадающих элементов

l0\_ = log(pw(jj)/pw(ii)); %порог принятия решения

L0 = log(pw(jj)/pw(ii)) / (2\*log(pI\_)-2\*log(pI)) + ns/2;

L0r = floor(L0);

% Копия 2.2.Вычисление коэффициентов разделяющей функции

for k=1:n,%вычисление коэффициентов разделяющей функции

G1(1,k)=log((s(k,ii)\*pI\_+s\_(k,ii)\*pI)/(s(k,jj)\*pI\_+s\_(k,jj)\*pI));

G2(1,k)=log((s(k,ii)\*pI+s\_(k,ii)\*pI\_)/(s(k,jj)\*pI+s\_(k,jj)\*pI\_));

end;

% ЗДЕСЬ вместо индексов классов будут ii и jj

% 3.2. Классификация искаженного образа

u=G1\*x+G2\*x\_-l0\_; %вычисление значения разделяющих функций

if u>0

iai=ii;

else

iai=jj;

end

% ЗДЕСЬ Запоминаем результат попарного сравнения

iais = [iais, iai];

end;

end;

% ЗДЕСЬ выбираем индекс класса, за который проголосовали

% большинство парных классификаторов

id = mode(iais); % самый часто повторяющийся индекс

% 3.3. Фиксация результата распознавания

Pc\_(i,id)=Pc\_(i,id) + 1; %фиксация результата распознавания

if (kk==1),%отображение примеров искажения символов

subplot(1,3,i), imshow(reshape(1 - x\_,5,7)')

end;

end

end

Pc\_ = Pc\_ / K; % Здесь можно убрать индексы при Pc\_

% end; % убирается end относящийся к циклу по t

% ДОБАВЛЯЕТСЯ вывод матриц

disp('Теоретическая матрица ошибок')

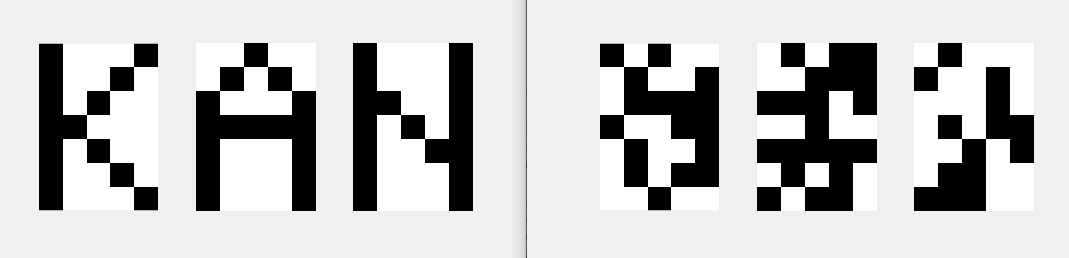
disp(Pt)

disp('Экспериментальная матрица ошибок')

disp(Pc\_)

**Результат:**

Значения априорных вероятностей: **[0.5 0.5 0.5]**



*Рис.1.Отображение исходных и искаженных образов*

Теоретическая матрица ошибок

0.4636 0.3075 0.2288

0.1501 0.6034 0.2465

0.2288 0.2465 0.5247

Экспериментальная матрица ошибок

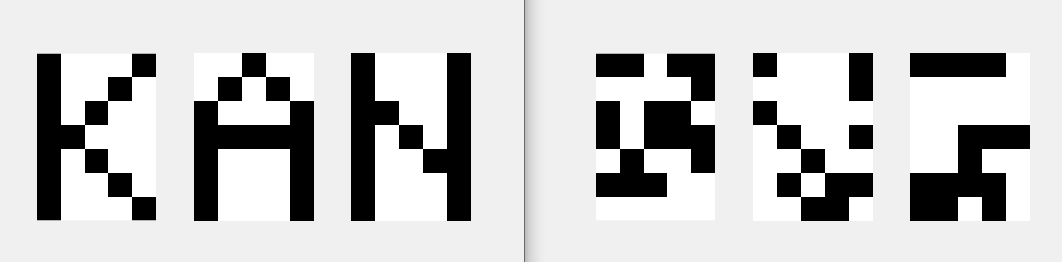
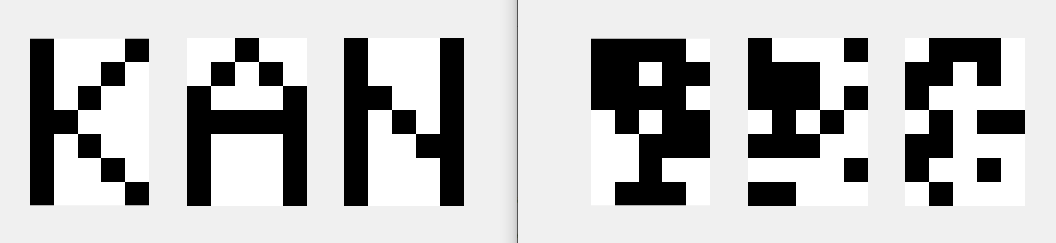
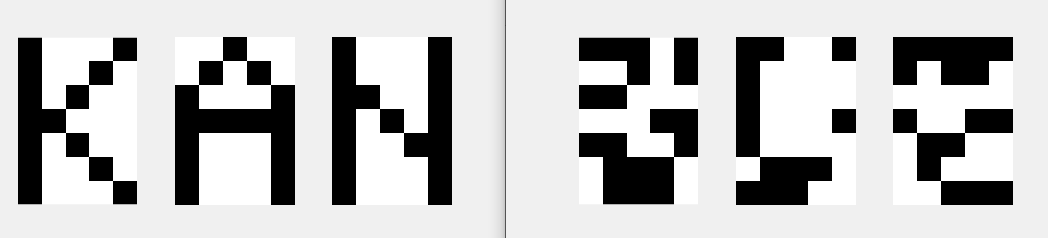
0.6140 0.1990 0.1870

0.1150 0.6870 0.1980

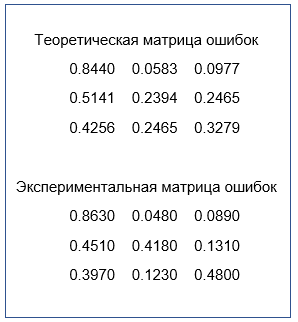
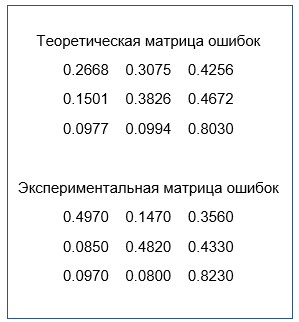
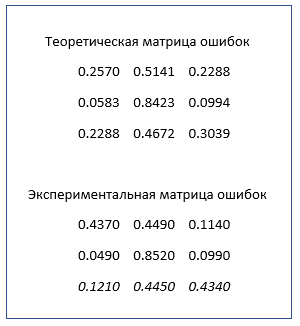
0.1530 0.2040 0.6430

*Табл. 1.*

**Анализ:** В данном случае все априорные вероятности равны, а значит единственный параметр принятия решений – вероятность искажения ( = 0.4). Вероятность правильного распознавания по матрицам, представленных в табл. 1. приблизительна равна 0.5, что объяснимо величиной вероятности распознавания. Принятие решения сводится к гипотезе с наибольшим значением условной вероятности.

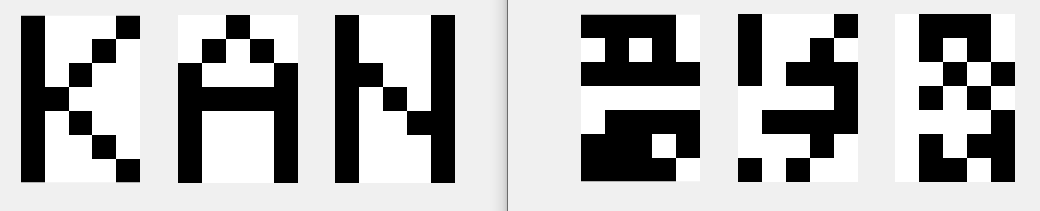
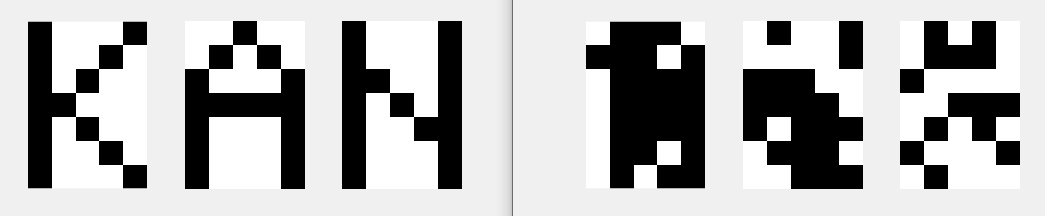
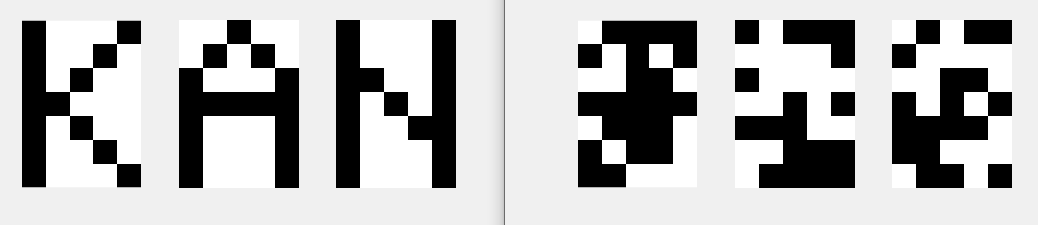
В этом пункте рассмотрим несколько ситуаций:

1. **[0.3 0.3 0.7]**
2. **[0.7 0.3 0.3]**
3. **[0.3 0.7 0.3]**

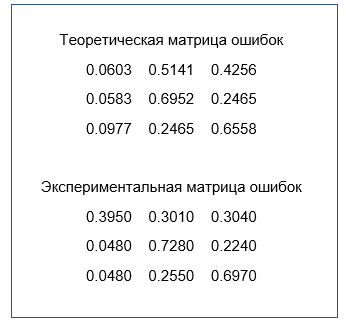
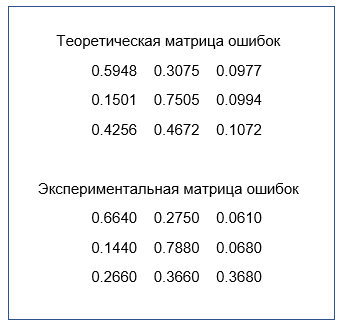
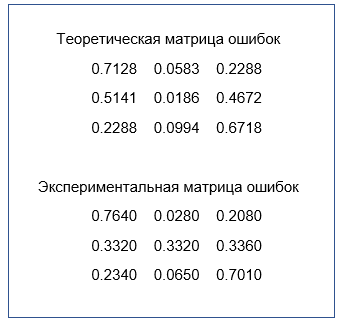
***Рис.2.Отображение исходных и искаженных образов: a), b),c)*

*рис.3. Матрицы ошибок : a), b),c)*

**Анализ:** В этом случае учитывается вероятность искажения ( **= 0.4**) и априорные  
вероятности. Решение принимается на основе попарного сравнения условных вероятностей , и с единицей. Принятие решения сводится также к выбору класса с наибольшим полученным значением. По матрице ошибок на рис.2 видно, что у класса с наибольшим значением априорной вероятности - наибольшая вероятность правильного распознавания среди всех классов.

В этом пункте рассмотрим несколько ситуаций:

1. **[0.3 0.7 0.7]**
2. **[0.7 0.7 0.3]**
3. **[0.7 0.3 0.7]**

***Рис.4.Отображение исходных и искаженных образов*

*рис.5. Матрицы ошибок : a), b),c)*

**Анализ:** Аналогично с п.2, решение принимается на основе попарного сравнения условных вероятностей , и с единицей. Касаемо матриц ошибок, отображенных на рис.5, можно отметить, что у класса j с наименьшим значением вероятности ( вероятность верного распознавания при  **= 0.4** будет так же наименьшей относительно имеющихся классов.

**Вывод:**

В результате работы мы синтезировали алгоритмы распознавания образов, описываемых бинарными признаками, и исследовали синтезированные алгоритмы распознавания с точки зрения ожидаемых потерь и ошибок. Также определили непосредственное влияние вероятностей исходных классов, если они находятся в значимом отличии друг от друга.