**Гузенко А.М. Группа 7.2. Вариант 4**

Лабораторная работа №2

Распознавание образов, описываемых гауссовскими случайными

векторами с одинаковыми матрицами ковариаций

**Цель работы**

Синтезировать алгоритмы распознавания образов, описываемых гауссовскими случайными векторами с одинаковыми матрицами ковариаций. Исследовать синтезированные алгоритмы распознавания с точки зрения ожидаемых потерь и ошибок.

**Задание**

1. Получить у преподавателя вариант задания.
2. Написать код реализующий алгоритм распознавания образов, описываемых гауссовскими случайными векторами с заданными параметрами.
3. Получить матрицы ошибок на основе аналитических выражений и вычислительного эксперимента.
4. Провести анализ полученных результатов и представить его в виде выводов по проделанной работе.

**Код программы**

%Синтез и анализ алгоритмов распознавания ГСВ с одинаковой

%матрицей ковариации (двумерный вектор признаков)

clear all; close all;

%1.Задание исходных данных

n=2; M=3; %%размерность признакового пространства и число классов

K = 200; %количество статистических испытаний

m = [0 -1; -4 2; -1 2]'; % координаты центров классов

% априорные вероятности классов (доля образов каждого класса в общей выборке)

pw = [0.333, 0.333, 0.333];

np=sum(pw); pw=pw/np;

C = [3 -2; -2 3]; % матрица ковариаций классов

C\_ = C^-1;

D = C(1,1);

% 1.1. Визуальзация исходной совокупности образов

N = K \* M;

NN = zeros(M, 1);

for k = 1 : M - 1

NN(k) = uint16(N \* pw(k));

end;

NN(M) = N - sum(NN);

label = {'bo', 'r+', 'k\*', 'gx'};

IMS = []; %общая совокупность образов

figure; hold on; title('Исходные метки образов');

for i=1:M,%цикл по классам

ims = repmat(m(:,i), [1, NN(i)]) + randncor(n,NN(i),C); %генерация К образов i-го класса

if (n == 2)

plot(ims(1, :), ims(2, :), label{i}, 'MarkerSize', 10, 'LineWidth', 1);

elseif (n == 3)

plot3(ims(1, :), ims(2, :), ims(3, :), label{i}, 'MarkerSize', 10, 'LineWidth', 1);

end;

IMS = [IMS, ims]; %добавление в общую совокупность образов

end;

%2.Расчет разделяющих функций и матрицы вероятностей ошибок распознавания

G=zeros(M,n+1); PIJ=zeros(M); l0\_=zeros(M);

for i = 1 : M,

G(i,1:n)=(C\_\*m(:,i))'; G(i,n+1)=-0.5\*m(:,i)'\*C\_\*m(:,i);

for j=i+1:M,

l0\_(i,j)=log(pw(j)/pw(i));

h=0.5\*(m(:,i)-m(:,j))'\*C\_\*(m(:,i)-m(:,j)); sD=sqrt(2\*h);

PIJ(i,j)=normcdf(l0\_(i,j),h,sD); PIJ(j,i)=1-normcdf(l0\_(i,j),-h,sD);

end;

PIJ(i,i)=1-sum(PIJ(i,:));%нижняя граница вероятности правильного распознавания

end;

% 2.1. Визуальзация результатов распознавания образов

figure; hold on; title('Результат классификации образов');

for i = 1 : N,%цикл по всем образам совокупности

z = [IMS(:, i); 1]; %значение очердного образа из общей совокупности

u=G\*z+log(pw');%вычисление значения разделяющих функций

[ui,iai]=max(u);%определение максимума (iai - индекс класса)

if (n == 2)

plot(IMS(1, i), IMS(2, i), label{iai}, 'MarkerSize', 10, 'LineWidth', 1);

elseif (n == 3)

plot3(IMS(1, i), IMS(2, i), IMS(3, i), label{iai}, 'MarkerSize', 10, 'LineWidth', 1);

end;

end;

%3.Тестирование алгоритма методом статистических испытаний

x=ones(n+1,1); Pc\_=zeros(M);%экспериментальная матрица вероятностей ошибок

for k=1:K,%цикл по числу испытаний

for i=1:M,%цикл по классам

[x\_,px]=randncor(n,1,C);

x(1:n,1)=m(:,i)+x\_;%генерация образа i-го класса

u=G\*x+log(pw');%вычисление значения разделяющих функций

[ui,iai]=max(u);%определение максимума

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

end;

end;

Pc\_=Pc\_/K;

disp('Теоретическая матрица вероятностей ошибок');disp(PIJ);

disp('Экспериментальная матрица вероятностей ошибок');disp(Pc\_);

%4.Визуализация областей принятия решений для двумерного случая

if n==2,

xmin1=-4\*sqrt(D)+min(m(1,:)); xmax1=4\*sqrt(D)+max(m(1,:));

xmin2=-4\*sqrt(D)+min(m(2,:)); xmax2=4\*sqrt(D)+max(m(2,:));

x1=xmin1:0.05:xmax1; x2=xmin2:0.05:xmax2;

axis([xmin1,xmax1,xmin2,xmax2]);%установка границ поля графика по осям

figure; hold on; grid on;

[X1,X2]=meshgrid(x1,x2); %матрицы значений координат случайного вектора

x12=[X1(:),X2(:)];

for i=1:M,

f2=mvnpdf(x12,m(:,i)',C); %массив значений плотности распределения

f3=reshape(f2,length(x2),length(x1));%матрица значений плотности распределения

[Ch,h]=contour(x1,x2,f3,[0.01,0.5\*max(f3(:))],'Color','b','LineWidth',0.75); clabel(Ch,h);

for j=i+1:M,%изображение разделяющих границ

wij=C\_\*(m(:,i)-m(:,j));

wij0=-0.5\*(m(:,i)+m(:,j))'\*C\_\*(m(:,i)-m(:,j));

f4=wij'\*x12'+wij0;

f5=reshape(f4,length(x2),length(x1));

[Ch\_,h\_] = contour(x1,x2,f5,[l0\_(i,j)+0.0001],'Color','k','LineWidth',1.25);

end;

end;

set(gca,'FontSize',13);

title('Области локализации классов и разделяющие границы','FontName','Courier');

xlabel('x1','FontName','Courier'); ylabel('x2','FontName','Courier');

strv1=' pw='; strv2=num2str(pw,'% G');

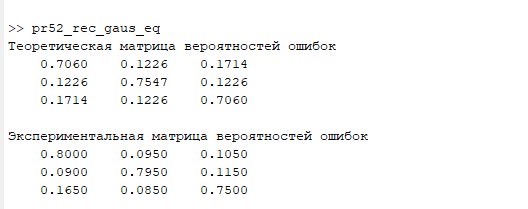
text(xmin1+1,xmax2-1, [strv1,strv2], 'HorizontalAlignment','left','BackgroundColor',...

[.8 .8 .8],'FontSize',12); legend('wi','gij(x)=0');hold off;

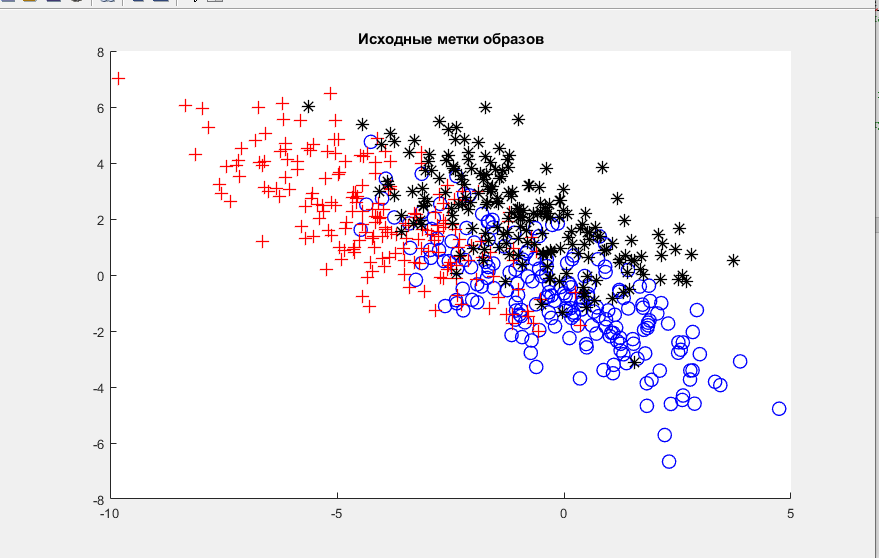
end;

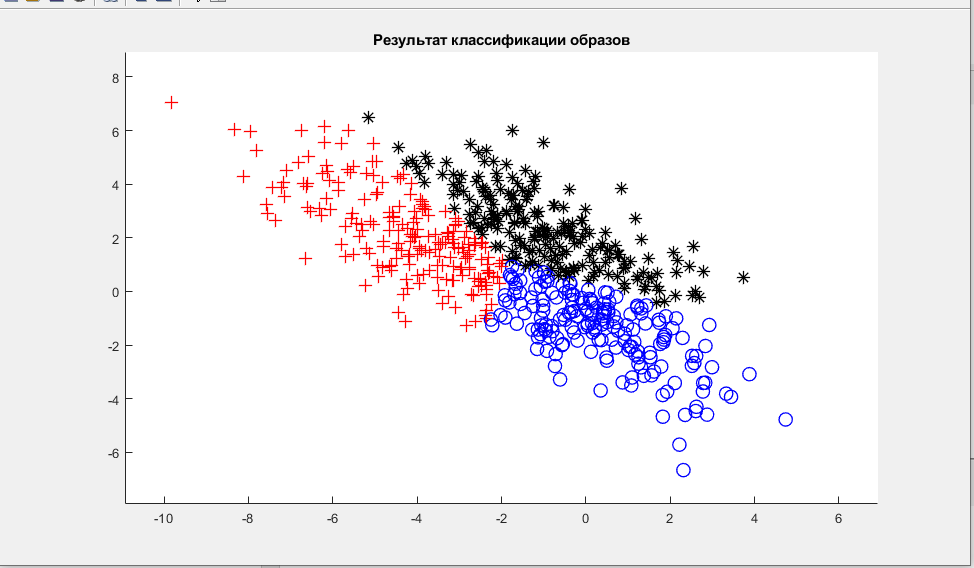
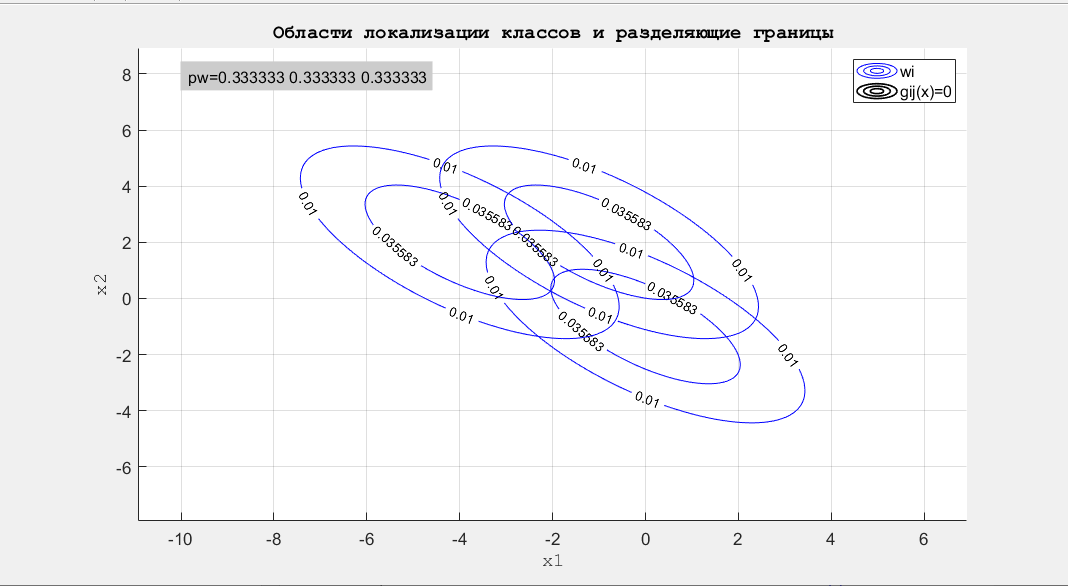
**Результаты выполнения задания**

1. Получена теоретическая и экспериментальная матрица ошибок, размерность матрицы – 3x3, кол-во испытаний - 200. Значения элементов примерно совпадают.



1. Исходные метки.



1. Результат классификации.
2. Области локализации классов и разделяющие границы.

**Вывод**

В пункте 2 и 3 мы можем увидеть, что метки успешно прошли классификацию. В пункте 1 значения элементов полученных матриц теоретических и экспериментальных ошибок примерно совпадают.