

Лабораторная работа № 3

Вариант № 4

Распознавание образов, описываемых гауссовскими случайными векторами с разными матрицами ковариаций

Цель работы

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

Задание

Построить график зависимости экспериментальной ошибки второго рода (для второго класса) от числа испытаний (объема выборки). Сравнить с теоретическим значением.

$$m1 = [0 \ -1], m2 = [-4 \ 2], C1 = [3 \ 1; 1 \ 3], C2 = [3 \ -2; -2 \ 4]$$

Код программы (внесённые изменения в шаблон кода выделены)

```
%Файл pr53_rec_gaus_uneq. Синтез и анализ алгоритмов распознавания ГСВ
с
%различными матрицами ковариации
clear all; close all;
%1.Задание исходных данных
n=2;M=2;%размерность признакового пространства и число классов
K=[100,200, 300, 400, 500, 600, 700, 800, 900, 1000];%количество
статистических испытаний
%Априорные вероятности, математические ожидания и матрицы ковариации
классов
dm=2.0;%расстояние между математическими ожиданиями классов по
координатным осям
C=zeros(n,n,M); C_=C;%матрица ковариации вектора признаков различных
классов
pw=[0.4 0.6];
pw=pw/sum(pw);
D=3*eye(2);
m=[0 -1; -4 2];
```

```

C(:,:,1)=[3 1; 1 3];
C(:,:,2)=[3 -2; -2 4];
for k=1:M,
    C_(:,:,k)=C(:,:,k)^-1;
end;
np=sum(pw); pw=pw/np; %исключение некорректного задания априорных
вероятностей
%2.Расчет матриц вероятностей ошибок распознавания
PIJ=zeros(M); PIJB=zeros(M); mg=zeros(M); Dg=zeros(M); l0_=zeros(M);
for i=1:M,
    for j=i+1:M,
        dmij=m(:,i)-m(:,j);
        l0_(i,j)=log(pw(j)/pw(i));
        dti=det(C(:,:,i)); dtj=det(C(:,:,j));
        trij=trace(C_(:,:,j)*C(:,:,i)-eye(n)); trji=trace(eye(n)-C_(:,:,i)*C(:,:,j));
        trij_2=trace((C_(:,:,j)*C(:,:,i)-eye(n))^2); trji_2=trace((eye(n)-
C_(:,:,i)*C(:,:,j))^2);
        mg1=0.5*(trij+dmij'*C_(:,:,i)*dmij-log(dti/dtj));
        Dg1=0.5*trij_2+dmij'*C_(:,:,j)*C(:,:,i)*C_(:,:,j)*dmij;
        mg2=0.5*(trji-dmij'*C_(:,:,j)*dmij+log(dtj/dti));
        Dg2=0.5*trji_2+dmij'*C_(:,:,i)*C(:,:,j)*C_(:,:,i)*dmij;
        sD1=sqrt(Dg1); sD2=sqrt(Dg2);
        PIJ(i,j)=normcdf(l0_(i,j),mg1,sD1); PIJ(j,i)=1-normcdf(l0_(i,j),mg2,sD2);
        mu2=(1/8)*dmij'*((C(:,:,i)/2+C(:,:,j)/2)^-1)*dmij...
            +0.5*log((dti+dtj)/(2*sqrt(dti*dtj)));%расстояние Бхатачария
        PIJB(i,j)=sqrt(pw(j)/pw(i))*exp(-mu2);PIJB(j,i)=sqrt(pw(i)/pw(j))*exp(-
mu2);%границы Чернова
    end;
    PIJB(i,i)=1-sum(PIJB(i,:));
    PIJ(i,i)=1-sum(PIJ(i,:));%нижняя граница вероятности правильного
распознавания
end;
%3.Тестирование алгоритма методом статистических испытаний
x=ones(n,1); u=zeros(M,1);
Pc_exp = zeros(1, numel(K)); %экспериментальная ошибка второго рода для
второго класса
Pc_theor = PIJ(2, 2); %теоретическая ошибка второго рода для второго
for idx = 1:numel(K) %экспериментальная матрица вероятностей ошибок
    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(:,:,i)); x=x+m(:,i);%генерация образа i-го класса
            for j=1:M,%вычисление значения разделяющих функций
                u(j)=-0.5*(x-m(:,j))'*C_(:,:,j)*(x-m(:,j))-
0.5*log(det(C(:,:,j)))+log(pw(j));

```

```

end;
[ui,iai]=max(u);%определение максимума
Pc_(i,iai)=Pc_(i,iai)+1;%фиксация результата распознавания
end;
end;
Pc_exp(idx) = Pc_(2, 2) / K(idx);%экспериментальная ошибка второго рода
для второго класса
end;
disp('Теоретическая матрица вероятностей ошибок');disp(PIJ);
disp('Матрица вероятностей ошибок на основе границы Чернова');disp(PIJB);
disp('Экспериментальная матрица вероятностей ошибок');disp(Pc_);

%4.Визуализация областей принятия решений для двумерного случая
if n==2,
    Es1=pw(1)*PIJ(1,2)+pw(2)*PIJ(2,1);
    Es2=sqrt(pw(1)*pw(2))*exp(-mu2);%граница Чернова для суммарной
ошибки
    Es3=pw(1)*Pc_(1,2)+pw(2)*Pc_(2,1);
    disp('Оценки суммарных ошибок'); disp([Es1,Es2,Es3]);%отображение
оценок суммарных ошибок
    xmin1=-3*sqrt(max(D(1,:)))+min(m(1,:));
    xmax1=3*sqrt(max(D(1,:)))+max(m(1,:));
    xmin2=-3*sqrt(max(D(2,:)))+min(m(2,:));
    xmax2=3*sqrt(max(D(2,:)))+max(m(2,:));
    x1=xmin1:0.1:xmax1; x2=xmin2:0.1:xmax2;
    axis([xmin1,xmax1,xmin2,xmax2]);%установка границ поля графика по
осям
    figure(1); 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(:,i)); %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_(:,i)*m(:,i)-C_(:,j)*m(:,j);
            wij0=-0.5*(m(:,i)*C_(:,i)*m(:,i)-m(:,j)*C_(:,j)*m(:,j));
            f4=wij'*x12'+wij0-0.5*log(det(C(:,i))/det(C(:,j)));
            fd=-0.5*(C_(:,i)-C_(:,j))*x12'; f4=f4+sum(x12'.*fd);
            f5=reshape(f4,length(x2),length(x1));
            [Ch_,h_]=contour(x1,x2,f5, l0_(i,j), '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(0,0, [strv1,strv2], 'HorizontalAlignment','left','BackgroundColor',...
        [.8 .8 .8],'FontSize',12); legend('wi','gij(x)=0');hold off;
end;

```

```

% Построение графика экспериментальной ошибки второго рода для второго
класса по числу испытаний
figure(2);
plot(K, Pc_exp, 'b'); hold on;
plot(K, Pc_theor*ones(size(K)), 'r');
xlabel('Число испытаний (объем выборки)'); ylabel('Ошибка второго рода');
title('Зависимость экспериментальной и теоретической ошибки второго рода
для второго класса от числа испытаний');
legend('Экспериментальная ошибка', 'Теоретическая ошибка'); grid on;

```

```

% Вывод теоретического и экспериментального значения ошибки первого
рода для второго класса
disp("Теоретическое значение: " + PIJ(2, 2));
disp("Экспериментальное значение: " + Pc_exp(end));

```

Результаты выполнения задания

В результате работы программы были выведены вот такие графики:

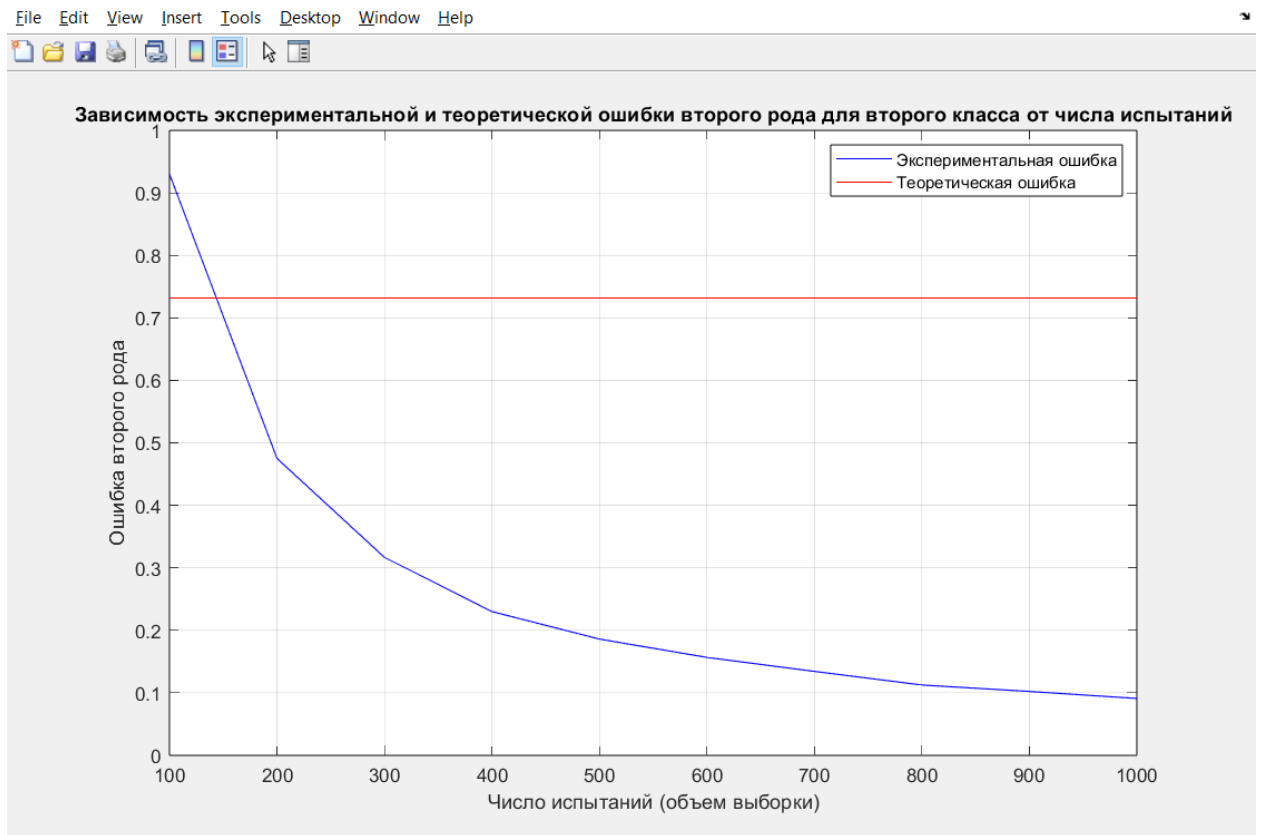


Рисунок 1 – График зависимости экспериментальной ошибки второго рода(для второго класса) от числа испытаний

```
Теоретическая матрица вероятностей ошибок
0.9946    0.0054
0.2685    0.7315

Матрица вероятностей ошибок на основе границы Чернова
0.4848    0.5152
0.3435    0.6565

Экспериментальная матрица вероятностей ошибок
93      7
9      91

Оценки суммарных ошибок
0.1632    0.2061    8.2000

Теоретическое значение: 0.73152
Экспериментальное значение: 0.091
```

Рисунок 2 – Матрицы вероятностей ошибок. Теоретическое и экспериментальное значения

Выводы

На графике зависимости ошибки экспериментальной ошибки второго рода от числа испытаний видно, что экспериментальная ошибка сильно отличается от теоретической ошибки. С увеличением числа испытаний экспериментальная ошибка уменьшается. Приблизительно равны значения ошибок при числе испытаний приблизительно равном 130.

Теоретическая матрица вероятностей ошибок показывает, как классификация должна происходить и какие ошибки могут возникнуть при распознавании.

Экспериментальная матрица вероятностей ошибок, дает оценку реальных ошибок распознавания, учитывая случайности в выборке.

Сравнение теоретической и экспериментальной ошибок распознавания для второго класса позволяет оценить эффективность алгоритма классификации и его соответствие предсказаниям с учетом случайностей в данных.