Кравцова Александра Николаевна, группа 12.1

Лабораторная работа № 6

Вариант № 4

Распознавание образов на основе непараметрических алгоритмов оценивания плотности распределения случайной величины

Цель работы

Исследовать алгоритмы распознавания образов на основе оценивания плотности распределения случайных величин и случайных векторов при использовании методов Парзена и k ближайших соседей.

Форма контроля

Письменный отчёт (допускается преставление в электронном виде). Опрос в устной форме в соответствии с перечнем контрольных вопросов.

Количество отведённых аудиторных часов

4

Содержание работы

Получить у преподавателя вариант задания и написать код, реализующий соответствующий алгоритм обработки информации. Для ответа на поставленные в задании вопросы провести численный эксперимент или статистическое имитационное моделирование и представить соответствующие графики. Провести анализ полученных результатов и представить его в виде выводов по проделанной работе.

Задание

Используя код Вашей лабораторной №3, реализуйте алгоритм распознавания образов, применив оценивание по методу k ближайших соседей. Вычислите экспериментально вероятности ошибок распознавания. Сравните их с вероятностями ошибок (теоретическими или экспериментальными), полученными в ходе выполнения лабораторной №3. Отобразите поверхности плотностей распределения классов, задаваемых теоретически, и полученных в результате оценивания.

Код программы (Желтым выделены отличные от шаблона фрагменты):

(Код программы лабораторной №3)

```
%Файл pr53_rec_gaus_uneq. Синтез и анализ алгоритмов распознавания ГСВ с
%различными матрицами ковариации
% Построить график зависимости  ошибки первого рода  в матрице Чернова
% (для  первого  класса)  от  расстояния  между  классами.  Сравнить  с
% теоретическим   значением.

DM = 0.5 : 5;     % разные значения масштаба расстояний
err2c1_p = zeros(size(DM));                % ошибка второго рода первого
класса
% Теоретические значения ошибок
terr2c1 = zeros(size(DM));                % ошибка второго рода первого
класса
err2c1 = zeros(size(DM));
% Добавляется цикл по масштабам
for tt = 1: numel(DM)   % цикл по расстояниям между классами (по масштабам)
    %1.Задание исходных данных
    n=2;M=2;%%размерность признакового пространства и число классов
    K=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);
    m=[2 7; 1 10]';
    % ЗДЕСЬ добавляется строка
    m=m*DM(tt);  % применение очередного масштаба
    C(:,:,1)=[4 -2; -2 4];
    C(:,:,2)=[5 1; 1 5];
    for l=1:M
        C_(:,:,l)=C(:,:,l)^-1;
    end
    np=sum(pw); pw=pw/np; %исключение некорректного задания априорных
вероятностей


    % 1.1. Визуализация 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
    subplot(2,3,tt);
    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(:,:,i));
%генерация K образов i-го
     plot(ims(1, :), ims(2, :), label{i}, 'MarkerSize', 10, 'LineWidth', 1);
     IMS = [IMS, ims]; %добавление в общую совокупность образов
    end;

        % + Этот пункт 1.2. Генерация обучающих выборок классов
```

```
    % число образов каждого класса
    Ks = fix(K * pw);
    Ks(end) = K - sum(Ks(1 : end - 1));
    for i=1:M,%цикл по классам
        XN{i} = repmat(m(:,i), [1, Ks(i)]) + randncor(n,Ks(i),C(:,:,i));
%генерация К образов i-го класса
    end;

    %2.Расчет матриц вероятностей ошибок распознавания (необходимо посчитать
теоретическую ошибку)
    PIJ=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));
                mg1=0.5*(trij+dmij'*C_(:,:,j)*dmij-log(dti/dtj));
                Dg1=0.5*trij^2+dmij'*C_(:,:,j)*C(:,:,i)*C_(:,:,j)*dmij;
                mg2=0.5*(trji-dmij'*C_(:,:,i)*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
        end
        PIJ(i,i)=1-sum(PIJ(i,:));%нижняя граница вероятности правильного
распознавания
    end

    %3.Тестирование алгоритма методом статистических испытаний (расчет
экспериментальных ошибок)
    Pcv=zeros(M); p=zeros(M,1); % +
    x=ones(n,1); u=zeros(M,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(:,:,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));
                p(j)=vknn(x, XN{j}, 300);
            end
            [ui,iai]=max(u);%определение максимума (3 лаба)
            Pc_(i,iai)=Pc_(i,iai)+1;%фиксация результата распознавания (3
лаба)

            [ui,iai]=max(p);         %определение максимума (k-соседей)
            Pcv(i,iai)=Pcv(i,iai)+1;%фиксация результата распознавания (k-
соседей)
        end
    end
    Pc_=Pc_/K;
    Pcv=Pcv/K;   % +
    % фиксируем значение теоретической ошибки
    terr2c1(tt) = PIJ(2, 1); % первый класс второй род
    % фиксируем значения экспериментальных ошибок
    err2c1_p(tt) = Pcv(2, 1); % значение ошибки при оценке к-соседей
    err2c1(tt) = Pc_(2,1);% значение ошибки (3 лаба)
end % конец цикла по расстояниям между классами
```

```
%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(:,:,i)); %массив значений плотности распределения
 f3=reshape(f2,length(x2),length(x1));%матрица значений плотности
 [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_(:,:,j)*(m(:,i)-m(:,j));
 wij0=-0.5*(m(:,i)+m(:,j))'*C_(:,:,j)*(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;

%5.Визуализация поверхностей плотностей распределения
x1 = 10:1:100;
x2 = -30:1:60;
[Y,X] = meshgrid(x1,x2);
x = [reshape(X.',1,[]);reshape(Y.',1,[])];
x_ = x.';
p = 0;
p2x = 0;
k = 100;
for i=1:M
      p = p + mvnpdf(x_,m(:,i)',C(:,:,i));
      p2x = p2x + vknn(x, XN{i}, 400);
end;
pi = reshape(p,[length(x2),length(x1)]);
p2x = reshape(p2x,[length(x2),length(x1)]);

disp(pi);
figure;
surf(X,Y,pi);
title('Поверхность плотности распределения классов (теоретически)');
figure;
surf(X,Y,p2x);
title('Поверхность плотности распределения классов (с оценкой)');



% ТЕПЕРЬ визуализация зависимостей ошибок от расстояния между классами
figure; hold on;     % новое графическое окно + режим дорисовки
plot(DM, err2c1);    % график экспериментальной ошибки (лабораторная №3)
```

```
plot(DM, err2c1_p);   % график экспериментальной ошибки (с оценками k-соседей)
plot(DM, terr2c1, 'g');% теоретическое значение ошибки (первый класс первый род)
title('Зависимость ошибки 2-го рода от расстояния между классами (первый класс)');
xlabel('Расстояние между классами','FontName','Courier');
ylabel('Значение ошибки 2-го рода','FontName','Courier');
legend('Экспериментальная ошибка(3 лабораторная)','Экспериментальная ошибка(при оценке к-соседей)', 'Теоретическое значение ошибки');
hold off;

% Теоретическая матрица ошибок
disp('Теоретическая матрица вероятностей ошибок');disp(PIJ);
% экспериментальные матрицы ошибок
disp('Экспериментальная матрица вероятностей ошибок (с оценками k-соседей)');disp(Pcv);
disp('Экспериментальная матрица вероятностей ошибок (лабораторная №3)');disp(Pc_);
```

Результаты выполнения задания

В данной лабораторной работе, подобно л.р. №3, мы работаем с данными:

$\boldsymbol{\mu}1$=[2 7], $\boldsymbol{\mu}2$=[1 10], C1=[4 -2; -2 4], C2=[5 1; 1 5]

1. Согласно заданию лабораторной №3, необходимо найти <u>зависимость ошибки второго рода от расстояния между классами.</u> Для этого найдем экспериментальную матрицу ошибок с помощью оценки k-соседей и сравним с матрицей, вычисленной в лабораторной №3.

Расстояние между классами растет:

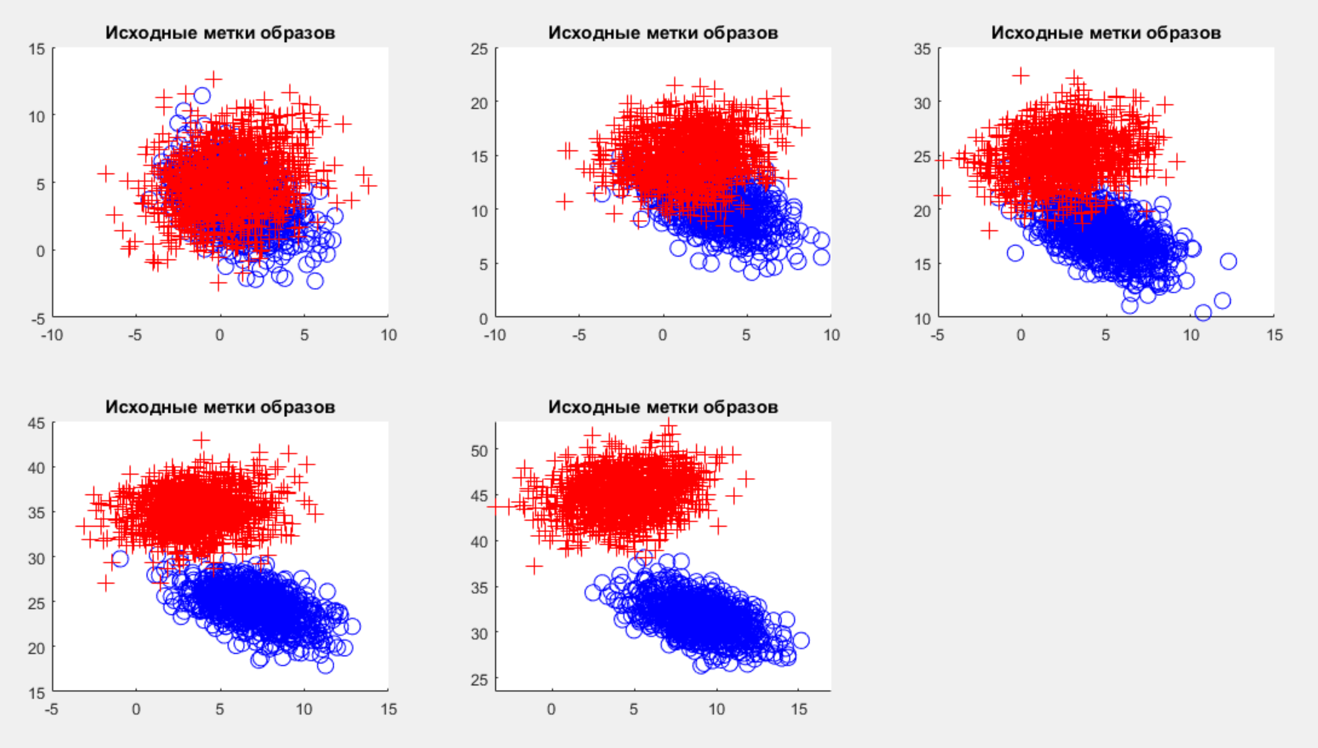

Рис.1.

При этом график зависимости выглядит следующим образом:

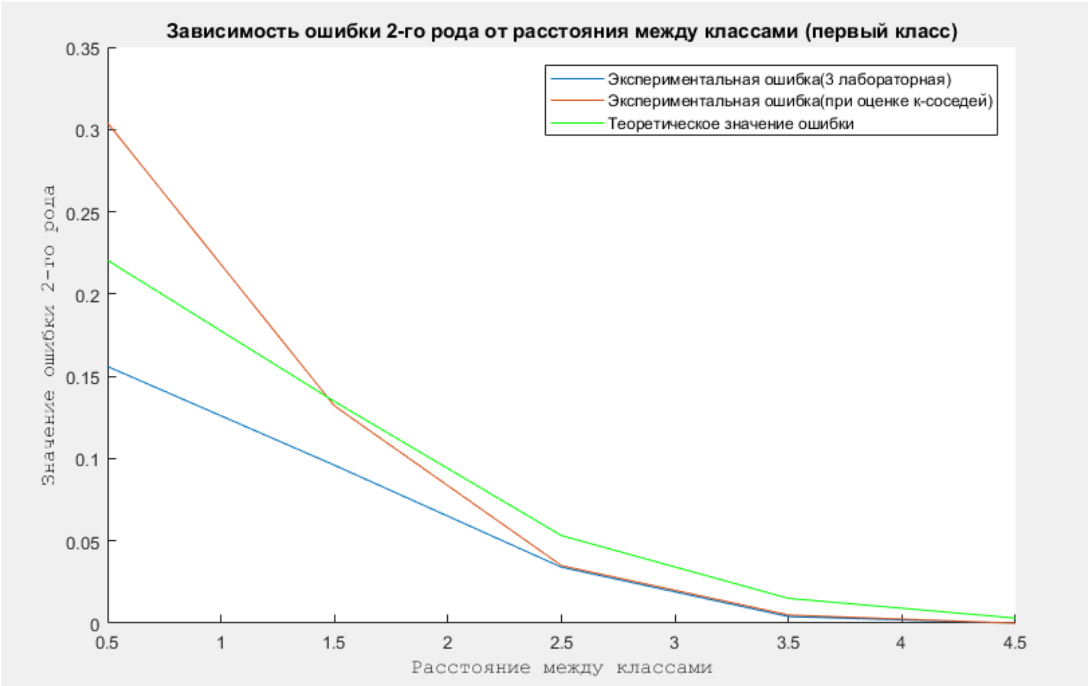

Рис.2.

При увеличении расстояния ошибка второго рода должна падать, что и отображает рис.2.

По графическому отображению результатов на рис.2. видно, что значение ошибки второго рода при экспериментальной матрице вероятностей ошибок, построенной с помощью оценки k-соседей уступает точности экспериментального метода в лабораторной №3. Однако, нужно заметить, значение ошибки второго рода при матрице с оценкой стремится к нулю быстрее, чем при теоретической матрице.

2. <u>Сравнение экспериментальных вероятностей ошибок:</u>

Для сравнения выведем данные:

- Теоретической матрицы ошибок
- Экспериментальной матрицы ошибок (из лабораторной №3)
- Экспериментальной матрицы ошибок (с помощью оценки)

```
Теоретическая матрица вероятностей ошибок
    0.9985    0.0015
    0.0031    0.9969

Экспериментальная матрица вероятностей ошибок (с оценками k-соседей)
    0.9990    0.0010
         0    1.0000

Экспериментальная матрица вероятностей ошибок (лабораторная №3)
     1     0
     0     1
```

Эти данные подтверждают рассуждения из пункта 1.

3. <u>Отображение поверхностей плотностей распределения классов</u>

Здесь мы рассматриваем конечное расстояние между классами.
Для начала отобразим области локализации образов (рис.3) и возьмем из п.1 визуализацию совокупности образов для конечного расстояния (рис.4).

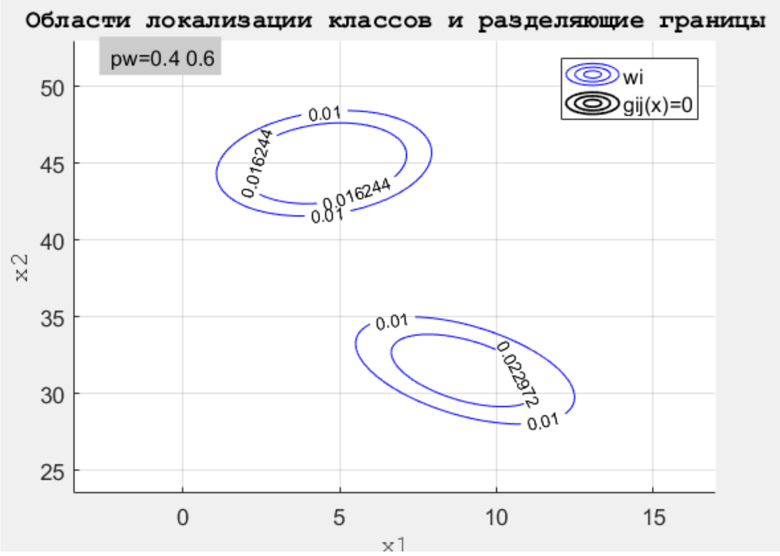

Рис.3

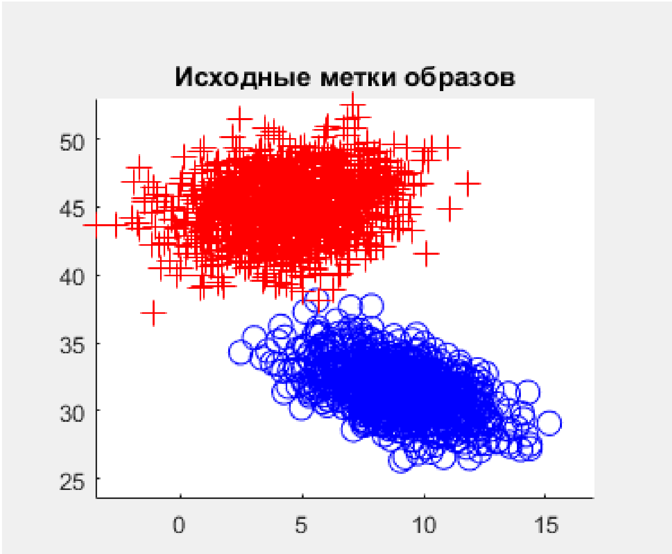

Рис.4

На рисунках видно координаты, в которых локализуются классы. Рассмотрим поверхность распределения на сетке, границы которых возьмем в зависимости от данных результатов отображения локализации.

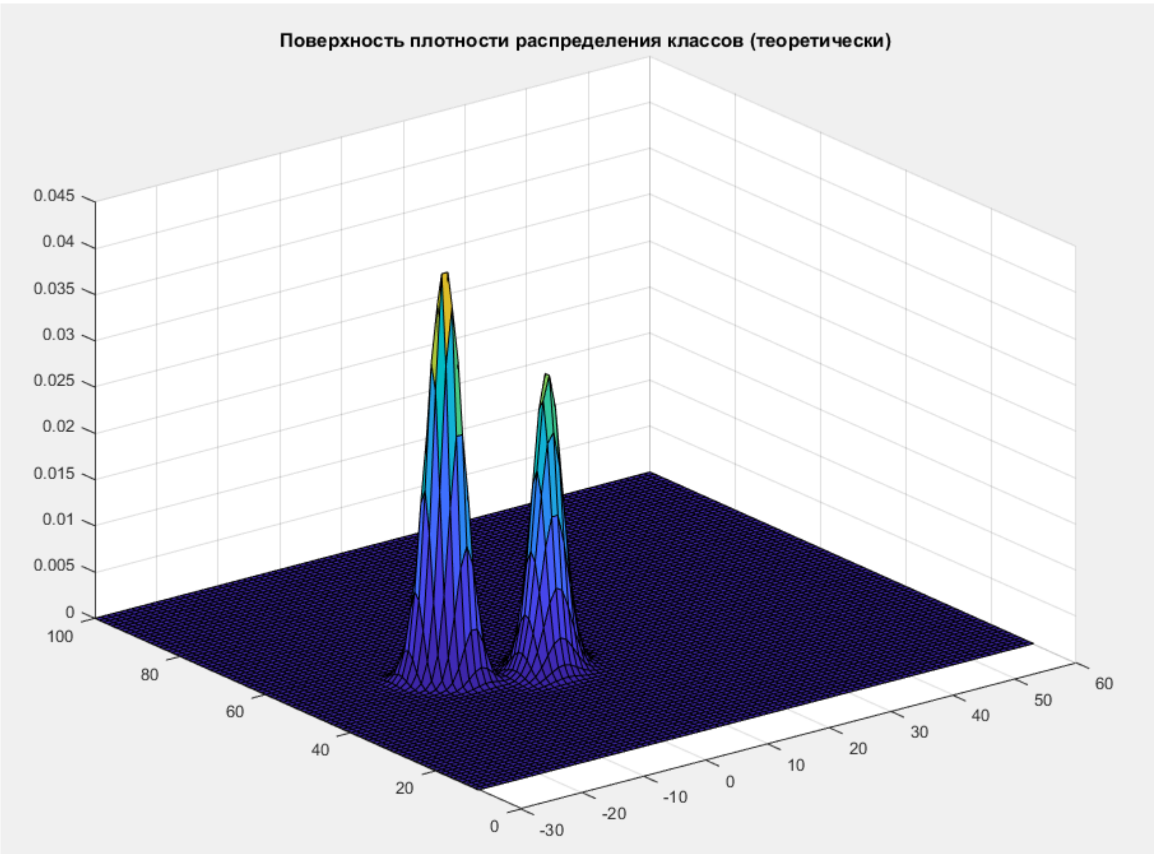

Рис.5

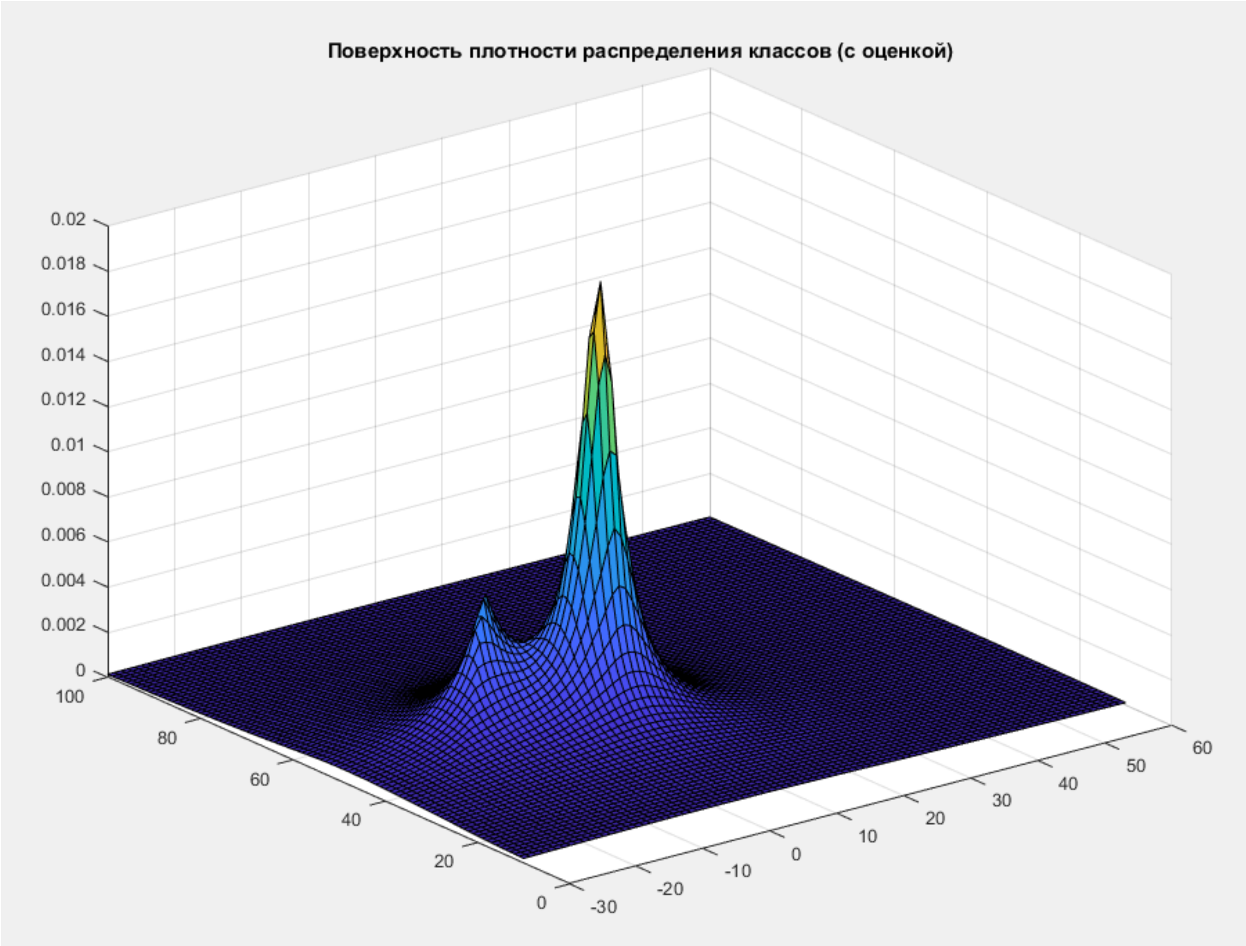

рис.6.

И отобразим результаты вместе для сравнения:

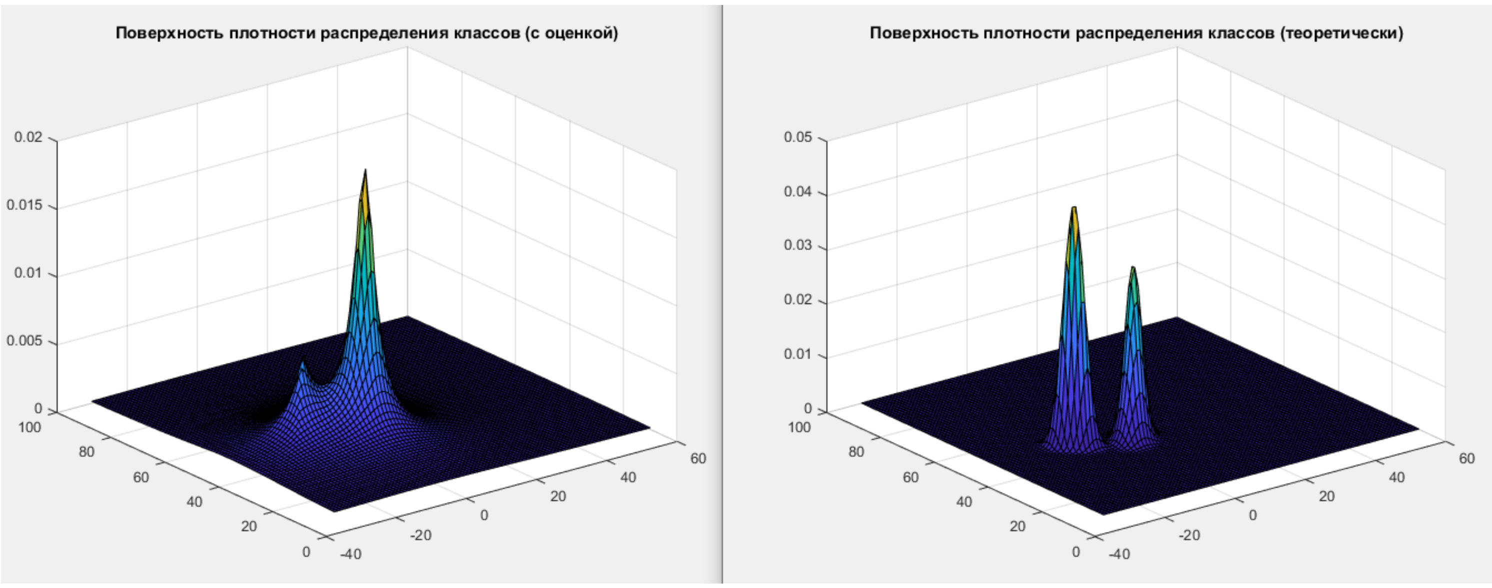

рис.7.

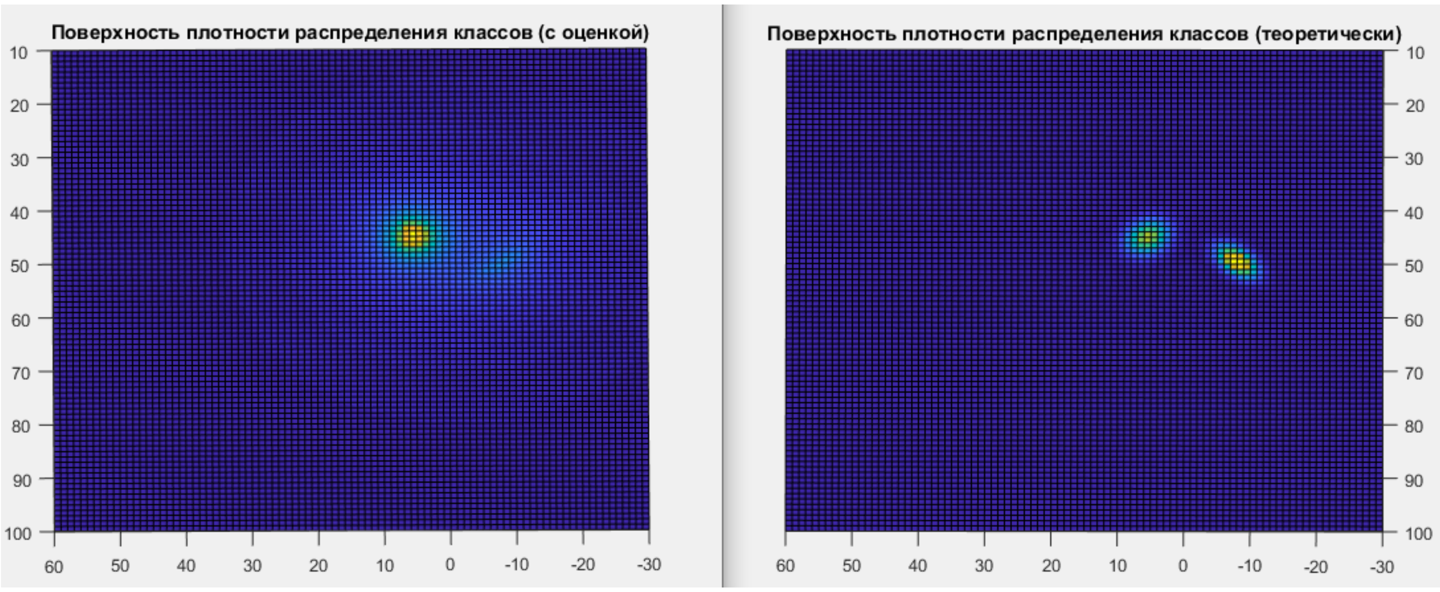

рис.8.

Анализ рисунков дает понять, что:

1. На обоих графиках плотности распределения классов расположены правильно, согласно областям локализации.
2. Поверхность теоретической плотности распределения классов дает наилучший результат, что видно по высоте плотностей распределения классов, а именно: Второй класс должен быть выше, чем первый, поскольку априорная вероятность второго больше. На графике поверхности плотности распределения с оценкой этот класс ниже первого.

Дополнительное измерение:

Поскольку результат оценивания k-соседей напрямую зависит от числа выборки, попробуем увеличить это число до 5000 и сравним результаты. Области локализации классов примут следующий вид:

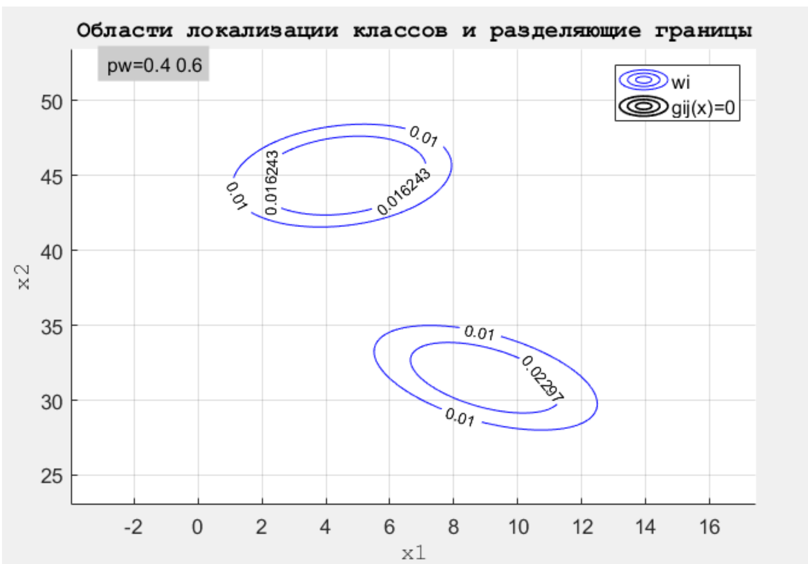

Рис.9.

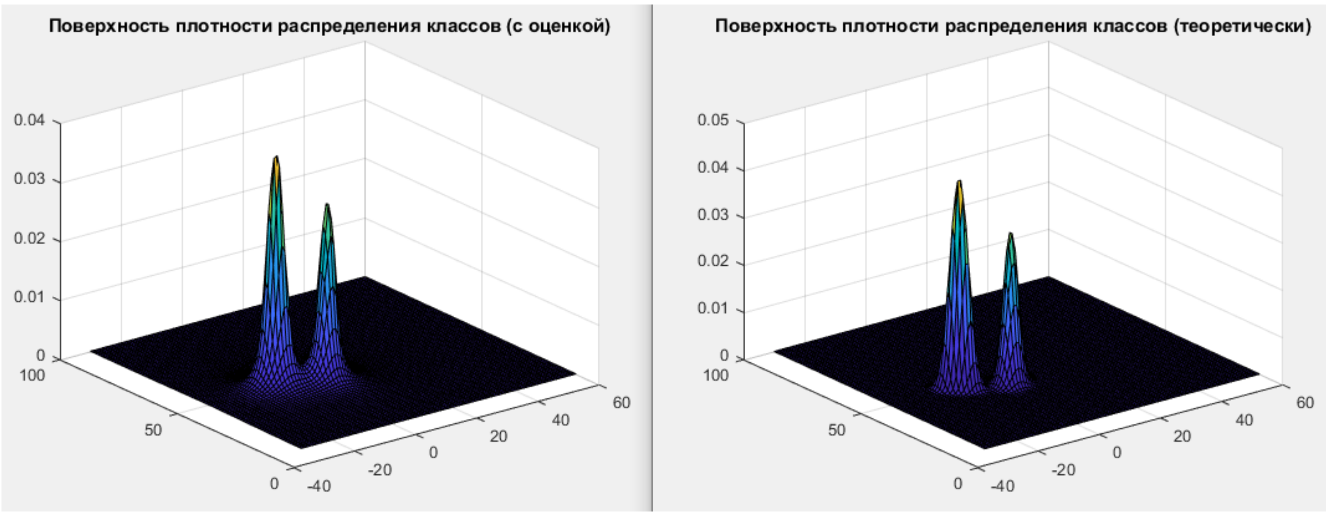

Рис.10

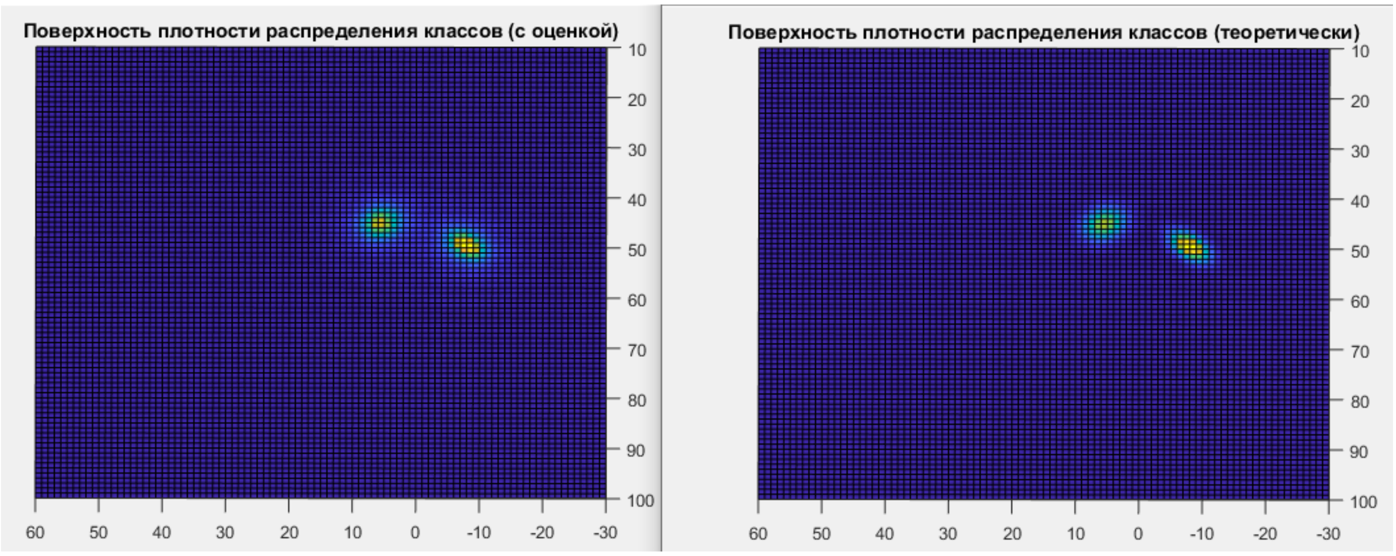

Рис.11.

По рисункам 10-11 видно, что распознавание с помощью оценки k-соседей сильно улучшилось благодаря увеличению объема выборки. Классы расположены верно и априорные вероятности классов в результате совпадают с заданными.

Выводы:

В результате лабораторной работы были исследованы алгоритмы распознавания образов на основе оценивания плотности распределения случайных величин и случайных векторов при использовании метода k ближайших соседей.