Кравцова Александра Николаевна, группа 12.1

Лабораторная работа № 7

**Вариант №** **2**

Распознавание образов с использованием машины опорных векторов

**Цель работы**

Исследовать алгоритмы распознавания образов на основе аппарата машины опорных векторов (Support Vector Machine).

**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

Воспользовавшись классификатором SVM, вычислите вероятности ошибки при классификации линейно разделимых выборок трёх классов (генерацию выборок можно реализовать на основе лаб.2/ лаб.3)

Код программы (Желтым выделены отличные от шаблона фрагменты):

%Файл pr53\_rec\_gaus\_uneq. Синтез и анализ алгоритмов распознавания ГСВ с

%различными матрицами ковариации

clear all;

close all;

% Построить график зависимости суммарной экспериментальной ошибки

% первого рода ( для третьего класса) от числа испытаний (объема выборки ).

% Сравнить с теоретическим значением.

DM = 1 : 5;

err2CHERNOV3 = zeros(size(DM));

terr2c13 = zeros(size(DM));

err2c3\_= zeros(size(DM));

err2c3 = zeros(size(DM));

K = 1000; % или DM = 1; или PW = 0.5;

% Добавляется цикл по объемам выборки

for tt = 1: numel(DM) % цикл по объемам выборки

%1.Задание исходных данных

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

%Априорные вероятности, математические ожидания и матрицы ковариации классов

dm=3.0;%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 0.5];

pw=pw/sum(pw);

m=[2 -3; 1 10; -2 -1]';

m=m\*DM(tt); % применение очередного масштаба

C(:,:,1)=[4 -2; -2 4];

C(:,:,2)=[5 1; 1 5];

C(:,:,3)=[5 -3; -3 5];

for k=1:M

C\_(:,:,k)=C(:,:,k)^-1;

end

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

% + ДОБАВИТЬ этот пункт (или скопировать из 6й лабы)

% 1.1. Генерация обучающих выборок классов

% Объемы выборок 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 Ks(i) образов i-го класса

end;

% + ДОБАВИТЬ Этот кусок (до п.2)

% 1.2. Сначала обучаем классификаторы, чтобы на эксперименте дёргать уже обученные.

% Поскольку классов может быть 3, а svm осуществляет только попарное

% сравнение, то обучаем свой классификатор для каждой пары классов

% (для сравнения 1го и 2го, 2го и 3го, 1го и 3го классов)

% Обучаем svm-классификаторы для каждой пары классов

show = true; % визуализация результатов обучения (если n=3, можно установить false - 3d пространство не визуализируется)

r = 0.1; % параметр регуляризации (балансирует положение разделяющей границы относительно положения опорных векторов и "объему" заступов за границу)

svm\_strs = cell(M);

for i=1:M-1, % цикл по парам классов (как в 4й лабе с 3мя буквами)

for j=i+1:M,

% Формирование смешанной обучающей выборки

data = [XN{i}'; XN{j}']; % помещаем образы

groups = [true(Ks(i), 1); false(Ks(j), 1)]; % метки классов

if show, figure; end % создаём графическое окно, если нужно

% Вар. а) Классификатор для линейно разделимых данных (разделяющая граница - прямая)

svm\_strs{i,j} = svmtrain(data, groups, 'boxconstrain', r, 'Method', 'QP', 'showplot', show);

if show, title(sprintf('Классы %d и %d при расстоянии = %f', i, j, DM(tt))); end % подпись, если включена визуализация

end;

end;

%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));

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

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

Pcv=zeros(M); % + инициализация экспериментальной матрицы ошибок

x=ones(n,1); u=zeros(M,1);

Pc\_=zeros(M);%экспериментальная матрица вероятностей ошибок из 3й лабы

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 из 3й лабы

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

% + ДОБАВИТЬ Этот кусок (до завершения цикла по классам)

% Прогон по всем классификаторам в том же порядке

% как мы их обучали

iais = []; % массив для фиксации результатов от разных классификаторов

for ii=1:M-1 % цикл по парам классов

for jj=ii+1:M

% Вызываем классификатор для сравнения ii-го и jj-го

% классов из массива svm\_strs и распознаём им образ x

cl = svmclassify(svm\_strs{ii,jj}, x', 'showplot', false);

if cl % cl будет true, если класс ii-й

iai = ii;

else % cl будет false, если класс jj-й (потому что так формировался массив groups при обучении)

iai = jj;

end

iais = [iais, iai]; % фиксируем результат распознавания

end

end

iai = mode(iais); % выбираем класс, за который проголосовало большинство классификаторов

Pcv(i,iai)=Pcv(i,iai)+1;% %фиксация результата распознавания

end; %цикл по классам

end;

Pc\_=Pc\_/K;

Pcv=Pcv/K; % + нормировка экспериментальной матрицы по svm на число испытаний

% + ЕСЛИ В 3й лабе нужно было считать зависимость ошибки от KK(или DM или PW)

% ТО комментим фиксацию ошибок

% % В конце шага цикла фиксируем очередное значение ошибки

fprintf('Расстояние между классами: %s\n', vpa(DM(tt), 3));

% В конце шага цикла фиксируем значение ошибки по Чернову

err2CHERNOV3(tt) = PIJB(3, 1) + PIJB(3, 2);

% фиксируем значение теоретической ошибки

terr2c13(tt) = PIJ(3, 1) + PIJ(3, 2);

% фиксируем значение эксперементальных ошибок

err2c3(tt) = Pcv(1, 3) + Pcv(2, 3); %(svm)

err2c3\_(tt) = Pc\_(1, 3) + Pc\_(2, 3); % (3 лабораторная)

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

disp('Матрица вероятностей ошибок на основе границы Чернова');disp(PIJB);

disp('Экспериментальная матрица вероятностей ошибок (3 лабораторная)');disp(Pc\_);

disp('Экспериментальная матрица вероятностей ошибок (svm)');disp(Pcv);

end % конец цикла по объемам выборки

% Визуализация зависимостей ошибок от объема выборки

figure; hold on; % новое графическое окно + режим дорисовки

plot(DM, err2CHERNOV3, 'y'); % по Чернову

plot(DM, terr2c13, 'r'); % теоретическое значение ошибки

plot(DM, err2c3, 'g'); % (svm)

plot(DM, err2c3\_, 'b'); % (3 лабораторная)

title('Зависимость ошибки 2-го рода от расстояния между классами (первый класс)');

xlabel('Расстояние между классами','FontName','Courier');

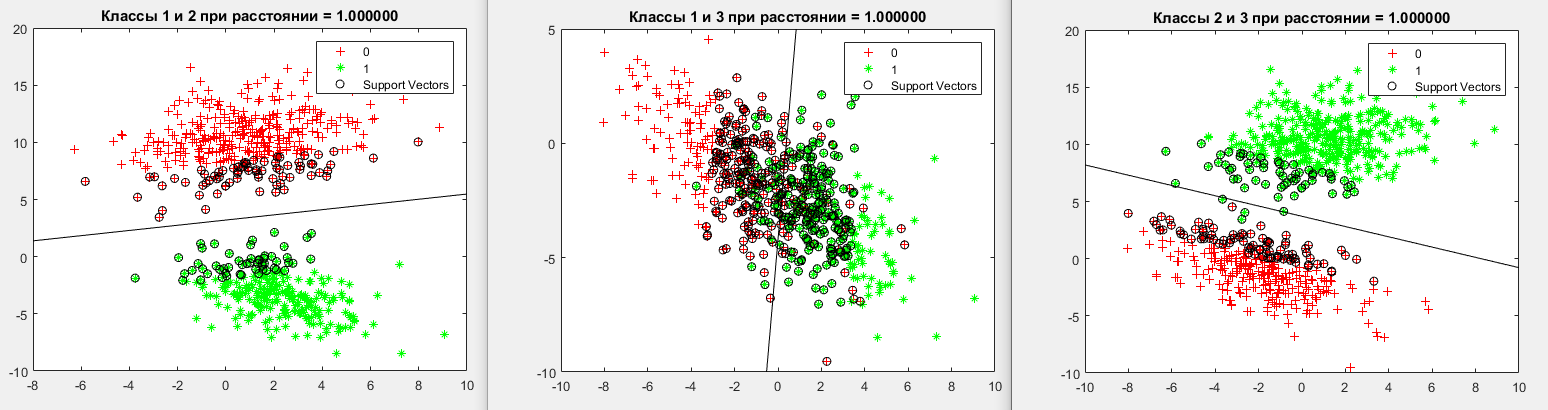
ylabel('Значение ошибки 2-го рода','FontName','Courier');

legend('по Чернову', 'Теоретическое значение', 'Экспериментальная (svm)', 'Экспериментальная (3 лаб.)');

hold off;

**Результаты выполнения задания**

Расстояние между классами: 1.0:

1. **Классификация**

*Рис.1.*

1. **Матрицы ошибок:**

Теоретическая матрица вероятностей ошибок

0.8033 0.0023 0.1944

0.0068 0.9801 0.0131

0.1589 0.0039 0.8372

Матрица вероятностей ошибок на основе границы Чернова

0.2774 0.0108 0.7118

0.0072 0.9711 0.0217

0.5694 0.0260 0.4046

Экспериментальная матрица вероятностей ошибок (svm)

0.8150 0.0010 0.1840

0.0010 0.9960 0.0030

0.1510 0.0020 0.8470

Экспериментальная матрица вероятностей ошибок (3 лабораторная)

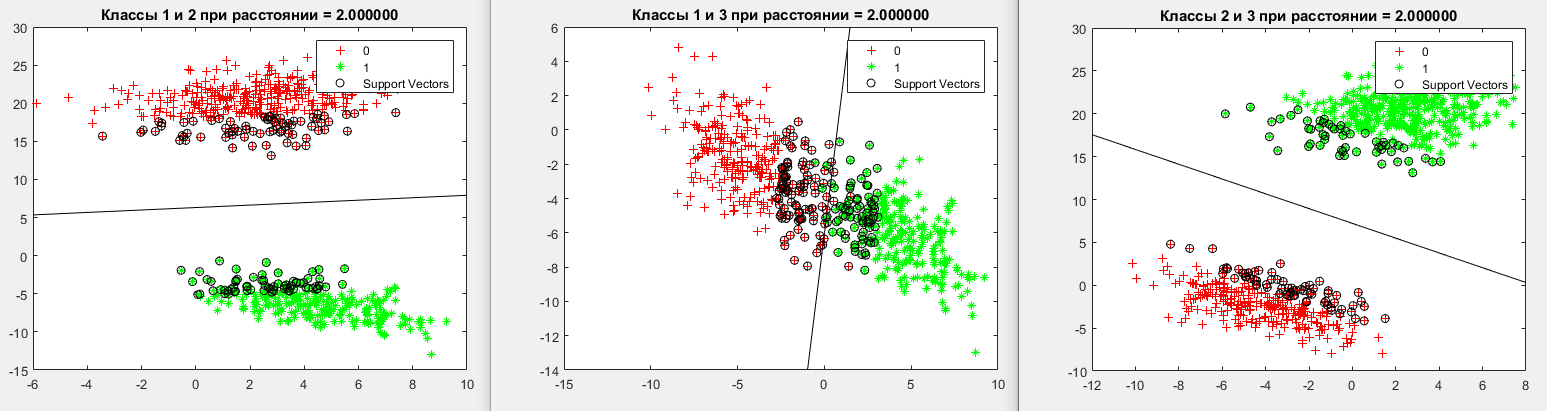
0.8230 0 0.1770

0.0010 0.9910 0.0080

0.1640 0.0010 0.8350

При данном расстоянии все методы распознавания имеют ошибки, однако, с распознаванием 2-го и 3-его классов наилучшим образом справляется метод svm.

Расстояние между классами: 2.0:

1. **Классификация**

*Рис.2.*

1. **Матрицы ошибок:**

Теоретическая матрица вероятностей ошибок

0.9724 0.0000 0.0276

0.0000 1.0000 0.0000

0.0328 0.0000 0.9672

Матрица вероятностей ошибок на основе границы Чернова

0.8135 0.0000 0.1865

0.0000 1.0000 0.0000

0.1492 0.0000 0.8508

Экспериментальная матрица вероятностей ошибок (svm 3)

0.9620 0 0.0380

0 1.0000 0

0.0280 0 0.9720

Экспериментальная матрица вероятностей ошибок (лабораторная)

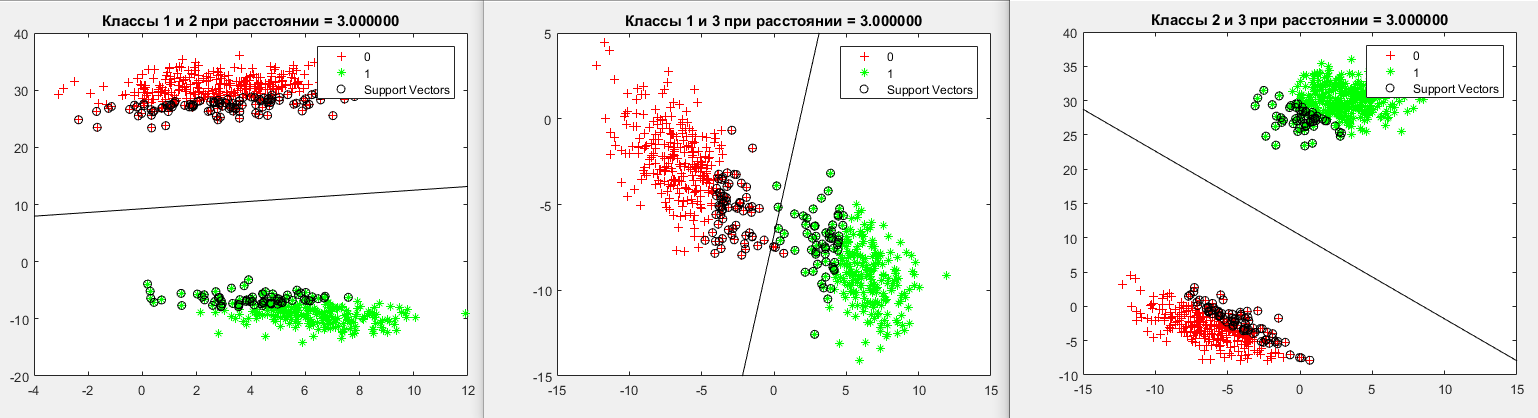
0.9640 0 0.0360

0 1.0000 0

0.0330 0 0.9670

В данном случае распознавание 1-го класса с помощью метода svm уступает теоретическому и справился значительно лучше метода по Чернову. В сравнении с экспериментальными значениями 3-ей лабораторной, метод svm дает лучший результат при распознавании 1-его класса, однако уступает в распознавании 3-ого класса. Поскольку 2-ой класс сильно удален от 1-го и 3-го классов, его распознавание ни у одного из методов не вызывает ошибок.

Расстояние между классами: 3.0:

1. **Классификация**

*Рис.3.*

1. **Матрицы ошибок:**

Теоретическая матрица вероятностей ошибок

0.9983 0.0000 0.0017

0.0000 1.0000 0.0000

0.0033 0.0000 0.9967

Матрица вероятностей ошибок на основе границы Чернова

0.9800 0.0000 0.0200

0.0000 1.0000 0.0000

0.0160 0.0000 0.9840

Экспериментальная матрица вероятностей ошибок (svm 3)

0.9980 0 0.0020

0 1.0000 0

0.0040 0 0.9960

Экспериментальная матрица вероятностей ошибок (лабораторная)

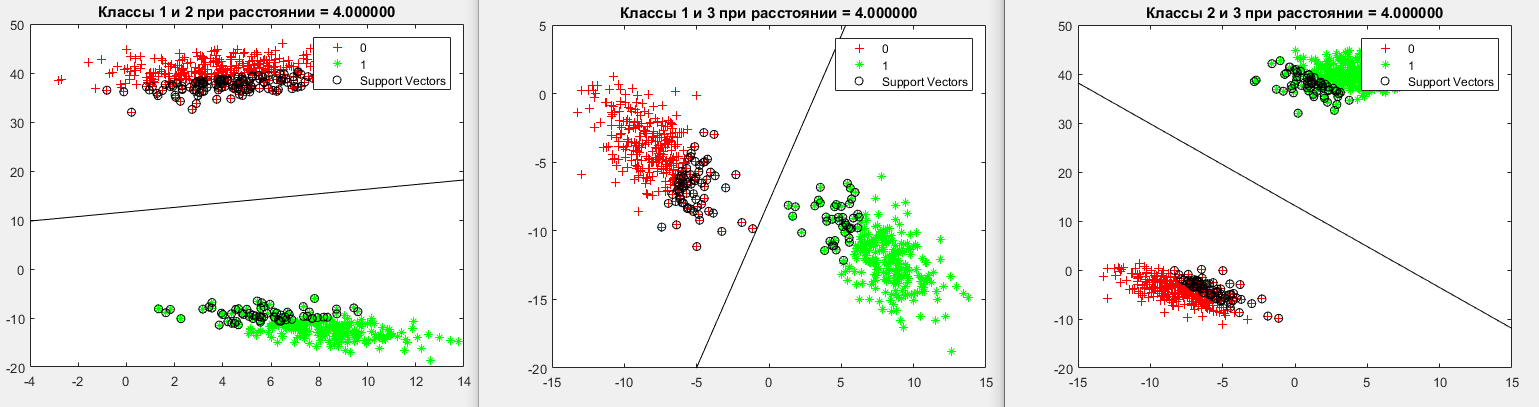
0.9990 0 0.0010

0 1.0000 0

0.0050 0 0.9950

В данном случае, из-за близкого расстояния классов 1 и 3 (как показано на рис.3.), возникают ошибки распознавания. На рисунке видно, что данные классы разделены неверно.

Расстояние между классами: 4.0:

1. **Классификация**

*Рис.4.*

1. **Матрицы ошибок:**

Теоретическая матрица вероятностей ошибок

1.0000 0.0000 0.0000

0 1.0000 0.0000

0.0002 0 0.9998

Матрица вероятностей ошибок на основе границы Чернова

0.9991 0.0000 0.0009

0.0000 1.0000 0.0000

0.0007 0.0000 0.9993

Экспериментальная матрица вероятностей ошибок (svm 3)

1 0 0

0 1 0

0 0 1

Экспериментальная матрица вероятностей ошибок (лабораторная)

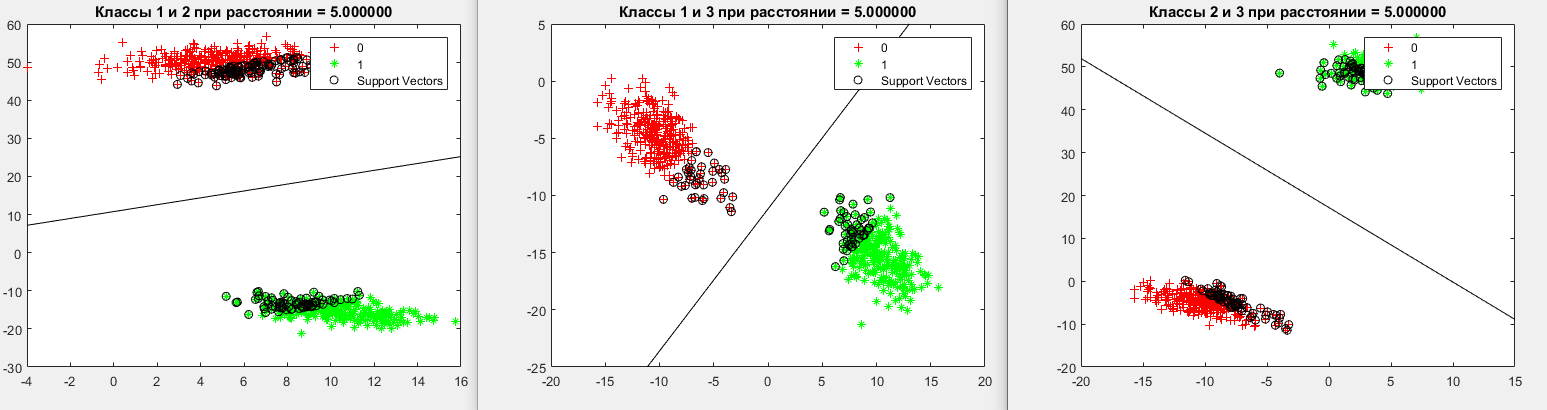
1.0000 0 0

0 1.0000 0

0.0010 0 0.9990

При данном значении межклассового расстояния в методе классификации с помощью svm не возникает никаких ошибок. Все классы распределены верно, что видно рис.4. В данном случае метод svm справился с распознаванием классов лучше других методов.

Расстояние между классами: 4.0:

1. **Классификация**

*Рис.5.*

1. **Матрицы ошибок:**

Теоретическая матрица вероятностей ошибок

1.0000 0.0000 0.0000

0 1.0000 0.0000

0.0000 0 1.0000

Матрица вероятностей ошибок на основе границы Чернова

1.0000 0.0000 0.0000

0.0000 1.0000 0.0000

0.0000 0.0000 1.0000

Экспериментальная матрица вероятностей ошибок (svm)

1 0 0

0 1 0

0 0 1

Экспериментальная матрица вероятностей ошибок (3 лабораторная)

1 0 0

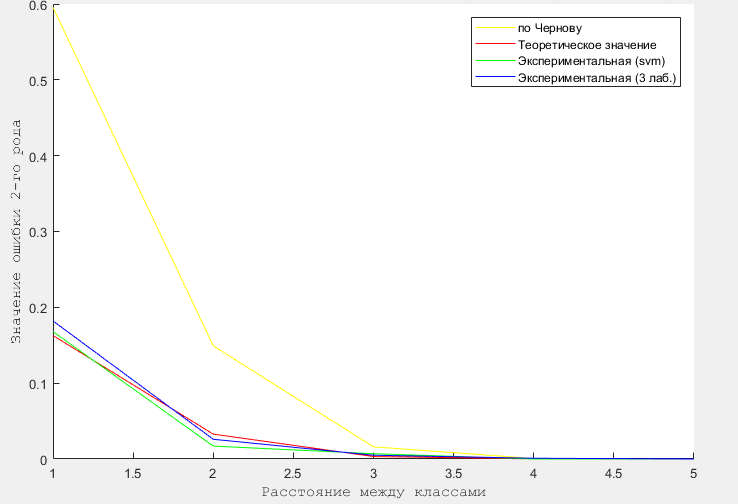
0 1 0

0 0 1

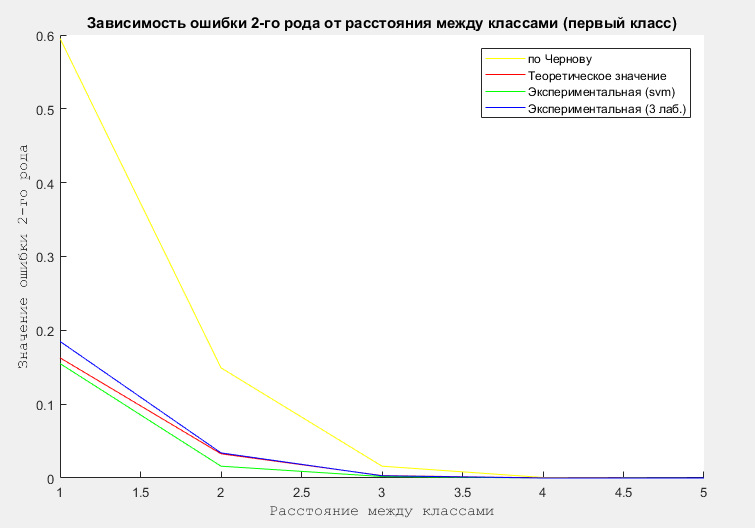
В данном случае все методы распознавания образов трех классов работают безошибочно. Это объяснимо большим значением межклассового расстояния.

**График зависимости ошибки 2-го рода от расстояния между классами (для 3-его класса)**

Посмотрим на значения ошибок при разных генерациях.

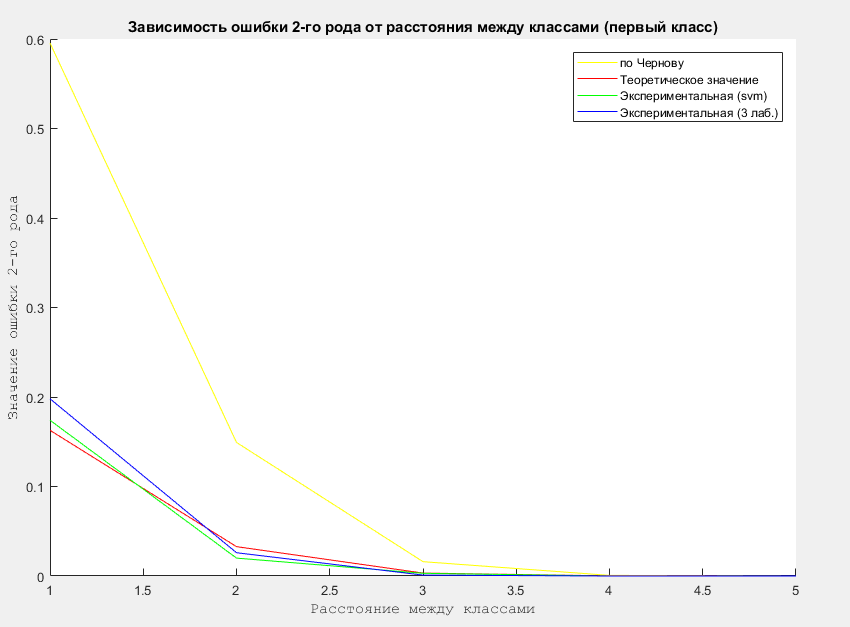
****Генерация №1**

*Рис.6.*

**Генерация №2**

*Рис.7.*

**Генерация №3**

**

*Рис.8.*

По графикам на рисунках 6-7 видно, что распознавание образов с помощью svm (зеленая кривая) дает наилучшие результаты, поскольку значение ошибки для третьего класса дает наименьшие значения.

**Исследование иного метода реализации SVM:**

**Код программы:**

%%% Пример переделки лабы 3 (ГСВ с различными ков. матрицами) на svm.

% Получение экспериментальной матрицы ошибок по svm

% Пример для 3х классов - попарное сравнение каждый с каждым.

%Воспользовавшись классификатором SVM, вычислите вероятности ошибки при классификации линейно

%разделимых выборок трёх классов (генерацию выборок можно реализовать на основе лаб.2/ лаб.3)

clear all;

close all;

DM = 1 : 5;

err2CHERNOV3 = zeros(size(DM));

terr2c13 = zeros(size(DM));

err2c3\_= zeros(size(DM));

err2c3 = zeros(size(DM));

%% 1. Задание исходных данных

n = 2; % размерность признакового пространства

M = 3; % число классов

K = 1000; % количество статистических испытаний

for tt = 1: numel(DM) % цикл по объемам выборки

% Априорные вероятности, математические ожидания и матрицы ковариации классов

C=zeros(n,n,M); C\_=C;%матрица ковариации вектора признаков различных классов

pw=[0.4 0.6 0.5];

pw=pw/sum(pw);

m=[2 -3; 1 10; -2 -1]';

m=m\*DM(tt);

C(:,:,1)=[4 -2; -2 4];

C(:,:,2)=[5 1; 1 5];

C(:,:,3)=[5 -3; -3 5];

for k=1:M

C\_(:,:,k)=C(:,:,k)^-1;

end

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

%% 2. Обучение svm классификаторов для каждой пары классов

% 2.1. Генерация обучающих выборок классов

% Объемы выборок каждого класса

Ks = fix(K \* pw);

Ks(end) = K - sum(Ks(1 : end - 1));

X = []; % общая обучающая выборка (все образы всех классов)

Y = []; % номера классов для каждого образа

% Генерация выборок

for i=1:M, % цикл по классам

XN{i} = repmat(m(:,i), [1, Ks(i)]) + randncor(n,Ks(i),C(:,:,i)); %генерация Ks(i) образов i-го класса

X = cat(1, X, XN{i}'); % помещаем образы в общую выборку

Y = cat(1, Y, i \* ones(Ks(i), 1)); % номер класса для каждого образа

end

%% 2.2. Сначала обучаем классификаторы, чтобы на эксперименте дёргать уже обученные.

% Поскольку классов может быть 3, а svm осуществляет только попарное

% сравнение, то обучаем свой классификатор для каждой пары классов

% (для сравнения 1го и 2го, 2го и 3го, 1го и 3го классов)

% Обучаем svm-классификаторы для каждой пары классов

r = 1; % параметр регуляризации (балансирует положение разделяющей границы относительно положения опорных векторов и "объему" заступов за границу)

svm\_strs = cell(M);

for i = 1 : M - 1 % цикл по парам классов (как в 4й лабе с 3мя буквами)

for j = i + 1 : M

% Формирование смешанной обучающей выборки для пары классов

Xij = [XN{i}'; XN{j}']; % помещаем образы 2х классов в смешаную выборку

D = [true(Ks(i), 1); false(Ks(j), 1)]; % метки классов

% Вар. а) Классификатор для линейно разделимых данных (разделяющая граница - прямая)

svm\_strs{i,j} = fitcsvm(Xij,D,'Solver','L1QP','KernelFunction','linear','BoxConstraint',r);

% Вар. б) Классификатор для линейно неразделимых данных (разделяющая граница - кривая)

% svm\_strs{i,j} = fitcsvm(Xij,D,'Solver','L1QP','KernelFunction','polynomial','PolynomialOrder',4);

% Другие варианты Kernel\_Function и её параметров (смотри файл fitcsvm.m)

svm\_strs{i,j} = fitcsvm(Xij,D,'Solver','L1QP','KernelFunction','rbf','KernelScale',1);

end

end

%% 2.3. Отображение областей локализации классов

show = true; % визуализация результатов обучения (если n=3, можно установить false - 3d пространство не визуализируется)

if show

% 2.3.1 Формируем дискретную двумерную сетку отсчётов

d = 0.05; % шаг сетки отсчётов

[x1Grid,x2Grid] = meshgrid(min(X(:,1)):d:max(X(:,1)),...

min(X(:,2)):d:max(X(:,2)));

xGrid = [x1Grid(:),x2Grid(:)];

N = size(xGrid,1);

idxs = []; % индексы классов

sv = []; % массив опорных векторов

% 2.3.2 Классифицируем узлы дискретной сетки

for i=1:M-1 % цикл по парам классов (как в 4й лабе с 3мя буквами)

for j=i+1:M

sv = cat(1, sv, svm\_strs{i,j}.SupportVectors); % опорные векторы

cls = predict(svm\_strs{i,j},xGrid); % логические метки классов (true и false)

% переводим метки в числовые индексы классов

idx = i \* cls + j \* ~cls; % если метки == true - то это i-й класс, если false - то j-й

idxs = cat(2, idxs, idx); % фиксируем индексы классов от всех классификаторов

end

end

iai = mode(idxs,2); % выбираем превалирующий по строке номер класса для каждого отсчёта

% 2.3.3 Отрисовка узлов двумерной сетки разными цветами

figure; % создаём графическое окно

if M == 2

h(1:M) = gscatter(xGrid(:,1), xGrid(:,2), iai, [0.5 0.1 0.5; 0.1 0.5 0.5]);

elseif M == 3

h(1:M) = gscatter(xGrid(:,1), xGrid(:,2), iai, [0.5 0.1 0.5; 0.5 0.5 0.1; 0.1 0.5 0.5]);

end

hold on

% 2.3.4 Отрисовка образов обучающих выборок

h(M+1:2\*M) = gscatter(X(:,1),X(:,2),Y);

axis tight

% 2.3.5 Отображение опорных векторов и подпись

plot(sv(:,1), sv(:,2), 'ko', 'MarkerSize', 10)

if M == 2

legend('class 1 region', 'class 2 region', 'class 1', 'class 2', 'Support Vector');

elseif M == 3

legend('class 1 region', 'class 2 region', 'class 3 region',...

'class 1', 'class 2', 'class 3', 'Support Vector');

end

hold off

end % подпись, если включена визуализация

%% 3. Расчет теоретических 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));

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

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,:)); % нижняя граница вероятности правильного распознавания

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

end;

%% 4. Тестирование алгоритма методом статистических испытаний

Pcv=zeros(M); % + инициализация экспериментальной матрицы ошибок

x=ones(n,1); u=zeros(M,1);

Pc\_=zeros(M);%экспериментальная матрица вероятностей ошибок из 3й лабы

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 из 3й лабы

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

% Прогон по всем классификаторам SVM в том же порядке как мы их обучали

iais = []; % массив для фиксации результатов от разных классификаторов

for ii=1:M-1 % цикл по парам классов

for jj=ii+1:M

% Вызываем классификатор для сравнения ii-го и jj-го

% классов из массива svm\_strs и распознаём им образ x

cl = predict(svm\_strs{ii,jj}, x');

if cl % cl будет true, если класс ii-й

iai = ii;

else % cl будет false, если класс jj-й (потому что так формировался массив меток при обучении)

iai = jj;

end

iais = [iais, iai]; % фиксируем результат распознавания

end

end

iai = mode(iais); % выбираем класс, за который проголосовало большинство классификаторов

Pcv(i,iai)=Pcv(i,iai)+1;% %фиксация результата распознавания

end %цикл по классам

end

Pc\_=Pc\_/K;

Pcv=Pcv/K; % + нормировка экспериментальной матрицы по svm на число испытаний

fprintf('Расстояние между классами: %s\n', vpa(DM(tt), 3));

% В конце шага цикла фиксируем значение ошибки по Чернову

err2CHERNOV3(tt) = PIJB(3, 1) + PIJB(3, 2);

% фиксируем значение теоретической ошибки

terr2c13(tt) = PIJ(3, 1) + PIJ(3, 2);

% фиксируем значение эксперементальных ошибок

err2c3(tt) = Pcv(1, 3) + Pcv(2, 3); %(3 лабораторная)

err2c3\_(tt) = Pc\_(1, 3) + Pc\_(2, 3); % (svm)

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

disp('Матрица вероятностей ошибок на основе границы Чернова');disp(PIJB);

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

disp('Экспериментальная матрица вероятностей ошибок SVM');disp(Pcv);

end

% Визуализация зависимостей ошибок от объема выборки

figure; hold on; % новое графическое окно + режим дорисовки

plot(DM, err2CHERNOV3, 'y'); % по Чернову

plot(DM, terr2c13, 'r'); % теоретическое значение ошибки

plot(DM, err2c3, 'g'); % (3 лабораторная)

plot(DM, err2c3\_, 'b'); % (svm)

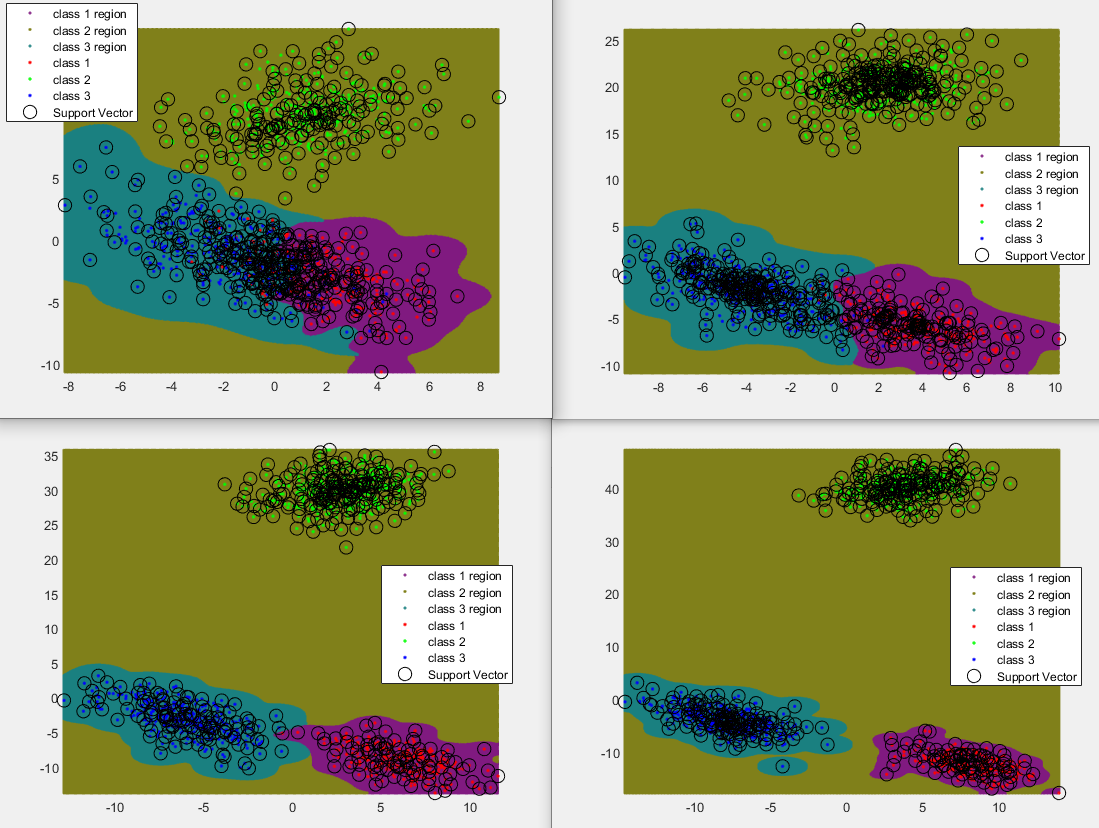
title('Зависимость ошибки 2-го рода от расстояния между классами (первый класс)');

xlabel('Расстояние между классами','FontName','Courier');

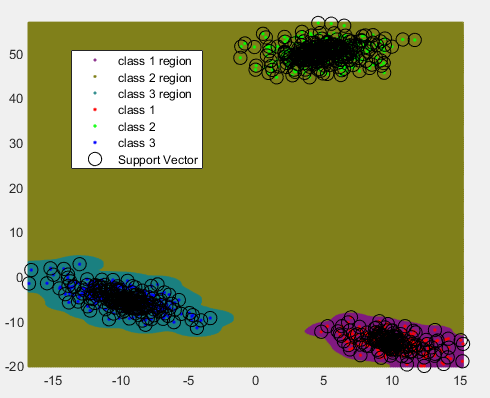
ylabel('Значение ошибки 2-го рода','FontName','Courier');

legend('по Чернову', 'Теоретическое значение', 'Экспериментальная (svm)', 'Экспериментальная (3 лаб.)');

hold off;

**Результаты:**

*Рис.6. Распознавание классов с коэфф-ми расстояний = 1, 2, 3, 4*

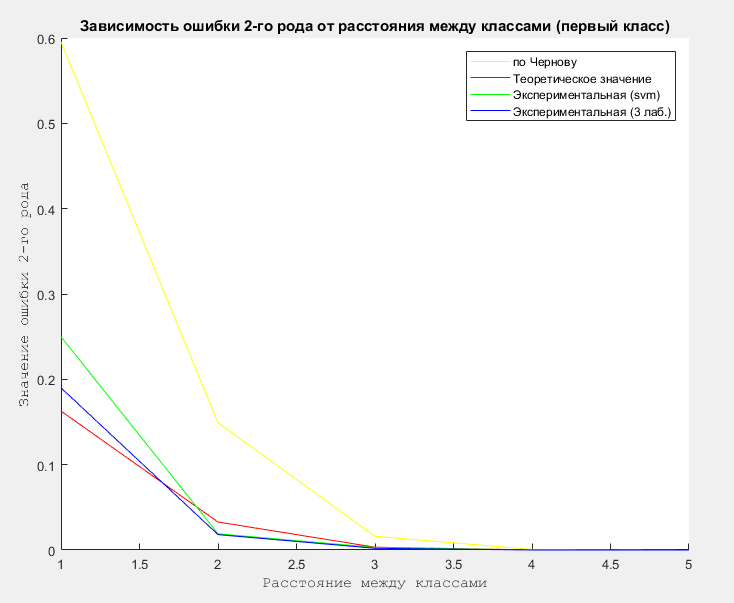
**

*Рис.7. Распознавание классов с коэфф-ом расстояния = 5*

**Матрицы ошибок:**

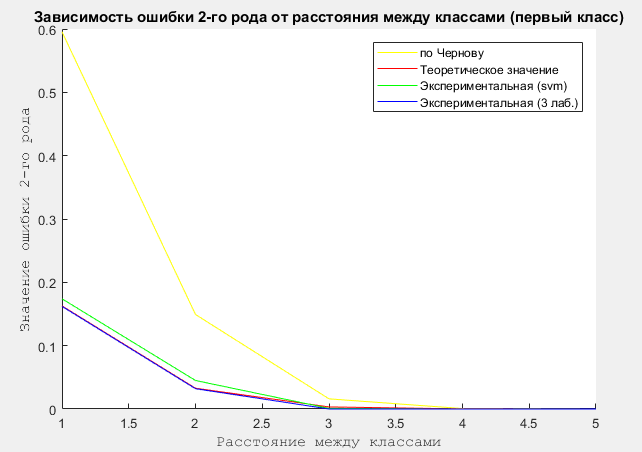
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dm** | **Теоретическая** | **По Чернову** | **Эксперимент. (3)** | **Эксперимент. (svm)** |
| **1** | **0.8033 0.0023 0.1944**  **0.0068 0.9801 0.0131**  **0.1589 0.0039 0.8372** | **0.2774 0.0108 0.7118**  **0.0072 0.9711 0.0217**  **0.5694 0.0260 0.4046** | **0.8130 0 0.1870**  **0 0.9970 0.0030**  **0.1510 0 0.8490** | **0.7480 0.0050 0.2470**  **0 0.9970 0.0030**  **0.1250 0.0020 0.8730** |
| **2** | **0.9724 0.0000 0.0276**  **0.0000 1.0000 0.0000**  **0.0328 0.0000 0.9672** | **0.8135 0.0000 0.1865**  **0.0000 1.0000 0.0000**  **0.1492 0.0000 0.8508** | **0.9820 0 0.0180**  **0 1.0000 0**  **0.0380 0 0.9620** | **0.9750 0.0060 0.0190**  **0 1.0000 0**  **0.0370 0.0070 0.9560** |
| **3** | **0.9983 0.0000 0.0017**  **0.0000 1.0000 0.0000**  **0.0033 0.0000 0.9967** | **0.9800 0.0000 0.0200**  **0.0000 1.0000 0.0000**  **0.0160 0.0000 0.9840** | **0.9980 0 0.0020**  **0 1.0000 0**  **0.0010 0 0.9990** | **0.9860 0.0110 0.0030**  **0 1.0000 0**  **0.0050 0.0020 0.9930** |
| **4** | **1.0000 0.0000 0.0000**  **0 1.0000 0.0000**  **0.0002 0 0.9998** | **0.9991 0.0000 0.0009**  **0.0000 1.0000 0.0000**  **0.0007 0.0000 0.9993** | **1 0 0**  **0 1 0**  **0 0 1** | **0.9860 0.0140 0**  **0 1.0000 0**  **0 0.0020 0.9980** |
| **5** | **1.0000 0.0000 0.0000**  **0 1.0000 0.0000**  **0.0000 0 1.0000** | **1.0000 0.0000 0.0000**  **0.0000 1.0000 0.0000**  **0.0000 0.0000 1.0000** | **1 0 0**  **0 1 0**  **0 0 1** | **0.9900 0.0100 0**  **0 1.0000 0**  **0 0.0020 0.9980** |

**График зависимости ошибки 2-го рода от расстояния между классами (для 3-его класса)**

**Генерация №1**

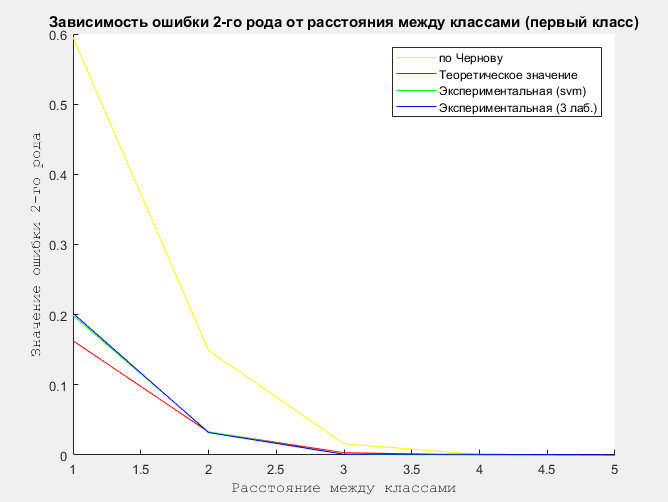
*Рис.8*

**Генерация №2**

****

*Рис.9*

**Генерация №3**

****

*Рис.10*

По данным результатам очевидно, что выбранный метод реализации svm работает хуже, чем первый. Метод опорных векторов в точности распознавания уступает другим методам.

**Выводы**

В результате работы были исследованы алгоритмы распознавания образов на основе аппарата машины опорных векторов (Support Vector Machine).