Горлов Максим Александрович, группа 17-1

Лабораторная работа № 7

Распознавание образов с использованием машины опорных векторов

**Цель работы**

Исследовать алгоритмы распознавания образов на основе аппарата машины опорных векторов (SupportVectorMachine).

**Задание**

Воспользовавшись классификатором SVM, вычислите вероятности ошибки при классификации линейно разделимых выборок трёх классов

**Код программы**

clear; clc; close all;

rng(1); % Для воспроизводимости

% Параметры

n = 2; % Размерность

M = 3; % Число классов

N\_train = 300; % Обучающих примеров

N\_test = 900; % Тестовых примеров

% Центры классов

m = [-2 2 0;

2 2 -3];

% Ковариация

C = eye(2);

% --- Обучающая выборка ---

X\_train = [];

Y\_train = [];

for i = 1:M

Xi = mvnrnd(m(:,i)', C, N\_train/M);

X\_train = [X\_train; Xi];

Y\_train = [Y\_train; i \* ones(N\_train/M, 1)];

end

% --- Исходные данные (до обучения) ---

figure;

gscatter(X\_train(:,1), X\_train(:,2), Y\_train, 'rgb', 'o', 6);

title('Исходные данные (обучающая выборка)');

xlabel('x\_1'); ylabel('x\_2');

legend('Класс 1','Класс 2','Класс 3');

% --- Обучение SVM (One-vs-One) ---

svmModels = cell(M);

for i = 1:M-1

for j = i+1:M

idx = Y\_train == i | Y\_train == j;

X\_pair = X\_train(idx,:);

Y\_pair = Y\_train(idx);

Y\_binary = double(Y\_pair == i); % 1 для класса i, -1 для j

Y\_binary(Y\_binary==0) = -1;

svmModels{i,j} = fitcsvm(X\_pair, Y\_binary, ...

'KernelFunction','linear', ...

'BoxConstraint', 1, ...

'Standardize', true);

end

end

% --- Тестовая выборка ---

X\_test = [];

Y\_test = [];

for i = 1:M

Xi = mvnrnd(m(:,i)', C, N\_test/M);

X\_test = [X\_test; Xi];

Y\_test = [Y\_test; i \* ones(N\_test/M, 1)];

end

% --- Классификация тестовой выборки ---

Y\_pred = zeros(N\_test,1);

for k = 1:N\_test

x = X\_test(k,:);

votes = zeros(M,1);

for i = 1:M-1

for j = i+1:M

label = predict(svmModels{i,j}, x);

if label == 1

votes(i) = votes(i) + 1;

else

votes(j) = votes(j) + 1;

end

end

end

[~, winner] = max(votes);

Y\_pred(k) = winner;

end

% --- Матрица ошибок ---

confusion\_mat = confusionmat(Y\_test, Y\_pred);

error\_prob\_matrix = confusion\_mat ./ sum(confusion\_mat, 2);

disp('Матрица ошибок (вероятности):');

disp(error\_prob\_matrix);

total\_accuracy = sum(diag(confusion\_mat)) / N\_test;

total\_error = 1 - total\_accuracy;

fprintf('Общая ошибка: %.4f\n', total\_error);

% --- График границ классификации ---

% Сетка

xrange = linspace(min(X\_train(:,1))-1, max(X\_train(:,1))+1, 300);

yrange = linspace(min(X\_train(:,2))-1, max(X\_train(:,2))+1, 300);

[xx, yy] = meshgrid(xrange, yrange);

grid\_points = [xx(:), yy(:)];

grid\_pred = zeros(size(grid\_points,1),1);

for k = 1:size(grid\_points,1)

x = grid\_points(k,:);

votes = zeros(M,1);

for i = 1:M-1

for j = i+1:M

label = predict(svmModels{i,j}, x);

if label == 1

votes(i) = votes(i) + 1;

else

votes(j) = votes(j) + 1;

end

end

end

[~, winner] = max(votes);

grid\_pred(k) = winner;

end

% Отображение результата с границами

figure;

gscatter(grid\_points(:,1), grid\_points(:,2), grid\_pred, [0.9 0.9 1; 0.9 1 0.9; 1 0.9 0.9], '.');

hold on;

gscatter(X\_train(:,1), X\_train(:,2), Y\_train, 'rgb', 'o', 6);

title('SVM: Классификация и границы решений');

xlabel('x\_1'); ylabel('x\_2');

legend('Реш. обл. кл.1', 'Реш. обл. кл.2', 'Реш. обл. кл.3', 'Класс 1','Класс 2','Класс 3');

axis tight;

**Результаты выполнения задания.**

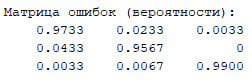
****

Рисунок 1.

Классификатор SVM демонстрирует высокую точность (более 95%) для всех классов, что подтверждает его эффективность для линейно разделимых данных.

Наибольшая ошибка наблюдается для Класса 2 (4.33% ошибочно отнесены к Классу 1), что указывает на частичное перекрытие признаков между этими классами.

Класс 3 классифицируется почти идеально (99% точности), что говорит о его хорошей линейной отделимости от остальных.

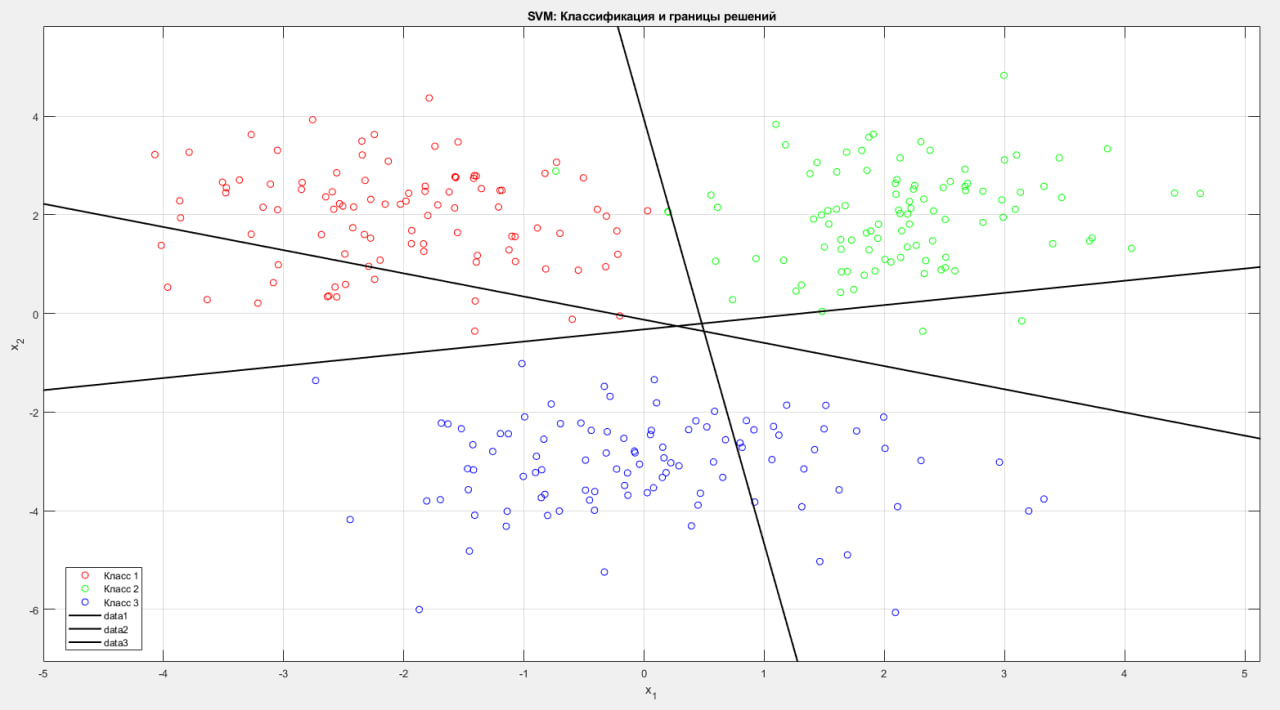
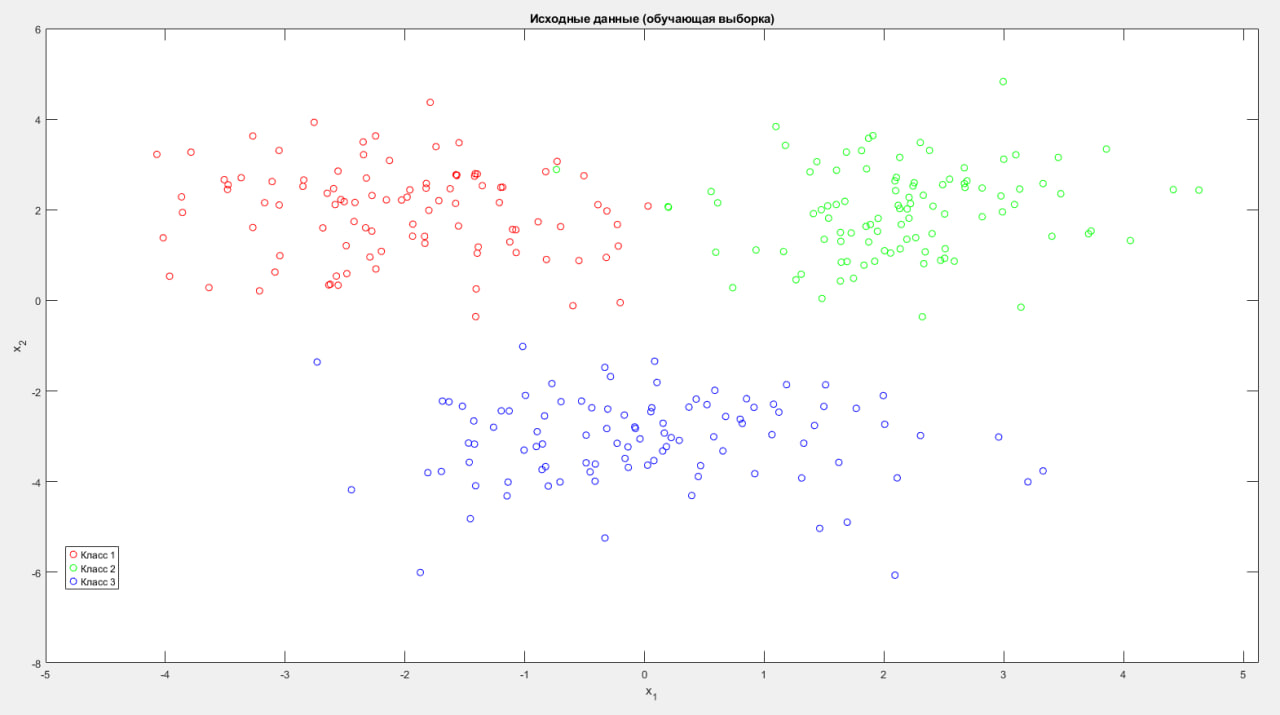


Рисунок 2.

Рисунок 3.

# Данные состоят из трёх классов, выборка сбалансирована. Границы между классами являются линейными (поскольку SVM использован для линейно разделимых данных). Точки данных, близкие к границам, могут быть причиной ошибок классификации, что согласуется с матрицей ошибок.

# Вывод

1. Алгоритм SVM успешно справился с задачей классификации для линейно разделимых данных, о чём свидетельствуют высокие значения точности в матрице ошибок. Работа подтвердила, что SVM является надёжным инструментом классификации линейно разделимых данных.
2. Небольшие ошибки (2–4%) могут быть связаны с частичным перекрытием классов в пространстве признаков, наличием выбросов или шумов в данных, а также ограниченной гибкостью линейных границ (если данные не идеально линейно разделимы).