

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
```

```

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
    grid_pred(k) = max(votes);
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.

Матрица ошибок (вероятности):

0.9733	0.0233	0.0033
0.0433	0.9567	0
0.0033	0.0067	0.9900

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.

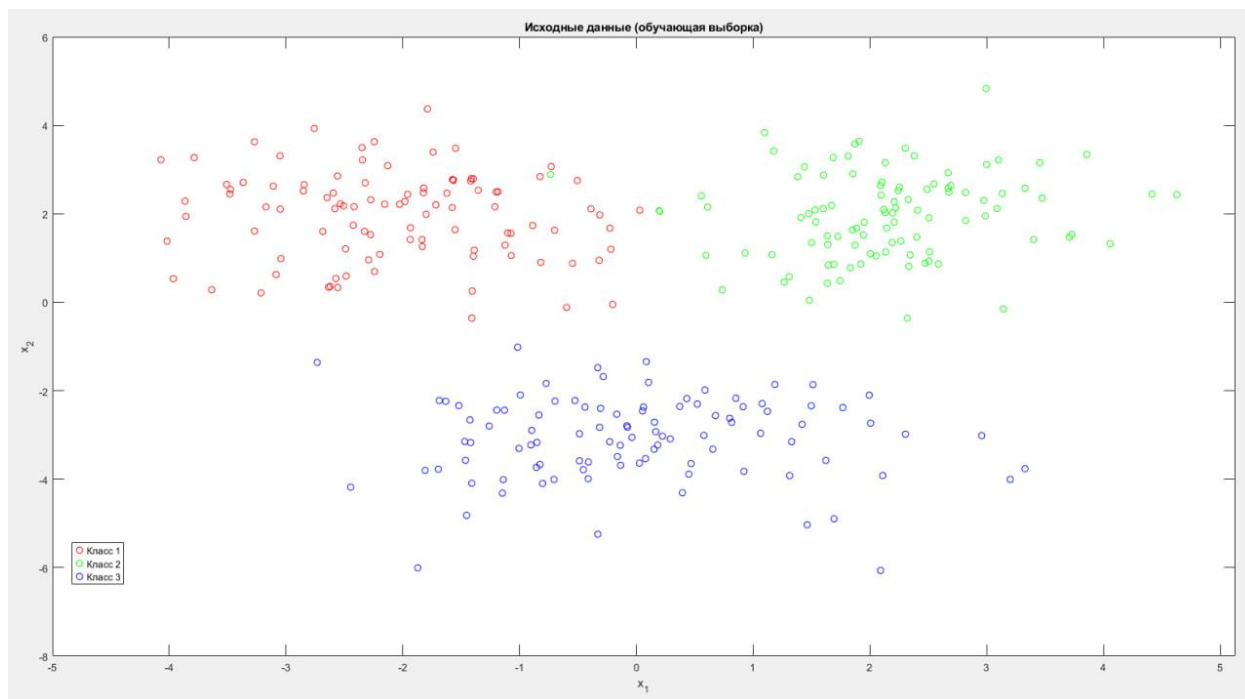


Рисунок 2.

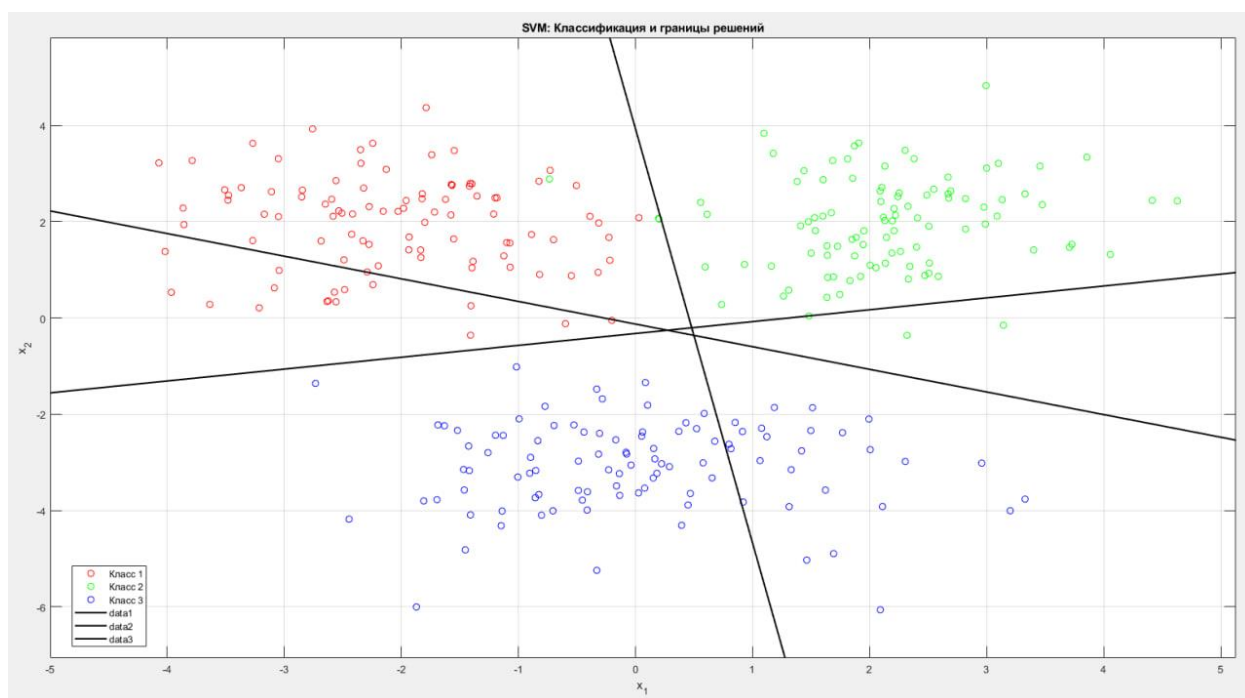


Рисунок 3.

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).