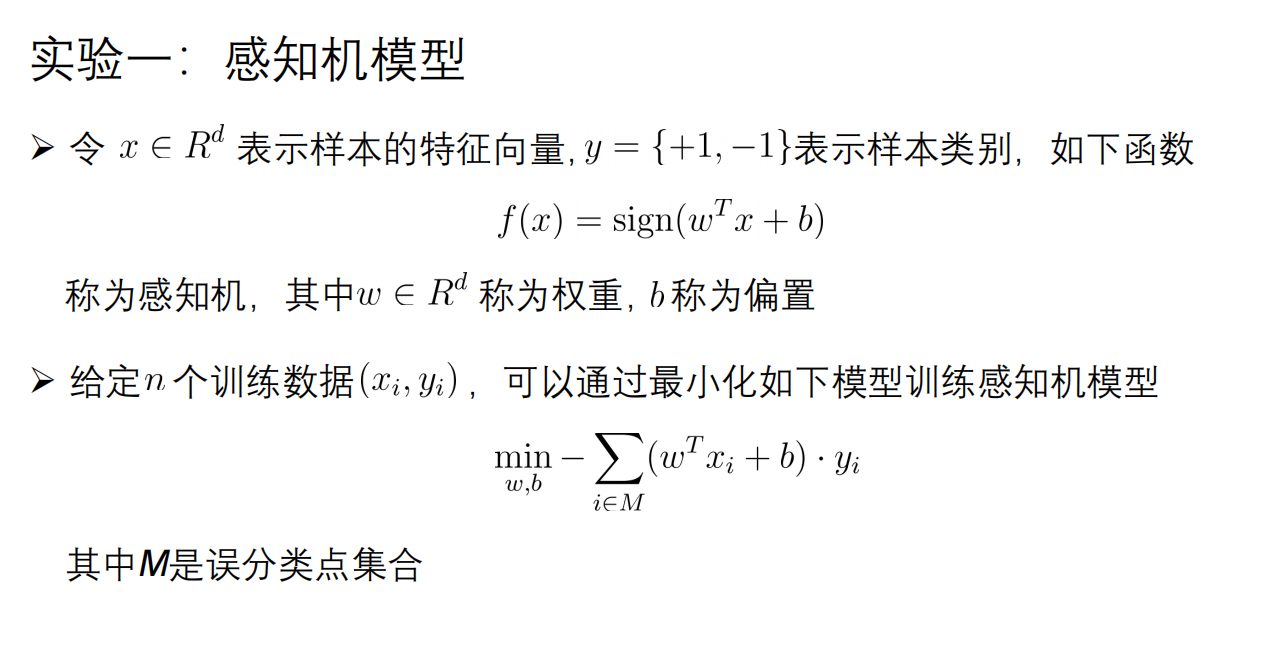
# 实验一 感知机模型

2313863 李佳明



## 一、实验过程解释

1. 数据生成

·首先设置两类数据的中间点

类别1：（1，1）

类别2：（3，4）

·生成训练数据：

每类100个样本，使用高斯分布添加噪声：中心点+randn（）

·生成测试数据

每类十个样本

同样基于高斯分布

1. 感知机训练

·初始化参数

w = [0; 0]; % 权重向量

b = 0; % 偏置项

eta = 0.1; % 学习率

·核心循环训练

for 每次迭代:

misclassified = 0; % 误分类计数器

for 每个样本 i:

if y\_i\*(w'\*x\_i + b) <= 0: % 检查是否误分类

w = w + eta \* y\_i \* x\_i % 更新权重

b = b + eta \* y\_i % 更新偏置

misclassified += 1

end

if misclassified == 0:

break % 提前终止

End

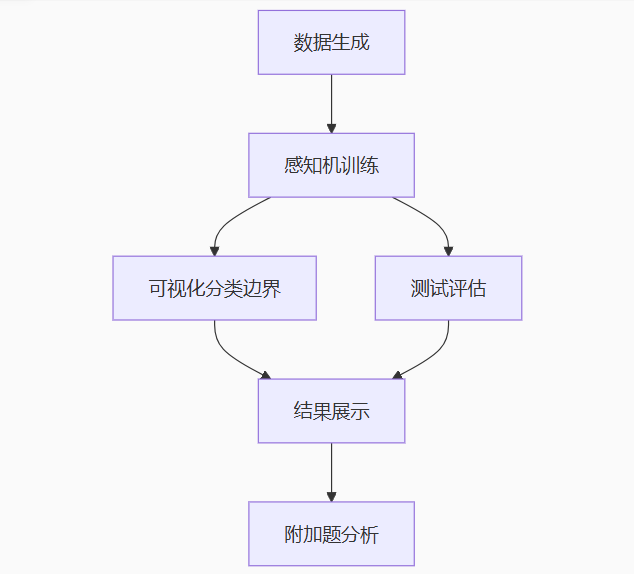
·保留训练历史

1. 可视化分类边界

x1 = 取值范围

y1 = (-b - w(1)\*x1) / w(2) % 解方程 w·x + b = 0

1. 测试评估



## 实现代码

%% 感知机实验完整实现 - 南开大学机器学习实验课

%% 1. 数据生成（基于提供的代码）

clear all; close all; clc;

% 训练数据生成（每类100个）

n\_train = 100; % 每类样本量

center1 = [1, 1]; % 第一类数据中心

center2 = [3, 4]; % 第二类数据中心

% 生成训练数据矩阵（200x2）

X\_train = zeros(2\*n\_train, 2);

Y\_train = zeros(2\*n\_train, 1);

% 第一类数据（标签=1）

X\_train(1:n\_train, :) = ones(n\_train, 1)\*center1 + randn(n\_train, 2);

Y\_train(1:n\_train) = 1;

% 第二类数据（标签=-1）

X\_train(n\_train+1:2\*n\_train, :) = ones(n\_train, 1)\*center2 + randn(n\_train, 2);

Y\_train(n\_train+1:2\*n\_train) = -1;

% 测试数据生成（每类10个）

n\_test = 10;

X\_test = zeros(2\*n\_test, 2);

Y\_test = zeros(2\*n\_test, 1);

% 第一类测试数据

X\_test(1:n\_test, :) = ones(n\_test, 1)\*center1 + randn(n\_test, 2);

Y\_test(1:n\_test) = 1;

% 第二类测试数据

X\_test(n\_test+1:2\*n\_test, :) = ones(n\_test, 1)\*center2 + randn(n\_test, 2);

Y\_test(n\_test+1:2\*n\_test) = -1;

% 可视化训练数据

figure(1)

set(gcf, 'Position', [100, 100, 700, 600], 'color', 'w')

set(gca, 'Fontsize', 12)

plot(X\_train(1:n\_train, 1), X\_train(1:n\_train, 2), 'ro', 'LineWidth', 1, 'MarkerSize', 6)

hold on;

plot(X\_train(n\_train+1:2\*n\_train, 1), X\_train(n\_train+1:2\*n\_train, 2), 'b\*', 'LineWidth', 1, 'MarkerSize', 6)

xlabel('x axis');

ylabel('y axis');

title('训练数据分布');

legend('class 1 (train)', 'class 2 (train)');

grid on;

%% 2. 感知机实现（手动实现随机梯度下降）

% 初始化参数

w = zeros(2, 1); % 权重向量 (2x1)

b = 0; % 偏置项

eta = 0.1; % 学习率

max\_epochs = 100; % 最大迭代次数

converged = false;% 收敛标志

% 训练过程记录（用于可视化）

training\_history = struct('w', cell(max\_epochs,1), 'b', cell(max\_epochs,1));

% 随机梯度下降

for epoch = 1:max\_epochs

misclassified = 0;

% 保存当前参数（用于历史记录）

training\_history(epoch).w = w;

training\_history(epoch).b = b;

% 遍历所有样本

for i = 1:length(Y\_train)

% 计算预测值

prediction = sign(w' \* X\_train(i, :)' + b);

% 检查是否误分类

if Y\_train(i) \* (w' \* X\_train(i, :)' + b) <= 0

% 更新权重和偏置

w = w + eta \* Y\_train(i) \* X\_train(i, :)';

b = b + eta \* Y\_train(i);

misclassified = misclassified + 1;

end

end

% 如果没有误分类点，提前终止

if misclassified == 0

converged = true;

fprintf('在第 %d 轮迭代后收敛\n', epoch);

break;

end

% 每10轮显示进度

if mod(epoch, 10) == 0

fprintf('第 %d 轮迭代，误分类点数: %d\n', epoch, misclassified);

end

end

% 最终训练结果

fprintf('训练完成! 最终参数: w = [%.4f, %.4f], b = %.4f\n', w(1), w(2), b);

%% 3. 可视化分类边界

% 计算分类边界

x\_range = [min(X\_train(:,1))-1, max(X\_train(:,1))+1];

x1 = linspace(x\_range(1), x\_range(2), 1000);

y1 = (-b - w(1)\*x1) / w(2); % w1\*x + w2\*y + b = 0

% 绘制分类结果

figure(2)

set(gcf, 'Position', [100, 100, 900, 700], 'color', 'w')

set(gca, 'Fontsize', 12)

% 绘制训练数据

plot(X\_train(1:n\_train, 1), X\_train(1:n\_train, 2), 'ro', 'LineWidth', 1, 'MarkerSize', 6)

hold on;

plot(X\_train(n\_train+1:2\*n\_train, 1), X\_train(n\_train+1:2\*n\_train, 2), 'b\*', 'LineWidth', 1, 'MarkerSize', 6)

% 绘制分类边界

plot(x1, y1, 'k-', 'LineWidth', 2)

xlabel('x axis');

ylabel('y axis');

title('感知机分类结果');

legend('class 1 (train)', 'class 2 (train)', '决策边界');

grid on;

axis equal;

%% 4. 测试集评估

% 预测测试集

correct = 0;

predictions = zeros(size(Y\_test));

for i = 1:length(Y\_test)

% 计算预测值

pred\_val = sign(w' \* X\_test(i, :)' + b);

predictions(i) = pred\_val;

% 检查是否正确分类

if pred\_val == Y\_test(i)

correct = correct + 1;

end

end

% 计算准确率

accuracy = correct / length(Y\_test) \* 100;

fprintf('测试准确率: %.2f%% (%d/%d)\n', accuracy, correct, length(Y\_test));

% 可视化测试结果

figure(3)

set(gcf, 'Position', [100, 100, 900, 700], 'color', 'w')

set(gca, 'Fontsize', 12)

% 绘制训练数据（浅色）

plot(X\_train(1:n\_train, 1), X\_train(1:n\_train, 2), 'ro', 'MarkerFaceColor', [1 0.8 0.8], 'MarkerSize', 4)

hold on;

plot(X\_train(n\_train+1:2\*n\_train, 1), X\_train(n\_train+1:2\*n\_train, 2), 'b\*', 'MarkerEdgeColor', [0.7 0.7 1], 'MarkerSize', 6)

% 绘制测试数据（深色）

plot(X\_test(1:n\_test, 1), X\_test(1:n\_test, 2), 'ro', 'MarkerFaceColor', 'r', 'MarkerSize', 8)

plot(X\_test(n\_test+1:2\*n\_test, 1), X\_test(n\_test+1:2\*n\_test, 2), 'b\*', 'MarkerSize', 10)

% 绘制分类边界

plot(x1, y1, 'k-', 'LineWidth', 2)

% 标记错误分类点

misclassified\_idx = find(predictions ~= Y\_test);

if ~isempty(misclassified\_idx)

plot(X\_test(misclassified\_idx, 1), X\_test(misclassified\_idx, 2), 'ks', 'MarkerSize', 12, 'LineWidth', 2)

legend\_str = {'class 1 (train)', 'class 2 (train)', 'class 1 (test)', 'class 2 (test)', '决策边界', '错误分类'};

else

legend\_str = {'class 1 (train)', 'class 2 (train)', 'class 1 (test)', 'class 2 (test)', '决策边界'};

end

xlabel('x axis');

ylabel('y axis');

title(sprintf('测试结果 (准确率: %.2f%%)', accuracy));

legend(legend\_str);

grid on;

axis equal;

%% 5. 附加题1：不同迭代次数的决策边界变化（示例）

if exist('training\_history', 'var')

figure(4)

set(gcf, 'Position', [100, 100, 1200, 900], 'color', 'w')

set(gca, 'Fontsize', 10)

% 选择关键迭代点

plot\_epochs = [1, 5, 10, 20, 50, min(epoch, max\_epochs)];

colors = lines(length(plot\_epochs));

% 绘制训练数据

plot(X\_train(1:n\_train, 1), X\_train(1:n\_train, 2), 'ro', 'MarkerSize', 4, 'MarkerFaceColor', [1 0.8 0.8])

hold on;

plot(X\_train(n\_train+1:2\*n\_train, 1), X\_train(n\_train+1:2\*n\_train, 2), 'b\*', 'MarkerEdgeColor', [0.7 0.7 1], 'MarkerSize', 6)

% 绘制不同迭代次数的决策边界

legend\_str = {'class 1', 'class 2'};

for i = 1:length(plot\_epochs)

e = plot\_epochs(i);

w\_hist = training\_history(e).w;

b\_hist = training\_history(e).b;

% 计算分类边界

y\_hist = (-b\_hist - w\_hist(1)\*x1) / w\_hist(2);

% 绘制边界

plot(x1, y\_hist, '-', 'LineWidth', 1.5, 'Color', colors(i, :))

legend\_str{end+1} = sprintf('迭代 %d', e);

end

xlabel('x axis');

ylabel('y axis');

title('不同迭代次数的决策边界变化');

legend(legend\_str, 'Location', 'eastoutside');

grid on;

axis equal;

end

%% 6. 实验报告生成（结果输出）

fprintf('\n===== 实验报告摘要 =====\n');

fprintf('实验名称: 感知机模型实现\n');

fprintf('训练数据: 每类100个样本 (共200个)\n');

fprintf('测试数据: 每类10个样本 (共20个)\n');

fprintf('学习率(η): %.2f\n', eta);

fprintf('最大迭代次数: %d\n', max\_epochs);

fprintf('实际迭代次数: %d\n', epoch);

fprintf('收敛状态: %s\n', converged);

fprintf('最终权重 w: [%.4f, %.4f]\n', w(1), w(2));

fprintf('最终偏置 b: %.4f\n', b);

fprintf('测试准确率: %.2f%%\n', accuracy);

fprintf('========================\n');

## 实验结果分析

