山东大学 计算机科学与技术 学院

机器学习与模式识别 课程实验报告

学号：202300130183	姓名： 宋浩宇	班级：23 级人工智能班
实验题目：Linear Discriminant Analysis		
实验学时：2	实验日期：2025/3/11	

实验环境：

软件环境：

系统：Windows 11 家庭中文版 23H2 22631.4317

计算软件：MATLAB 版本: 9.8.0.1323502 (R2020a)

Java 版本: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode

硬件环境：

CPU：13th Gen Intel(R) Core(TM) i9-13980HX 2.20 GHz

内存：32.0 GB (31.6 GB 可用)

磁盘驱动器：NVMe WD_BLACKSN850X2000GB

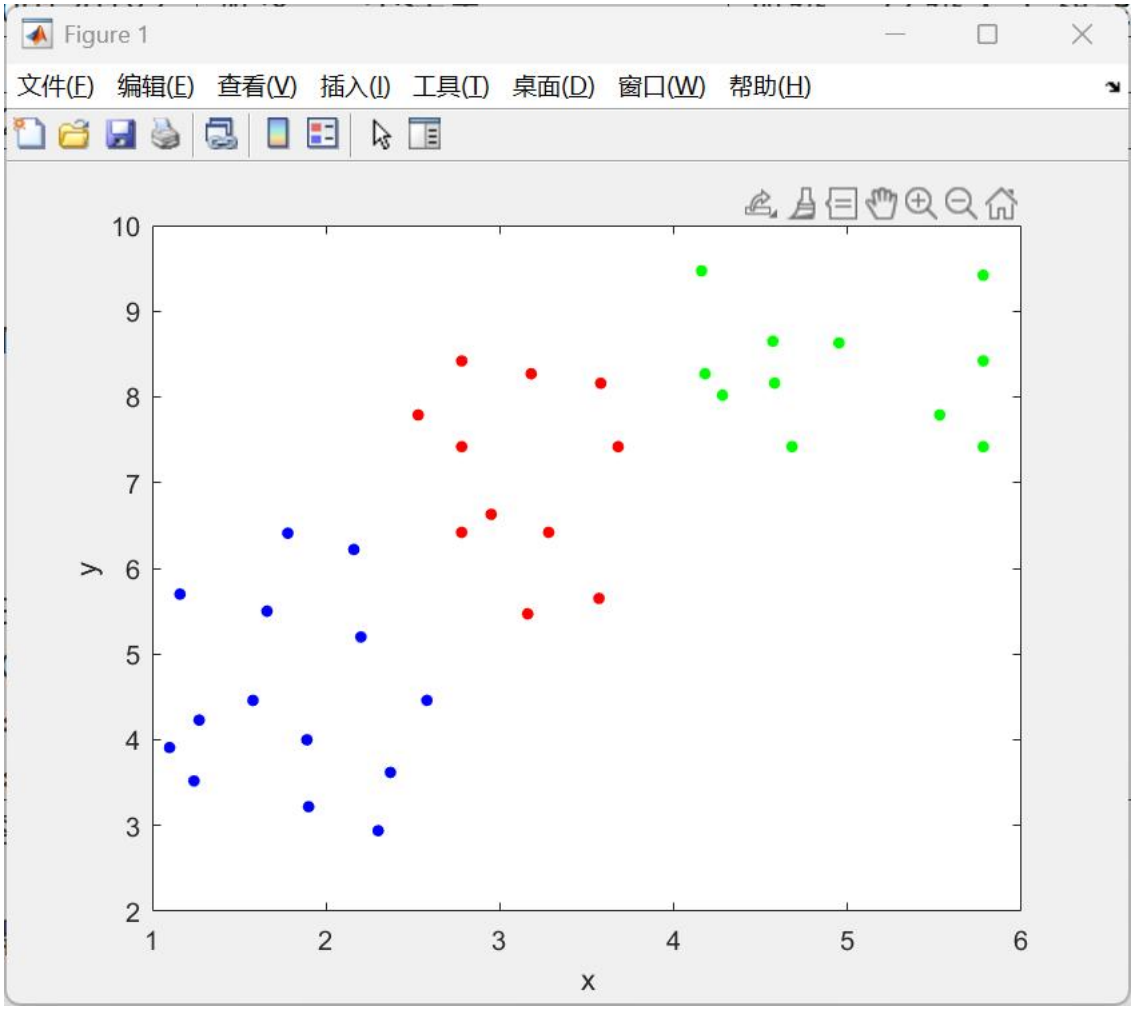
显示适配器：NVIDIA GeForce RTX 4080 Laptop GPU

1. 实验内容

In this exercise, you need to implement Linear Discriminant Analysis(LDA)

2. 实验步骤

(1) 获取实验使用的数据。



（2） 构造二分类 LDA 模型：

用$\sum_0$和$\sum_1$分别表示$\sum_{x\in X_0}(x-\mu_0)(x-\mu_0)^T$和$\sum_{x\in X_1}(x-\mu_1)(x-\mu_1)^T$，并令$S_w=\sum_0+\sum_1$

$$w=S_w^{-1}(\mu_0-\mu_1)$$

（3） 用 matlab 代码实现并进行计算。

（4） 完成结果的可视化。

（5） 构造多分类的 LDA 模型：

$$\begin{aligned}S_w&=\sum_{i=1}^{N}S_{w_i}\\&=\sum_{i=1}^{N}\sum_{x\in X_i}(x-\mu_i)(x-\mu_i)^T\end{aligned}$$

$$
\begin{aligned}
S_b &= S_t - S_w \\
&= \sum_{i=1}^{N}\sum_{j=1}^{m_i}(x_j-\mu)(x_j-\mu)^T - \sum_{i=1}^{N}\sum_{x\in X_i}(x-\mu_i)(x-\mu_i)^T \\
&= \sum_{i=1}^{N}\sum_{j=1}^{m_i}(x_j-\mu)(x_j-\mu)^T - \sum_{i=1}^{N}\sum_{j=1}^{m_i}(x_j-\mu_i)(x_j-\mu_i)^T \\
&= \sum_{i=1}^{N}\sum_{j=1}^{m_i}[(x_j-\mu)(x_j-\mu)^T - (x_j-\mu_i)(x_j-\mu_i)^T] \\
&= \sum_{i=1}^{N}\sum_{j=1}^{m_i}[(x_j x_j^T - x_j\mu^T - \mu x_j^T + \mu\mu^T) - (x_j x_j^T - x_j\mu_i^T - \mu_i x_j^T + \mu_i\mu_i^T)] \\
&= \sum_{i=1}^{N}[\sum_{j=1}^{m_i}x_j(-\mu^T+\mu_i^T) + (-\mu+\mu_i)\sum_{j=1}^{m_i}x_j^T + \sum_{j=1}^{m_i}(\mu\mu^T - \mu_i\mu_i^T)] \\
&= \sum_{i=1}^{N}[m_i\mu_i(-\mu^T+\mu_i^T) + (-\mu+\mu_i)m_i\mu_i^T + m_i(\mu\mu^T - \mu_i\mu_i^T)] \\
&= \sum_{i=1}^{N}m_i[-\mu_i\mu^T + \mu_i\mu_i^T - \mu\mu_i^T + \mu\mu^T] \\
&= \sum_{i=1}^{N}m_i(\mu_i-\mu)(\mu_i-\mu)^T
\end{aligned}
$$

而W的闭式解就是**$S_w^{-1}S_b$的d'个最大非零广义特征值所对应的特征向量组成的矩阵**，$d' \le N-1$。

（6） 用 matlab 代码实现并进行计算。

（7） 完成结果的可视化。

3. 测试结果

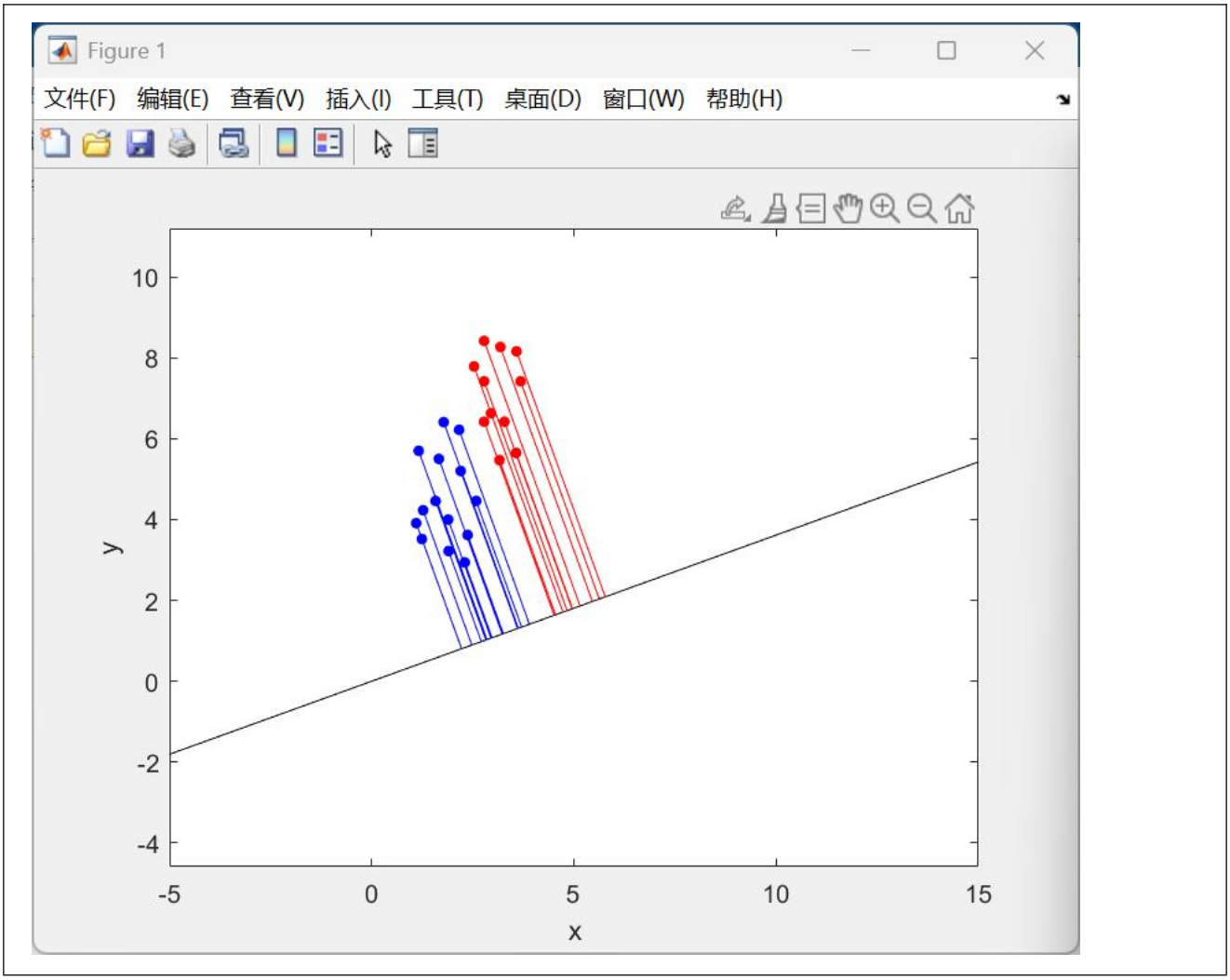
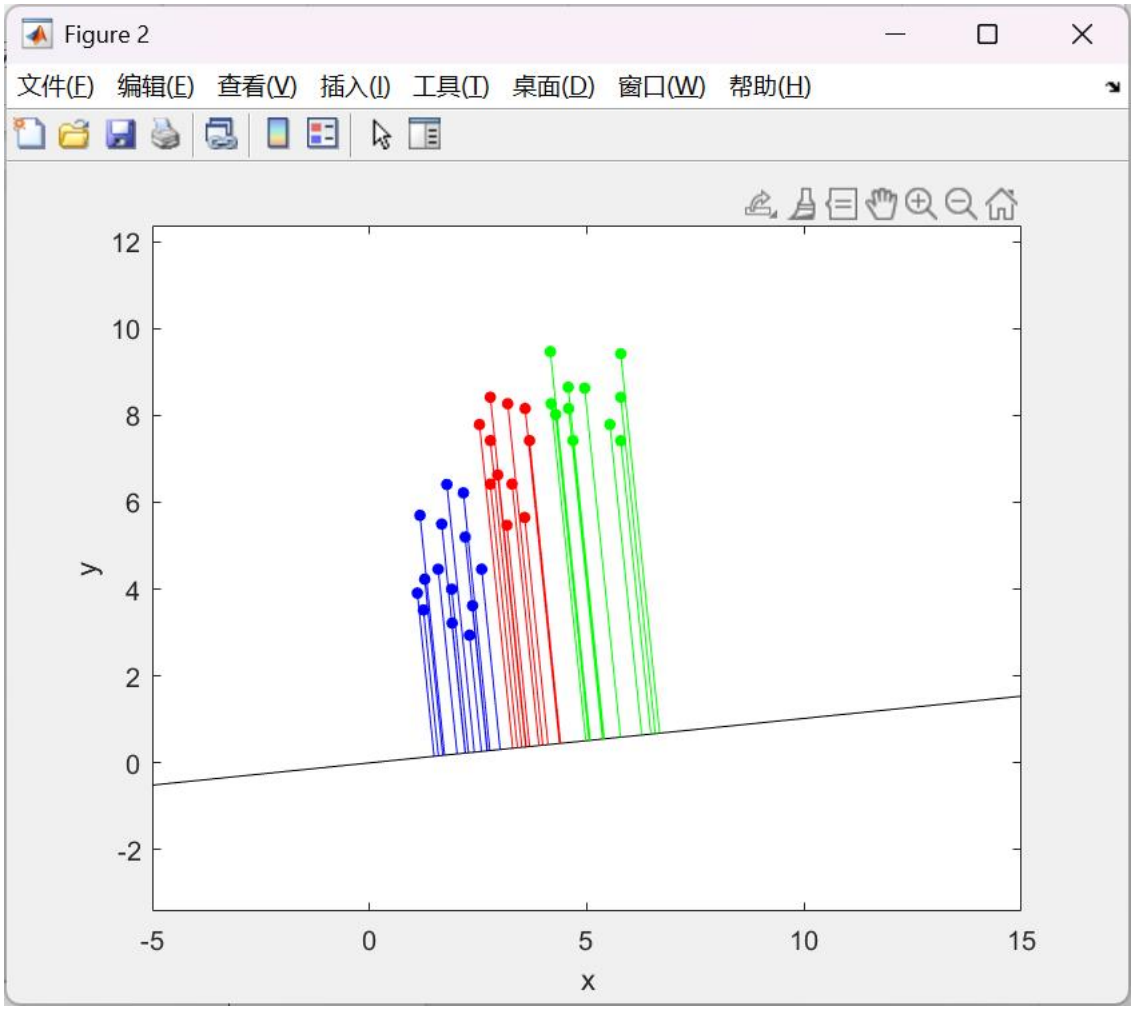


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
10
8
6
4
2
0
-2
-4
y
-5
0
5
10
15
x



4. 附录：实现源代码

```
%% 清空环境变量
clear;
clc;

%% 加载数据
redPoints = load('ex3Data/ex3red.dat');    % 红色点
bluePoints = load('ex3Data/ex3blue.dat');  % 蓝色点
greenPoints = load('ex3Data/ex3green.dat');% 绿色点
%% 绘制数据点
figure;
plot(redPoints(:, 1), redPoints(:, 2), 'r.', 'MarkerSize', 15);
axis equal;
hold on;
plot(bluePoints(:, 1), bluePoints(:, 2), 'b.', 'MarkerSize', 15);
% plot(greenPoints(:, 1), greenPoints(:, 2), 'g.', 'MarkerSize', 15);
xlabel('x');
ylabel('y');
%% 二分类 LDA
% disp(redPoints);
```

```
mu0 = mean(redPoints);
mu1 = mean(bluePoints);
sum0 = cov(redPoints - mu0);
sum1 = cov(bluePoints - mu1);
sum_w = sum0+sum1;
w = inv(sum_w)*(mu0-mu1)';
% disp(w);
x = linspace(-5,15);
y = x * (w(2)/w(1));
plot(x,y,'k-');
for i = 1:size(redPoints, 1)
    m = w(2) / w(1);
    b = 0;
    x0 = redPoints(i, 1);
    y0 = redPoints(i, 2);
    x1 = (m * (y0 - b) + x0) / (m^2 + 1);
    y2 = (m^2 * y0 + m * x0 + b) / (m^2 + 1);
    plot([x0 x1], [y0 y2], 'r-');
end
for i = 1:size(bluePoints, 1)
    m = w(2) / w(1);
    b = 0;
    x0 = bluePoints(i, 1);
    y0 = bluePoints(i, 2);
    x1 = (m * (y0 - b) + x0) / (m^2 + 1);
    y2 = (m^2 * y0 + m * x0 + b) / (m^2 + 1);
    plot([x0 x1], [y0 y2], 'b-');
end
%% 多分类 LDA
figure;
plot(redPoints(:, 1), redPoints(:, 2), 'r.', 'MarkerSize', 15);
axis equal;
hold on;
plot(bluePoints(:, 1), bluePoints(:, 2), 'b.', 'MarkerSize', 15);
plot(greenPoints(:, 1), greenPoints(:, 2), 'g.', 'MarkerSize', 15);
xlabel('x');
ylabel('y');
mu = mean([bluePoints;redPoints;greenPoints]);
disp(mu);
sb = size(bluePoints, 1) * cov(mean(bluePoints) - mu) + size(greenPoints,
1) * cov(mean(greenPoints) - mu) + size(redPoints, 1) * cov(mean(redPoints)
- mu);
sw = cov(bluePoints - mean(bluePoints)) + cov(greenPoints -
mean(greenPoints)) + cov(redPoints - mean(redPoints));
S = inv(sw) * sb;
```

```
[eigvecs, eigvals] = eig(S);
disp(eigvecs);
disp(eigvals);
max_eigval = max(diag(eigvals));
[~, idx] = max(diag(eigvals));
max_eigvec = eigvecs(:, idx);
disp(max_eigval);
disp(max_eigvec);
unit_max_eigvec = max_eigvec / norm(max_eigvec);
x = linspace(-5,15);
y = x * (unit_max_eigvec(2)/unit_max_eigvec(1));
plot(x,y,'k-');
for i = 1:size(redPoints, 1)
    m = unit_max_eigvec(2) / unit_max_eigvec(1);
    b = 0;
    x0 = redPoints(i, 1);
    y0 = redPoints(i, 2);
    x1 = (m * (y0 - b) + x0) / (m^2 + 1);
    y2 = (m^2 * y0 + m * x0 + b) / (m^2 + 1);
    plot([x0 x1], [y0 y2], 'r-');
end
for i = 1:size(bluePoints, 1)
    m = unit_max_eigvec(2) / unit_max_eigvec(1);
    b = 0;
    x0 = bluePoints(i, 1);
    y0 = bluePoints(i, 2);
    x1 = (m * (y0 - b) + x0) / (m^2 + 1);
    y2 = (m^2 * y0 + m * x0 + b) / (m^2 + 1);
    plot([x0 x1], [y0 y2], 'b-');
end
for i = 1:size(greenPoints, 1)
    m = unit_max_eigvec(2) / unit_max_eigvec(1);
    b = 0;
    x0 = greenPoints(i, 1);
    y0 = greenPoints(i, 2);
    x1 = (m * (y0 - b) + x0) / (m^2 + 1);
    y2 = (m^2 * y0 + m * x0 + b) / (m^2 + 1);
    plot([x0 x1], [y0 y2], 'g-');
end
```