# 实验报告五

**学号：**20201060287

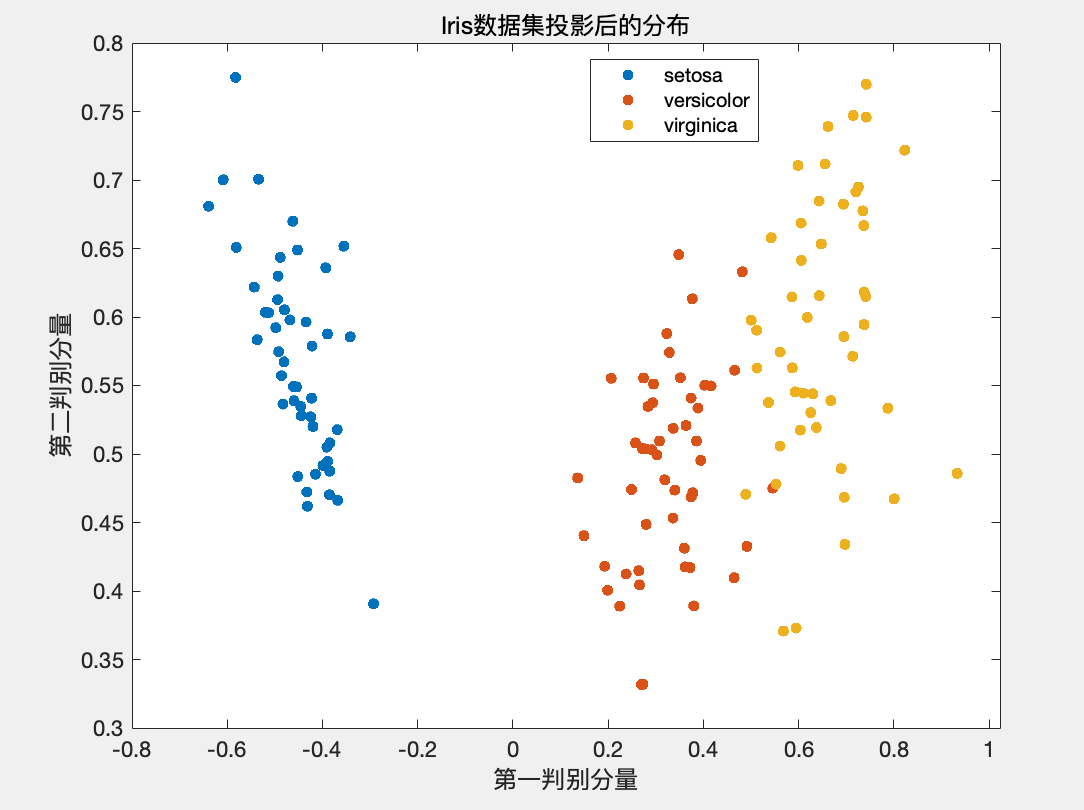
**姓名：**李昂

**实验名称：**线性判别分析

**实验内容：**使用MATLAB，用线性判别分析(LDA)算法对Iris数据集进行降维处理

**实验要求及结果：**

采用LDA算法，将Iris数据集从原始的四维空间投影到二维空间，请画出Iris数据集在投影后二维空间中的分布情况（要求用不同的颜色/线型来画不同类别的投影后数据）



**问题回答：**

**LDA算法中的类内离散度矩阵和类间离散度矩阵分别描述的是什么？它们分别需要最大化还是最小化？这样做有什么好处？**

类内离散度矩阵：描述同一类别内数据点之间的差异性，需要最小化

类间离散度矩阵：描述不同类别之间的差异性，需要最大化

好处：减少数据的冗余信息，提高分类的准确性和效率。可视化高维数据，帮助我们更好地理解数据的分布和结构

**附实验代码：**

|  |
| --- |
| % 读取原始数据  [attrib1, attrib2, attrib3, attrib4, class] = textread('iris.data', '%f%f%f%f%s', 'delimiter', ',');  % 将数据矩阵和类别标签合并为一个矩阵  x = [attrib1, attrib2, attrib3, attrib4]';  y = grp2idx(class);  % 进行LDA降维  [vec, val] = LDA(x, y);  W = vec(:, 1:2); % 取前两个判别分量  y\_proj = W' \* x; % 投影后的数据  % 画出投影后的数据分布  figure  gscatter(y\_proj(1, :), y\_proj(2, :), y)  title('Iris数据集投影后的分布')  xlabel('第一判别分量')  ylabel('第二判别分量')  legend('setosa', 'versicolor', 'virginica', 'Location', 'best')  function [vec, val] = LDA(xtr, ytr)  %%%%%%%%%%%%%%%%%%%%%%%%%%%%  % Input:  % xtr: data matrix (Each column is a data point)  % ytr: class label (class 1, ..., k)  % Output:  % vec: sorted discriminative components  % val: corresponding eigenvalues  %%%%%%%%%%%%%%%%%%%%%%%%%%%%  [D, ntr] = size(xtr);  classnum = length(unique(ytr));  miu = mean(xtr, 2);  sigmaB = sparse(D, D);  for i = 1:classnum  miu\_class(:, i) = mean(xtr(:, find(ytr == i)), 2);  sigmaB = sigmaB + length(find(ytr == i)) \* (miu\_class(:, i) - miu) \* (miu\_class(:, i) - miu)';  end  sigmaB = (sigmaB + sigmaB') / 2;  sigmaT = (ntr - 1) \* cov(xtr');  sigmaT = (sigmaT + sigmaT') / 2;  sigmaW = sigmaT - sigmaB;  sigmaW = (sigmaW + sigmaW') / 2;  [eigvector, eigvalue] = eig(sigmaB, sigmaW);  [val, id] = sort(-diag(eigvalue));  vec = eigvector(:, id);  val = -val;  end |