

# 深圳大学实验报告

课程名称 机器学习

项目名称 实验二：LDA 算法

学 院 计算机与软件学院

专 业 软件工程

指导教师 赖志辉

报 告 人 郑杨 学号 2020151002

实验时间 2022 年 3 月 19 日至 2022 年 3 月 24 日

实验报告提交时间 2022 年 3 月 24 日

教务处制

## 一、实验目的与要求

1. 简述 LDA 原理、算法模型与优化问题，给出全程推导细节；证明  $S_t = S_b + S_w$ ；
2. 给出 LDA 的各种等价模型表示(除法的、减法的及其调换位置的等)，在各数据集（不少于 3 个）比较 PCA 与“LDA 的各种等价模型与正则模型”的人脸识别精度。
3. 比较 eigenface 与 fisherface 的不同，并取 3 个类的图像投影在二维和三维空间中，并用不同颜色的点表示不同的类，每个类选 3 个有代表性的点对应的人脸图像显示在该点的边上（用 plot 命令或 imshow），比较 PCA 与 LDA 的结果的异不同。
4. 参看前人论文，设计一个全新的线性鉴别分析方法或子空间学习方法，把简要内容写在本实验报告中。

## 二、实验内容与方法

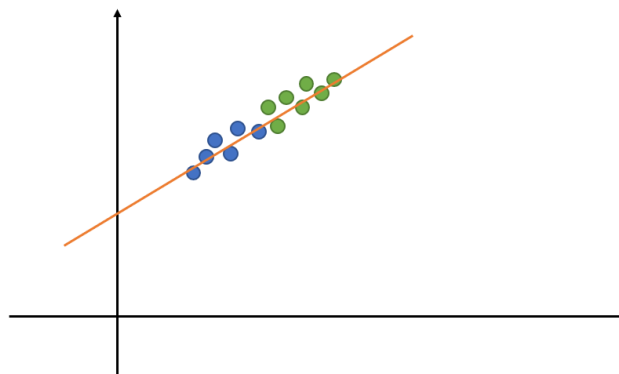
1. 阐述 LDA 原理，利用 `mathtype` 编辑公式推导细节，证明相关优化原理。
2. 使用 `matlab` 编写程序实现 LDA 降维，并在 3 个数据集上对比 PCA 与 LDA 的识别率。
3. 比较 eigenface 与 fisherface 的不同，并取 3 个类的图像投影在二维和三维空间中，并用不同颜色的点表示不同的类，每个类选 3 个有代表性的点对应的人脸图像显示在该点的边上（用 `plot` 命令或 `imshow`），比较 PCA 与 LDA 的结果的异不同。

## 三、实验步骤与过程

### 1. LDA 原理阐述

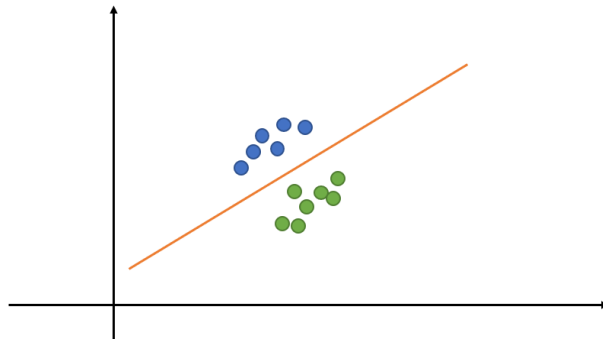
#### (1) 基本思想

上一次实验所涉及到的 PCA，是一种经典的无监督学习方法，它无法考虑数据样本的类别信息，思想是将数据投影到方差最大的几个相互正交的方向上希望保留尽可能多的样本信息。方差体现的就是样本信息的多少，方差越大，样本的多样性就越好。比如下图数据使用 PCA 投影的最佳方向为红线：

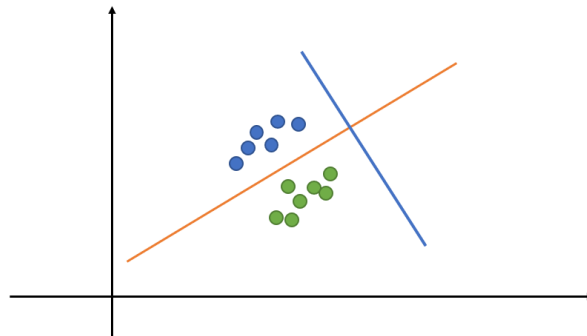


但是在其他不同分布的数据集上，PCA 所选择的方向就不一定合适，比如说下图，红线

依然是 PCA 所选择的方向，虽然说数据的方差是最大的，但是他把原来线性可分的两类数据混在了一起，将不再线性可分，甚至是不可分的，这样做就达不到预期的效果了。



于是我们引进另一种降维方式，LDA 线性判别分析。与 PCA 不同的是，LDA 是一种监督学习的方法，利用了数据集样本的类别信息。LDA 的目标是希望投影之后每一类数据类内的投影点尽可能接近，而不同类别间的投影点尽可能远离。其基本思想就是，在尽可能多地保留原始样本信息的前提下，使得投影后类内方差小，类间距离大。比如说上面那个例子用 LDA 找出的方向将会是下图中的蓝线：



## (2) 数学原理推导（二分类）

我们先从最简单的二分类问题出发，就是样本的类别数只有两类。我们先定义一些符号：

$x_i^j$ ：表示第*i*类中的第*j*个样本

$N_i$ ：表示第*i*类的样本数

$\mu_i$ ：表示第*i*类的样本均值

$S_i$ ：表示第*i*类的样本方差

$w$ ：投影基向量，由于是二分类问题，所以只需要投影到一维即可

$y_i^j$ ：表示投影之后第*i*类的第*j*个样本

$z_i$ ：表示投影之后第*i*类的样本均值

$s_i$ ：表示投影之后第*i*类的样本方差

那么我们可以得到：

$$\begin{aligned}\mu_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j \\ S_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} (x_i^j - \mu_i)^2 = \frac{1}{N_i} \sum_{j=1}^{N_i} (x_i^j - \mu_i)(x_i^j - \mu_i)^T \\ y_i^j &= w^T x_i^j \\ z_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} y_i^j = \frac{1}{N_i} \sum_{j=1}^{N_i} w^T x_i^j = w^T \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j = w^T \mu_i\end{aligned}$$

$$\begin{aligned}s_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} (y_i^j - z_i)^2 = \frac{1}{N_i} \sum_{j=1}^{N_i} (w^T x_i^j - w^T \mu_i)(w^T x_i^j - w^T \mu_i)^T \\ &= w^T \frac{1}{N_i} \sum_{j=1}^{N_i} (x_i^j - \mu_i)(x_i^j - \mu_i)^T w = w^T S_i w\end{aligned}$$

我们考虑**类间距离最大**怎么用数学语言形式化表示。在二分类问题中，我们可以用投影之后两类的样本均值的距离来体现两类间的距离，也就是：

$$(z_1 - z_2)^2 = (w^T \mu_1 - w^T \mu_2)(w^T \mu_1 - w^T \mu_2)^T = w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w$$

我们再考虑**类内方差最小**的问题，就是使投影之后两个类别内的方差最小化的问题，那么我们直接表示成他们的和，也就是：

$$s_1 + s_2 = w^T S_1 w + w^T S_2 w = w^T (S_1 + S_2) w$$

要同时处理两个问题，一个最小化一个最大化，我们通常写成分数形式，那么我们的目标函数可以写为：

$$J(w) = \frac{w^T (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w}{w^T (S_1 + S_2) w}$$

我们令  $S_b = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ ， $S_w = S_1 + S_2$ ，则目标函数可以重写为：

$$J(w) = \frac{w^T S_b w}{w^T S_w w}$$

接下来我们就只需要最大化  $J(w)$  即可，即优化目标为：

$$\arg \max_w J(w) = \frac{w^T S_b w}{w^T S_w w}$$

这里使用两个方法进行优化问题的求解。

#### 方法一：拉格朗日乘子法

由于分子和分母都是关于  $w$  的二次项，因此上式的解与  $w$  的长度无关，只与方向有关。直观的理解就是，我们只需要找出投影基底的方向，而大小没有什么关系。那么我们可以

以不失一般性地令  $w^T S_w w = 1$ ，则优化目标等价于：

$$\begin{aligned} \min_w & -w^T S_b w \\ \text{s.t.} & w^T S_w w = 1 \end{aligned}$$

利用拉格朗日乘子法，设拉格朗日函数为：

$$L(\lambda, w) = -w^T S_b w + \lambda(w^T S_w w - 1)$$

令：

$$\frac{\partial L(\lambda, w)}{\partial w} = -2S_b w + 2\lambda S_w w = 0$$

得：

$$S_b w = \lambda S_w w$$

由于  $S_b w = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w$ ，且  $(\mu_1 - \mu_2)^T w$  是一个常数，那么  $S_b w$  的方向恒

为  $\mu_1 - \mu_2$ ，不妨令  $S_b w = \alpha(\mu_1 - \mu_2)$ ，则有：

$$\alpha(\mu_1 - \mu_2) = \lambda S_w w$$

即：

$$\frac{\alpha}{\lambda} S_w^{-1} (\mu_1 - \mu_2) = w$$

由于我们只关心  $w$  的方向，那么使得目标函数最优的方向  $w$  为  $S_w^{-1}(\mu_1 - \mu_2)$ 。

**方法二：广义瑞利商辅助求解**

广义瑞利商是指这样的函数：

$$R(A, B, x) = \frac{x^H A x}{x^H B x}$$

其中  $x$  为非零向量， $A, B$  是 *Hermitan* 矩阵。所谓的 *Hermitan* 矩阵，就是满足共轭转置

矩阵和自身相等的矩阵。如果  $A, B$  都为实矩阵，则满足  $A^T = A, B^T = B$  的矩阵则为 *Hermitan* 矩阵。

这个函数可以变换为瑞利商的格式，令  $x = B^{-1/2} x'$ ，原函数通过变换可以转化为：

$$R(A, B, x') = \frac{x'^H B^{-1/2} A B^{-1/2} x'}{x'^H x'}$$

根据瑞利商的性质，我们可以得到  $R(A, B, x')$  的最大值为  $B^{-1/2} A B^{-1/2}$  的最大特征值。

又因为我们的目标函数为：

$$\arg \max_w J(w) = \frac{w^T S_b w}{w^T S_w w}$$

可以神奇的发现这就是一个广义瑞利商的格式，由上面结论可以得到  $J(w')$  的最大值为

$S_w^{-1/2} S_b S_w^{-1/2}$  的最大特征值，对应的  $w'$  为  $S_w^{-1/2} S_b S_w^{-1/2}$  的最大特征值对应的特征向量。

又因为  $S_w^{-1} S_b$  的特征值与  $S_w^{-1/2} S_b S_w^{-1/2}$  的特征值相同，故转而求  $S_w^{-1} S_b$  的最大特征值对应的特征向量  $w$  即是最佳投影方向。进而可以转化成方法一的式子得出相同的结论。上面用两个方法推导出了二分类情况下 LDA 的使用方法。接下来推导多分类的情况。

### (3) 数学原理推导（多分类）

对于上面的二分类问题，我们只是讨论了降低到一维的情况，但是现在类别数（假设有  $C$  类）多了，只有一个维度可能分离效果不是很好。我们尝试增加选取基的数量，假设我们选取了  $k$  个基向量，组成了一个矩阵  $W = (w_1, w_2, \dots, w_k)$ ，那么采用二分类中的符号定义有：

$$\begin{aligned} y_i^j &= W^T x_i^j \\ z_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} y_i^j = \frac{1}{N_i} \sum_{j=1}^{N_i} W^T x_i^j = W^T \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j = W^T \mu_i \\ s_i &= \frac{1}{N_i} \sum_{j=1}^{N_i} (y_i^j - z_i)^2 = \frac{1}{N_i} \sum_{j=1}^{N_i} (W^T x_i^j - W^T \mu_i)(W^T x_i^j - W^T \mu_i)^T \\ &= W^T \frac{1}{N_i} \sum_{j=1}^{N_i} (x_i^j - \mu_i)(x_i^j - \mu_i)^T W = W^T S_i W \end{aligned}$$

首先我们考虑**类内距离最小**，和二分类问题类似，我们用投影之后所有类别类内方差之和来衡量这一指标，也就是：

$$\sum_{i=1}^C s_i = \sum_{i=1}^C W^T S_i W = W^T \sum_{i=1}^C S_i W$$

然后我们考虑**类间距离最大**，与二分类不同的是，这里如果用每两类之间的距离去衡量的话，并不能体现所有类间距离的关系，我们采用一种新的衡量方法，定义  $\mu$  是**所有**样本的均值， $z$  为投影之后**所有**样本的均值，那么有：

$$\begin{aligned} \mu &= \sum_{i=1}^C \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j \\ z &= \sum_{i=1}^C \frac{1}{N_i} \sum_{j=1}^{N_i} y_i^j = \sum_{i=1}^C \frac{1}{N_i} \sum_{j=1}^{N_i} W^T x_i^j = W^T \sum_{i=1}^C \frac{1}{N_i} \sum_{j=1}^{N_i} x_i^j = W^T \mu \end{aligned}$$

我们用投影之后每一类中心点与所有类中心点的距离之和来衡量类间的距离，由于每一类的样本点数量可能不同，我们把每一类中心点与所有类中心点的距离加上权值之后再求和，如下，用这个和来衡量类间距离：

$$\sum_{i=1}^C N_i (z_i - z)(z_i - z)^T = \sum_{i=1}^C N_i (W^T \mu_i - W^T \mu)(W^T \mu_i - W^T \mu)^T = W^T \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T W$$

类似的，我们令：

$$S_w = \sum_{i=1}^C S_i$$

$$S_b = \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

与二分类类似地，我们定义目标函数为：

$$J(w) = \frac{\text{tr}(W^T S_b W)}{\text{tr}(W^T S_w W)}$$

优化目标为：

$$\arg \max_w J(w) = \frac{\text{tr}(W^T S_b W)}{\text{tr}(W^T S_w W)}$$

不难证明这个形式是之前二分类目标函数形式的推广，证明如下：

由于  $W = (w_1, w_2, \dots, w_k)$ ，故：

$$\begin{cases} \text{tr}(W^T S_b W) = \sum_{i=1}^k w_i^T S_b w_i \\ \text{tr}(W^T S_w W) = \sum_{i=1}^k w_i^T S_w w_i \end{cases}$$

那么优化目标可以变为：

$$\arg \max_w J(w) = \frac{\sum_{i=1}^k w_i^T S_b w_i}{\sum_{i=1}^k w_i^T S_w w_i}$$

故是之前讲过的优化函数的推广。

那么与二分类一样的思路，我们将分母固定为 1，即：

$$\begin{aligned} \min_w & -\text{tr}(W^T S_b W) \\ \text{s.t.} & \text{tr}(W^T S_w W) = 1 \end{aligned}$$

使用拉格朗日乘子法，定义拉格朗日函数为：

$$L(\lambda, W) = -\text{tr}(W^T S_b W) + \lambda(\text{tr}(W^T S_w W) - 1)$$

对上式关于  $W$  求偏导有：

$$\frac{\partial L(\lambda, W)}{\partial W} = -\frac{\partial(\text{tr}(W^T S_b W))}{\partial W} + \lambda \frac{\partial(\text{tr}(W^T S_w W) - 1)}{\partial W}$$

由矩阵微分公式， $\frac{\partial \text{tr}(X^T A X)}{\partial X} = (A + A^T)X$  代入上式得：

$$\frac{\partial L(\lambda, W)}{\partial W} = -(S_b + S_b^T)W + \lambda(S_w + S_w^H)W$$

由于：

$$\begin{aligned}S_b &= S_b^T \\ S_w &= S_w^H\end{aligned}$$

故：

$$\frac{\partial L(\lambda, W)}{\partial W} = -2S_b W + 2\lambda S_w W$$

令上式等于 0 得：

$$S_b W = \lambda S_w W$$

即：

$$S_w^{-1} S_b W = \lambda W$$

所以使得目标函数最大的  $W$  为  $S_w^{-1} S_b$  前  $k$  大特征值对应的特征向量组成的矩阵。

#### (4) 证明 $S_t = S_b + S_w$

我们定义全局散度矩阵（其实就是 PCA 的优化目标）：

$$S_t = \sum_{i=1}^C \sum_{j=1}^{N_i} (x_i^j - \mu)(x_i^j - \mu)^T$$

由于：

$$\begin{aligned}S_w &= \sum_{i=1}^C S_i \\ S_b &= \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T\end{aligned}$$

我们只需证明  $S_b = S_t - S_w$  即可。

证明过程如下：



$$\begin{aligned}
& S_t - S_w \\
&= \sum_{i=1}^C \sum_{j=1}^{N_i} (x_i^j - \mu)(x_i^j - \mu)^T - \sum_{i=1}^C \sum_{j=1}^{N_i} (x_i^j - \mu_i)(x_i^j - \mu_i)^T \\
&= \sum_{i=1}^C \sum_{j=1}^{N_i} ((x_i^j - \mu)(x_i^j - \mu)^T - (x_i^j - \mu_i)(x_i^j - \mu_i)^T) \\
&= \sum_{i=1}^C \sum_{j=1}^{N_i} ((x_i^j - \mu)(x_i^{jT} - \mu^T) - (x_i^j - \mu_i)(x_i^{jT} - \mu_i^T)) \\
&= \sum_{i=1}^C \sum_{j=1}^{N_i} (x_i^j x_i^{jT} - x_i^j \mu^T - \mu x_i^{jT} + \mu \mu^T - x_i^j x_i^{jT} + x_i^j \mu_i^T + \mu_i x_i^{jT} - \mu_i \mu_i^T) \\
&= \sum_{i=1}^C \sum_{j=1}^{N_i} (-x_i^j \mu^T - \mu x_i^{jT} + \mu \mu^T + x_i^j \mu_i^T + \mu_i x_i^{jT} - \mu_i \mu_i^T) \\
&= \sum_{i=1}^C (\sum_{j=1}^{N_i} -x_i^j \mu^T - \sum_{j=1}^{N_i} \mu x_i^{jT} + \sum_{j=1}^{N_i} \mu \mu^T + \sum_{j=1}^{N_i} x_i^j \mu_i^T + \sum_{j=1}^{N_i} \mu_i x_i^{jT} - \sum_{j=1}^{N_i} \mu_i \mu_i^T) \\
&= \sum_{i=1}^C (-N_i \mu_i \mu^T - N_i \mu \mu_i^T + N_i \mu \mu^T + N_i \mu_i \mu_i^T + N_i \mu_i \mu_i^T - N_i \mu_i \mu_i^T) \\
&= \sum_{i=1}^C N_i (-\mu_i \mu^T - \mu \mu_i^T + \mu \mu^T + \mu_i \mu_i^T) \\
&= \sum_{i=1}^C N_i (\mu_i - \mu)(\mu_i - \mu)^T = S_b
\end{aligned}$$

## (5) LDA 算法流程

### 二分类

输入：数据集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ，其中  $x_i$  为任意  $n$  维向量，总共 2 类

样本，那么  $y_i \in [1, 2]$

输出：降维向量  $w$

- 首先计算每一类的样本均值向量  $\mu_1, \mu_2$
- 然后计算类内散度矩阵  $S_w$
- 计算向量  $w = S_w^{-1}(\mu_1 - \mu_2)$

### 多分类

输入：数据集  $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ ，其中  $x_i$  为任意  $n$  维向量，假设总共

有  $C$  类样本，那么  $y_i \in [1, C]$ ，假设降低到  $k$  维

输出：降维矩阵  $W$

- 首先计算类内散度矩阵  $S_w$  和类间散度矩阵  $S_b$

- 然后计算矩阵  $S_w^{-1}S_b$
- 计算  $S_w^{-1}S_b$  的最大  $k$  个特征值对应的特征向量组成投影矩阵  $W$

## 2. LDA 编程实现与识别率测试

### (1) 二分类

对于二分类问题，我们需要计算的投影向量为  $w = S_w^{-1}(\mu_1 - \mu_2)$ 。代码实现上，首先

我是把两个类的数据矩阵拆分出来，顺便计算两个类的样本数：

```
X1 = []; % class one
X2 = []; % class two
N1 = 0; % number of class one
N2 = 0; % number of class two
for i = 1 : col
    if (X_Label(i) == 1)
        X1 = [X1 X(:, i)];
        N1 = N1 + 1;
    else
        X2 = [X2 X(:, i)];
        N2 = N2 + 1;
    end
end
```

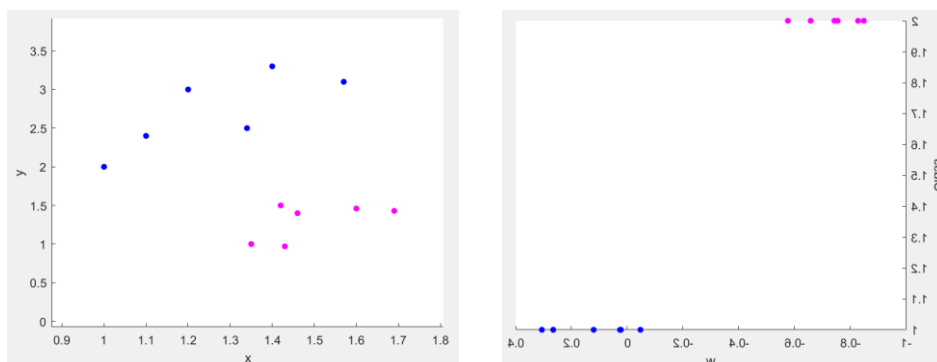
然后分别计算两个类的均值向量与类内散度矩阵，最后得到用于投影的向量方向：

```
% each class mean vector
mean1 = mean(X1, 2);
mean2 = mean(X2, 2);

% calculate Sw
S1 = (X1 - repmat(mean1, 1, N1)) * (X1 - repmat(mean1, 1, N1))' / N1;
S2 = (X2 - repmat(mean2, 1, N2)) * (X2 - repmat(mean2, 1, N2))' / N2;
Sw = S1 + S2;

% calculate w
w = Sw^(-1) * (mean1 - mean2);
w = w / norm(w);
```

我使用了一个小数据对针对于二分类的 LDA 进行了测试，测试效果如下，左边为 LDA 投影前的原始数据分布（二维数据），右边为投影之后的每一类在投影方向上的分布（投影到一维空间中）：



可以看到，投影之后的样本类内尽可能聚拢，类间尽可能分散

### (2) 多分类人脸识别

对于多分类 LDA，首先需要计算类内散度矩阵  $S_w$ ，就是把所有类的协方差矩阵都加起

来得到，那么首先把每一类的矩阵划分出来，求出他们的协方差矩阵，然后加到  $S_w$  上

即可，代码实现上顺便存储了每一类的均值向量，以便下面  $S_b$  的计算。

```
%% calculate Sw
means = []; % mean of each class
Sw = zeros(feature_size, feature_size);
tmp = 1;
for i = 1 : class_cnt
    tmp_X = X(:, tmp : tmp + N(i) - 1);
    means = [means mean(tmp_X, 2)];
    S = (tmp_X - repmat(means(:, i), 1, N(i))) * (tmp_X - repmat(means(:, i), 1, N(i)))';
    Sw = Sw + S;
    tmp = tmp + N(i);
end
```

然后计算类间散度矩阵  $S_b$ ，首先求出总体均值向量，然后把每一类的均值向量减去总体均值向量之后求他们的平方和即可：

```
%% calculate Sb
Sb = zeros(feature_size, feature_size);
miu = mean(X, 2); % mean of all training samples
for i = 1 : class_cnt
    Sb = Sb + (means(:, i) - miu) * (means(:, i) - miu)' * N(i);
end
```

之后我们需要求出  $S_w^{-1}S_b$  的特征值与特征向量，由于  $S_w$  不一定可逆，可能是一个奇异

矩阵，我们可以利用  $S_bW = \lambda S_wW$  求出  $S_b$  关于  $S_w$  的广义特征向量。

```
%% calculate eigen vector and W
[W, D] = eig(Sb, Sw);
```

### (3) 多分类人脸识别数据测试，LDA 与 PCA 对比。

#### 实验一：

使用了三个数据集，分别为 ORL、AR 和 FERET 进行测试。对于 ORL 数据集，使用了 40 类人脸，每一类人脸的前 7 张作为训练集，后 3 张作为测试集。对于 AR 数据集，使用了 40 类人脸，每一类人脸的前 20 张作为训练集，后 6 张作为测试集。对于 FERET 数据集，使用了 40 类人脸，每一类人脸的前 5 张作为训练集，后 2 张作为测试集。使用 LDA 分别降维到 1、4、7、...、46 维进行识别率测试，使用 PCA 分别降低到与 LDA 相同的维度进行识别率测试，分类器采用 KNN 分类器，K 取值为 4。测试结果如下：

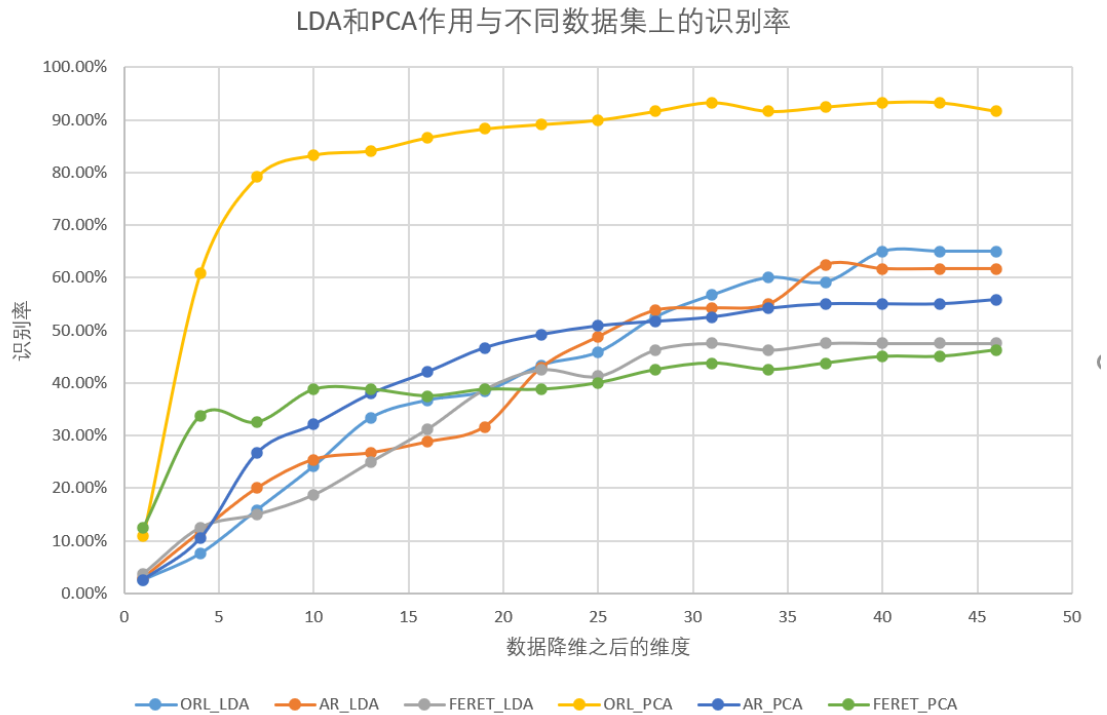
LDA 识别率随维数的变化情况表

Rate	data's dimension after LDA projection															
database	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
ORL	2.50%	7.50%	15.83%	24.17%	33.33%	36.67%	38.33%	43.33%	45.83%	52.50%	56.67%	60.00%	59.17%	65.00%	65.00%	65.00%
AR	2.92%	11.67%	20.00%	25.42%	26.67%	28.75%	31.67%	42.92%	48.75%	53.75%	54.17%	55.00%	62.50%	61.67%	61.67%	61.67%
FERET	3.75%	12.50%	15.00%	18.75%	25.00%	31.25%	38.75%	42.50%	41.25%	46.25%	47.50%	46.25%	47.50%	47.50%	47.50%	47.50%

PCA 识别率随维数的变化情况表

Rate	data's dimension after PCA projection															
database	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
ORL	10.83%	60.83%	79.17%	83.33%	84.17%	86.67%	88.33%	89.17%	90.00%	91.67%	93.33%	91.67%	92.50%	93.33%	93.33%	91.67%
AR	2.50%	10.42%	26.67%	32.08%	37.92%	42.08%	46.67%	49.17%	50.83%	51.67%	52.50%	54.17%	55.00%	55.00%	55.00%	55.83%
FERET	12.50%	33.75%	32.50%	38.75%	38.75%	37.50%	38.75%	38.75%	40.00%	42.50%	43.75%	42.50%	43.75%	45.00%	45.00%	46.25%

LDA 和 PCA 在不同数据集下的识别率变化图



可以看到，PCA 在降低到相同维数的情况下，识别率在大多数情况下要优于 LDA，但也有少数情况下，LDA 利用类别信息使得识别率比 PCA 高。但是两种算法单独作用在 AR 数据集和 FERET 数据集上的识别效果都不是很好。

#### 实验二：

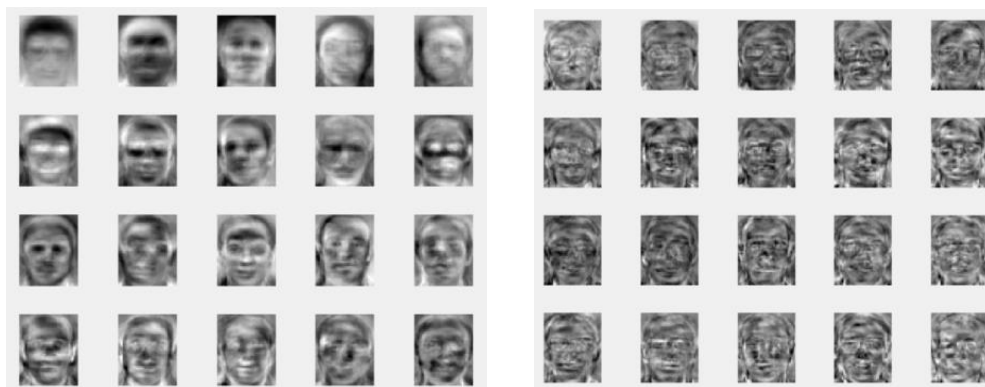
之后我把两种方法进行了结合并测试识别率，先用 PCA 降维去除噪声，再用 LDA 进行二次降维，在 AR 数据集和 FERET 数据集上（两种数据集的样本选取同实验一）测试了 PCA 降低到 100 维之后，使用 LDA 降低到不同维度的识别率：

Rate	data's dimension after LDA projection															
database	1	4	7	10	13	16	19	22	25	28	31	34	37	40	43	46
AR	12.50%	56.25%	77.50%	82.50%	84.58%	87.92%	88.75%	91.67%	93.33%	94.17%	96.25%	95.83%	95.42%	95.42%	93.75%	90.42%
FERET	13.75%	43.75%	48.75%	57.50%	62.50%	66.25%	66.25%	65.00%	65.00%	66.25%	68.75%	72.50%	70.00%	73.75%	75.00%	68.75%

我们发现，在两个数据集上的识别率都得到了较好的提升。

### 3. 比较 eigenface 与 fisherface 的不同，并做数据可视化

#### (1) eigenface 与 fisherface 的比较



**Eigenface**

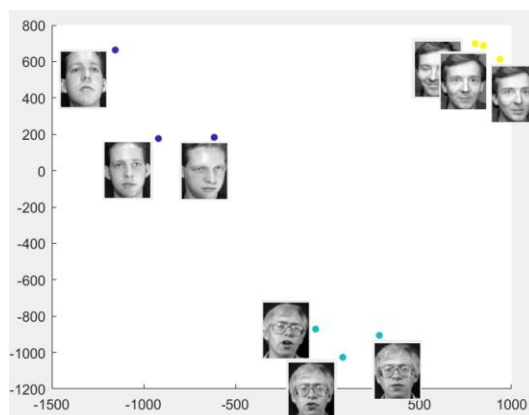
**fisherface**

我们发现，eigenface 更加注重人物特征提取，fisherface 更加注重人物类别分类

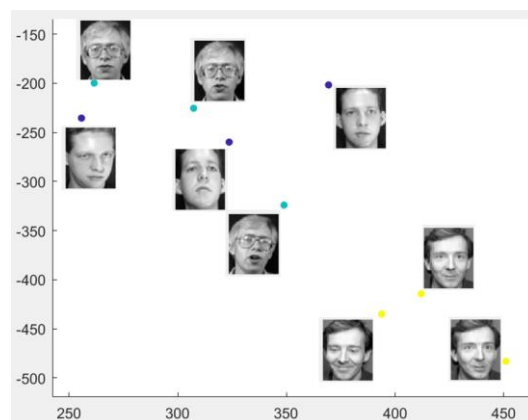
## (2) 二维与三维可视化

使用 ORL 数据集的前三类人脸，每一类采用前 7 张人脸作为训练集，后 3 张人脸作为测试集，分别做二维与三维数据可视化，对比 PCA 和 LDA 的不同。

二维数据:

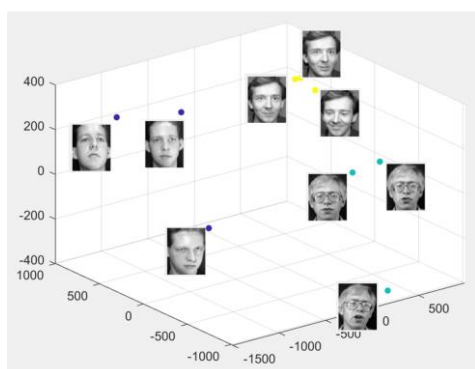


PCA

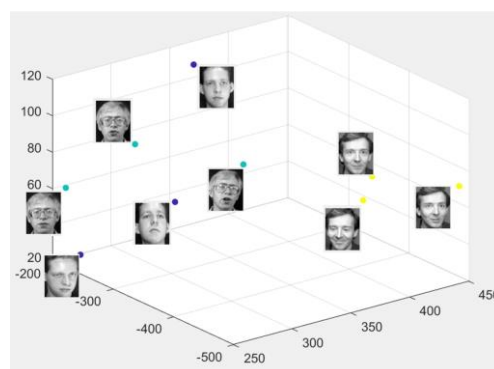


LDA

三维数据:



PCA



LDA

我们可以发现，无论是投影到二维空间还是三维空间上，LDA 的效果都不是很好，由于测试集的图片受光线的影响和人脸姿势的影响较大，我们可以初步判定，单纯用 LDA 降维受图片光照和人脸姿态影响较大，而 PCA 较 LDA 而言，鲁棒性较强一些。

## 四、实验结论或体会

本次实验中，在写代码的时候，我发现虽然说包装了很多函数，但是有时候在调用的时候还是会产生许多冗余的代码，所以我把 PCA 和 LDA 的经典模型包装成了更全面的函数，接受训练集，返回训练完的模型，模型中包括了投影所需要的矩阵等一系列东西。就如下图所示：

```
function model = LDA(X_train, Label_train, components)
% Inputs:
%   -X_train: training set, [dimension x data_num]
%   -Label_train: labels of training set, [1 x data_num]
%   -components: number of components to keep, [int]
% Outputs:
%   -model:
%       model.components: number of components in this model, [int]
%       model.W: the matrix for projection by LDA, [dimension x components]
% Author: Yang Zheng
% Date: 2022. 3. 24
```

在这次实验中，我对于数学推导更加的严谨细致，掌握了拉格朗日乘子法等推导方法。

对比了 PCA 和 LDA，PCA 是无监督的，目的是为了去除原始数据中冗余的信息，让投影之后的数据各个维度的方差尽可能大，不同维度间的协方差为零。LDA 是有监督的，目的是使得投影之后的数据不同类别尽可能分开，同类别数据尽可能聚拢。

对于 PCA 和 LDA 的相同点也进行了分析，两者均可以对数据进行降维，两者在降维时均使用了矩阵特征分解的思想，两者都假设数据符合高斯分布。

同时清楚了 LDA 的局限性，LDA 最多只能把数据维数降低到类别数减一，可能过拟合数据。在人脸识别上，受光线，人脸姿态影响较大。

不足之处是，还是没什么时间看论文，没有提出自己的新方法。感觉自己用了太多时间在实验上，没有仔细地阅读论文，接下来会尽力把论文都仔细阅读一遍。

<p>指导教师批阅意见：</p>	
<p>成绩评定：</p>	
<p>指导教师签字：</p>	
<p>年 月 日</p>	
<p>备注：</p>	

<p>指导教师批阅意见：</p>	
<p>成绩评定：</p>	
<p>指导教师签字：</p>	
<p>年 月 日</p>	
<p>备注：</p>	

<p>指导教师批阅意见：</p>	
<p>成绩评定：</p>	
<p>指导教师签字：</p>	
<p>年 月 日</p>	
<p>备注：</p>	

<p>指导教师批阅意见：</p>	
<p>成绩评定：</p>	
<p>指导教师签字：</p>	
<p>年 月 日</p>	
<p>备注：</p>	

<p>指导教师批阅意见：</p>	
<p>成绩评定：</p>	
<p>指导教师签字：</p>	
<p>年 月 日</p>	
<p>备注：</p>	

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。