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（腾班）

指导教师 赖志辉

报 告 人 黄亮铭 学号 2022155028

实验时间 2024 年 3 月 18 日至 2024 年 3 月 24 日

实验报告提交时间 2024 年 3 月 24 日

教务处制

一、实验目的与要求

1. 熟悉机器学习的基础算法之一 LDA；
2. 学习从零实现 LDA 算法；
3. 了解不同监督学习算法之间的优劣性。

二、实验内容与方法

1. 简述 LDA 原理、 算法模型与优化问题，并给出求解的全程推导细节（拉格朗日乘子法）；证明 St=Sb+Sw；

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。

4.参看前人其它论文实现一个新的线性鉴别分析方法或子空间学习方法，并与 PCA,LDA 做比较实验，打印出各方法训练样本数的增长其识别曲线变化图等各种情况，把简要内容写在本实验报告中

三、实验步骤与过程

1. 简述 LDA 原理、 算法模型与优化问题，证明 St=Sb+Sw。

I. 简述 LDA 原理。

LDA（线性判别分析）利用了样本的类别（数据）标签，为有监督学习，是主流的一种线性降维算法。以最小化类内方差，最大化类间方差为目标导向，通过降维（投影），达到降维的目的更好地将样本分类。

假设我们有两类数据 分别为红色和蓝色，如下图所示，这些数据特征是二维的，我们希望将这些数据投影到一维的一条直线，让每一种类别数据的投影点尽可能的接近，而红色和蓝色数据中心之间的距离尽可能的大。

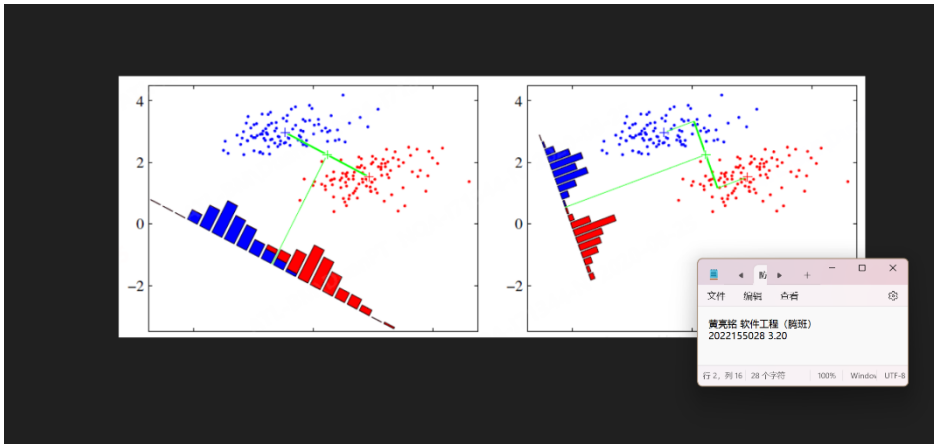


图 1

从直观上可以看出，右图要比左图的投影效果好，因为右图的黑色数据和蓝色数据各个较为集中，且类别之间的距离明显。左图则在边界处数据混杂。以上就是LDA的主要思想了，当然在实际应用中，我们的数据是多个类别的，我们的原始数据一般也是超过二维的，投影后的也一般不是直线，而是一个低维的超平面。

LDA的数学原理：

类内散度： $S_W=\sum_{i=1}^{c}S_i$

类间散度： $S_B=\sum_{i=1}^{c}N_i(\mu_i-\mu)(\mu_i-\mu)^T$

我们的优化目标可以描述为： $max\, w\frac{|w^TS_Bw|}{|w^TS_Ww|}$，*这是一个广义瑞利商，令* $J=max\, w\frac{|w^TS_Bw|}{|w^TS_Ww|}$，*对J求导我们可以得到：* $S_bu=\lambda S_wu$

由上式我们可以得到： $S_w^{-1}S_bu=\lambda u$

计算出 $S_w^{-1}S_b$ *的* k *个最大特征值和* k *个对应特征向量即可得到投影矩阵* W

II. LDA算法模型。

1. 将数据集分类，计算每个类的均值；
2. 计算类间散度矩阵Sb与类内散度矩阵Sw；
3. 构造目标函数（多种不同的目标函数）并对其进行特征分解；
4. 取出一定数量的特征向量得到投影矩阵；
5. 将测试数据投影到子空间中，使用KNN进行分类（实际问题中）。

III. LDA优化问题。

缺点：① LDA不适合处理非线性问题；② LDA降维最多降到类别数k-1的维数，如果我们降维的维度大于k-1，则不能使用LDA。；③ LDA处理高维数据时，计算代价很大；④ LDA可能过度拟合数据。

优化方法：对于①，我们可以使用带有核函数的LDA升维处理数据；对于③，先用PCA预处理，再使用LDA；对于④，我们可以加入正则项，用于防止过拟合。

IV. 证明$S_T=S_W+S_B$。

$$S_W=\sum_{i=1}^{c}S_i$$

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

$$S_T=\sum_{i=1}\sum_{x\in X_i}(x-\mu)(x-\mu)^T$$

$$=\sum_{i=1}\sum_{x\in X_i}[(x-\mu_i)+(\mu_i-\mu)][(x-\mu_i)+(\mu_i-\mu)]^T$$

$$=i=1\sum c x\in X_i\sum[(x-\mu i)(x-\mu i)T+(\mu i-\mu)(x-\mu i)T+(x-\mu i)(\mu i-\mu)T+(\mu i-\mu)(\mu i-\mu)T]$$

$$=i=1\sum c x\in X_i\sum(x-\mu i)(x-\mu i)T+i=1\sum c x\in X_i\sum(\mu i-\mu)(x-\mu i)T+i=1\sum c x\in X_i\sum(x-\mu i)(\mu i-\mu)T+i=1\sum c x\in X_i\sum(\mu i-\mu)(\mu i-\mu)T$$

$$=S_W+S_B$$

所以，得证$S_T = S_W + S_B$。

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。

I. 数据的导入

利用 imread 导入数据集，并在导入时将每个人脸拉成列向量；同时分割数据集，将前 30%作为测试数据，后 70%作为训练数据。（这里只给出 AR 数据集的导出代码截图，其他的数据集代码类似）

```
clear all;
% 1.数据集导入
reshaped_faces=[];
database_name = "AR";
%AR5040
if (database_name == "AR")
    for i=1:40
      for j=1:10
        if(i<10)
           pic=imread(strcat('C:\Users\黄亮铭\Desktop\大学课程\机器学习\实验1PCA\AR
        else
            pic=imread(strcat('C:\Users\黄亮铭\Desktop\大学课程\机器学习\实验1PCA\A
        end
        reshaped_pic = reshape(pic,2000,1);
        reshaped_pic=double(reshaped_pic);
        reshaped_faces=[reshaped_faces, reshaped_pic];
      end
    end
row = 50;
col = 40;
people_num = 40;
each_pic_num = 10;
each_train_pic_num = 7;
each_test_pic_num = 3;
end
% ORL5646
```

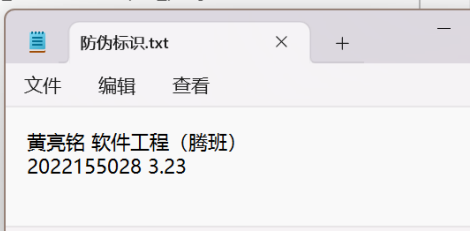


图 1

```
% 取出前30%作为测试数据，剩下70%作为训练数据
test_data_idx = [];
train_data_idx = [];
for i=0:people_num-1
    test_data_idx = [test_data_idx each_pic_num*i+1:each_pic_num*i+each_test_pic
    train_data_idx = [train_data_idx each_pic_num*i+each_test_pic_num+1:each_pic
end

train_data = reshaped_faces(:,train_data_idx);
test_data = reshaped_faces(:, test_data_idx);
dimension = row * col;

% 2.数据预处理
% 求类均值
```

图 2

II．数据预处理

计算不同类类内的均值、散度矩阵，以及类间的散度矩阵。

```
% 2.数据预处理
% 求类均值
k = 1;
class_mean = zeros(dimension, people_num);
for i=1:people_num
    tmp = class_mean(:,i);
    for j=1:each_train_pic_num
        tmp = tmp + train_data(:,k);
        k = k + 1;
    end
    class_mean(:,i) = tmp / each_train_pic_num;
end

% 求类间散度矩阵Sb
Sb = zeros(dimension, dimension);
all_mean = mean(train_data, 2); % 全部的平均
for i=1:people_num
    % 以每个人的平均脸进行计算，同时中心化
    cen_data = class_mean(:,i) - all_mean;
    Sb = Sb + cen_data * cen_data';
end
Sb = Sb / people_num;

% 求类内散度矩阵Sw
Sw = zeros(dimension, dimension);
k = 1;
for i=1:people_num
    for j=1:each_train_pic_num
        cen_data = train_data(:,k) - class_mean(:,i);
        Sw = Sw + cen_data * cen_data';
        k = k + 1;
    end
end
Sw = Sw / (people_num * each_train_pic_num);
```

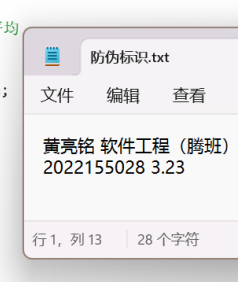


图 3

III．构造目标函数，并对目标函数进行特征值分解。

每个目标函数分别对应不同 LDA 的等价模型以及经典 PCA 模型。（每一次选择一个目标函数，并将其他的目标函数注释，一共跑 6 次）

```
% 3.构造目标函数，并对目标函数进行特征值分解
% 目标函数一：经典LDA
target = pinv(Sw) * Sb;
% 目标函数二：不可逆时需要正则项扰动
Sw = Sw + eye(dimension)*10^-6;
target = Sw^-1 * Sb;
% 目标函数三：相减形式
target = Sb - Sw;
% 目标函数四：相除
target = Sb/Sw;
% 目标函数五：调换位置
target = Sb * pinv(Sw);
%PCA
cen_face = (train_data - all_mean);
cov_matrix = cen_face * cen_face';
target = cov_matrix;
% 求特征值、特征向量
[eigen_vectors, dianogol_matrix] = eig(target);
eigen_values = diag(dianogol_matrix);
% 对特征值、特征向量进行排序
[sorted_eigen_values, index] = sort(eigen_values, 'descend');
eigen_vectors = eigen_vectors(:, index);
```

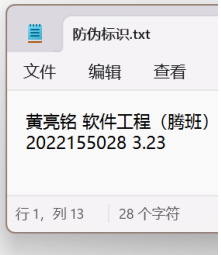


图 4

IV．实现人脸识别

使用 KNN 进行分类预测，实现人脸识别，同时比较不同 LDA 的等价模型和 PCA 的人脸识别率。（投影到 10、20……160 维）

```
% 4人脸识别
index = 1;
X = [];
Y = [];
for i=10:10:160
    project_matrix = eigen_vectors(:,1:i);
    projected_train_data = project_matrix' * (train_data - all_mean);
    projected_test_data = project_matrix' * (test_data - all_mean);

    K=1;
    % 用于保存数据结果
    min_k_values = zeros(K,1);
    min_k_values_label = zeros(K,1);

    test_face_number = size(projected_test_data, 2);
    correct_pre_number = 0;

    for each_test_face_index = 1:test_face_number
        each_test_face = projected_test_data(:,each_tes
        % 先把k个值填满，避免在迭代中反复判断
```

防伪标识.txt
文件 编辑 查看
黄亮铭 软件工程（腾班）
2022155028 3.23
行 1，列 13 28 个字符

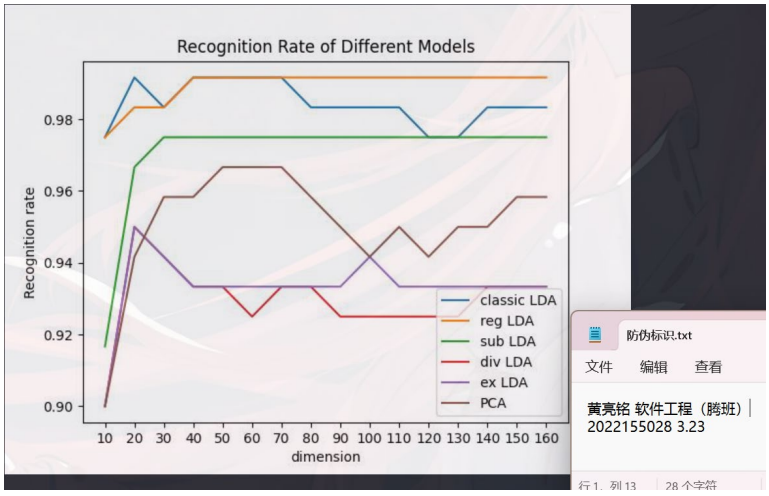
图 5

```
    for each_test_face_index = 1:test_face_number
        each_test_face = projected_test_data(:,each_test_face_index);
        % 先把k个值填满，避免在迭代中反复判断
        for each_train_face_index = 1:K
            min_k_values(each_train_face_index,1) = norm(each_test_face - projec
            min_k_values_label(each_train_face_index,1) = floor((train_data_idx(
        end

        [max_value, index_of_max_value] = max(min_k_values);

        % 计算与剩余每一个已知人脸的距离
        for each_train_face_index = K+1:size(projected_train_data,2)
            dis = norm(each_test_face - projected_train_data(:,each_train_face_i
            % 判断是否更新标签
            if (dis < max_value)
                min_k_values(index_of_max_value,1) = dis;
                min_k_values_label(index_of_max_value,1) = floor((train_data_idx
                [max_value, index_of_max_value] = max(min_k_values);
            end
        end

        % 得到结果
        predict_label = mode(min_k_values_label);
        real_label = floor((test_data_idx(1,each_test_face_index) - 1) / each_pi

        if (predict_label == real_label)
            correct_pre_number = correct_pre_number + 1
        end
    end

    correct_rate = correct_pre_number/test_face_number;
    X = [X i];
    Y = [Y correct_rate];

    if (i == 160)
        waitfor(plot(X,Y));
    end
end
```

防伪标识.txt
文件 编辑 查看
黄亮铭 软件工程（腾班）
2022155028 3.23
行 1，列 13 28 个字符

图 6

不同的 LDA 等价模型和 PCA 的人脸识别率在不同维度下的对比如下图：可以看出经典 LDA 和正则 LDA 的识别率较高并且比较稳定，其他 LDA 等价模型和 PCA 在低维度识别较低，并且在高维下识别率稳定性较差（除相见形式的 LDA）。其中，调换 LDA 和相除 LDA 的识别率低于 PCA，其他 LDA 等价模型的识别率均高于 PCA。



防伪标识.txt
文件 编辑 查看
黄亮铭 软件工程（腾班）
2022155028 3.23

图 7

该图使用了 python 的 matplotlib 库，代码如下：

```python
import matplotlib.pyplot as plt
import numpy as np


if __name__ == '__main__':
    with open("E:\\VScode\\C++Code\\ML\\data.txt") as fp:
        lines = fp.readlines()
        title = ["classic LDA", "reg LDA", "sub LDA", "div LDA", "ex LDA",
                 "PCA"]
        X = np.array([x for x in range(10, 170, 10)])
        Y = []
        for i in range(len(lines)):
            Y.append(np.array([float(x) for x in lines[i].split()]))

        fig, ax = plt.subplots()
        ax.set_title("Recognition Rate of Different Models")

        for i in range(len(lines)):
            ax.plot(X, Y[i], label=title[i])

        ax.set_xlabel("dimension")
        ax.set_ylabel("Recognition rate")
        ax.legend()

        plt.xticks(np.linspace(10, 160, 16))
        plt.savefig("E:\\VScode\\C++Code\\ML\\result.jpg")
        plt.show()
```


图 8

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。

I. LDA 与 PCA 图像降维与可视化对比（二维）

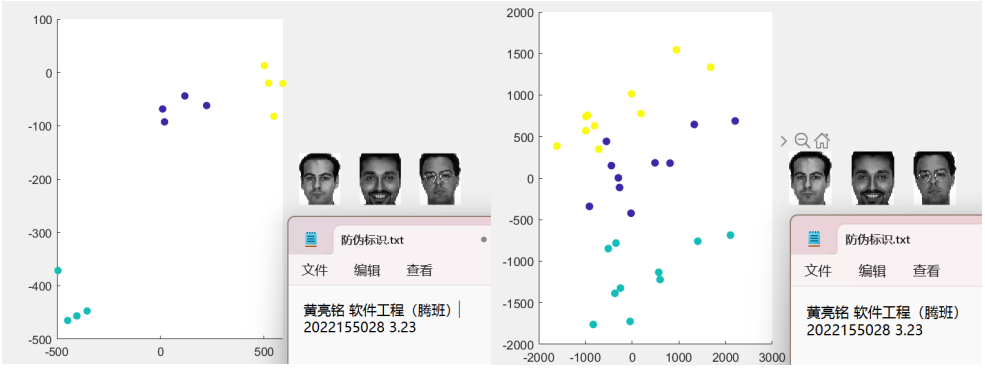


图 9：左边 LDA，右边 PCA

II. LDA 与 PCA 图像降维与可视化对比（三维）

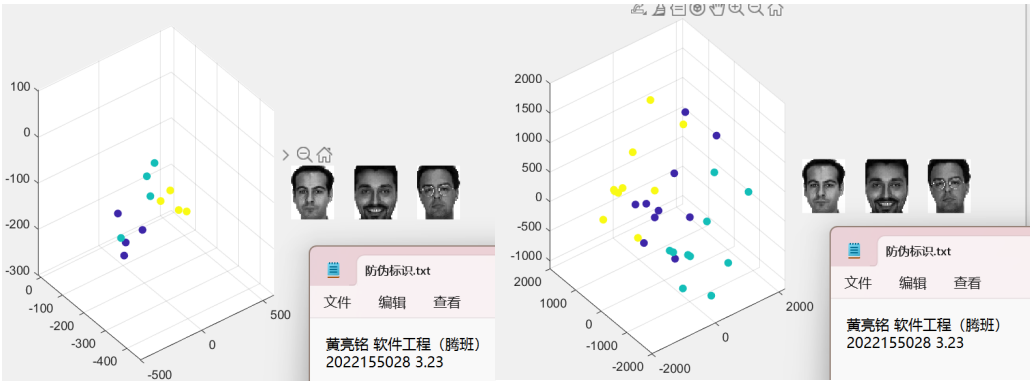


图 10：左边 LDA，右边 PCA

III. 比较 LDA 与 PCA 结果的异同。

① 无论是在二维还是在三维，都可以明显看出 LDA 同类图像较为聚集，PCA 相对混乱，没有明显规律。由 LDA 和 PCA 的算法目的及原理，可以比较出 eigenface 和 fisherface 的不同。

② LDA 和 PCA 都是降维技术，并且都是线性变换；

③ LDA 的目标是最小类内散度，最大类间散度，而 PCA 的目标是最小化重构误差；

④ LDA 是一种监督学习方法，可以利用类别标签的信息来优化降维过程，而 PCA 是一种无监督学习方法；

四、实验结论或体会

1. 在小范围数据集中，PCA 的效果与 LDA 相近，甚至超过 LDA；但是，在大范围数据集中，LDA 明显优于 PCA，特别是正则化 LDA 和经典 LDA。

2. 通过本次实验，我了解了不同 LDA 的等价模型及它们在不同数据集下的表现。

3. 通过本次实现，我了解了同为线性降维算法的 LDA 和 PCA 之间的相同之处和不同之处。

4. 通过本次实验，我了解了 LDA 存在的问题，例如：有限投影轴问题、高维数据计算代价大问题等问题。

指导教师批阅意见： 成绩评定： 指导教师签字： 年　　月　　日
备注：

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。