

专业： 信息工程
姓名： _____
学号： _____
日期： _____

浙江大学实验报告

课程名称： 矩阵论 指导老师： 程磊 成绩： _____
实验名称： 用 SVD 进行人脸数据分析 实验类型： _____ 同组学生姓名： _____

一、问题描述

结合课上学习到的 SVD 相关的知识, 学习了解 PCA(Principle Component Analysis)方法, 并利用给定的 Extend Yale B Faces 数据集进行人脸的特征分析和预处理。具体任务和示例如后面各小节所示, 要求提交一份代码以及报告, 说明各任务是如何实现的, 以及自己的理解。

二、问题背景

PCA 方法

1、目标问题：

给定 m 个 n 维样本 $X = \{x_0, x_0, \dots, x_m\}$, 希望找到一个变换 $z = Ux$ ($U_{k \times n}$ 为变换矩阵), 使得样本从 n 维降到 k 维, 同时最大程度减少降维带来的信息损失。

2、问题分析

原则 1: 降维之后各维度之间不相关, 即对原样本维度间去相关。

PCA 方法认为, 低维数据可以信息损失较小地表征高维数据的前提是, 高维数据由于维度之间的相关性导致了冗余信息, 即实则并不需要那么多维度就能表征绝大部分数据空间。因此, 原则 1 要求降维后各维度之间不相关。

原则 2: 最大化不同维度间的方差。

PCA 方法认为, 维度间的方差表征了不同维度基向量之间的差异度, 而各维度的基向量之间的差异体现了该维度是“特征”。维度间的方差越大, 则所保留的特征信息也越多,

假设原数据向量 $E(x_i) = 0$, 则降维后的数据向量 $E(z_i) = 0$

考察降维后样本集的协方差矩阵 $B_{k \times k} = \frac{1}{m} ZZ^T$

由原则 1, $B_{k \times k}$ 为对角阵; 由原则 2, $B_{k \times k} = \text{diag}(b_1, b_2, \dots, b_k)$ 中, b_i 越大越好

则带入变换关系至协方差矩阵中:

$$B_{k \times k} = \frac{1}{m} Z Z^T = \frac{1}{m} (UX)(UX)^T = U_{k \times n} C_{n \times n} U_{n \times k}^T \quad (1)$$

其中， $C_{n \times n}$ 为原数据矩阵 X 的协方差矩阵。

而考虑到 $C_{n \times n}$ 的特征值分解形式为：

$$D_{n \times n} = Q_{n \times n} C_{n \times n} Q_{n \times n}^T \quad (2)$$

对比 (1) (2) 式可发现仅有维度不同。结合原则 2， $U_{k \times n}$ 只需取 $C_{n \times n}$ 的前 k 大的特征值对应的特征向量（的转置）。

联想到 EVD 和 SVD 的关系，设 X 的 SVD 为： $X_{n \times m} = U_{n \times n} \Sigma_{n \times m} V_{m \times m}^T$

则 $C_{n \times n} = X X^T = U_{n \times n} \Sigma^2 V_{m \times m}^T$ ，该式等价于 $C_{n \times n}$ 的特征值分解。由于 SVD 中奇异值为从大到小排列，仅需取 $U_{n \times n}$ 的前 k 列再转置，即为所需的 $U_{k \times n}$ 。

总结：用 PCA 降维，先对数据向量做 SVD，再取左奇异向量矩阵的前 k 列（的转置）作为降维矩阵即可。

3、实际意义理解

左奇异向量矩阵 $U_{n \times n}$ ，同时也是原数据协方差矩阵的特征向量矩阵。特征向量为一组正交向量，因此求得 $U_{n \times n}$ 即求原数据所构成向量空间的一组正交基。而此时数据降维的意义也很明显了， $z_i = (U_{n \times k})^T x_i = (< u_1, x_i >, < u_2, x_i >, \dots, < u_k, x_i >)^T$ ，即把 x_i 投影在 n 个正交基中的 k 个上。

而由于在此方法下，特征向量的顺序对应了特征值的从大到小排列，类比信号向量的某分量的功率与特征值平方的对应关系，此方法下的特征向量可解释为原数据的“主成分”。

4、本题应用

对于本题，任务四的实质是投影到 2 个正交基上=降维至 2 维；

任务三的实质则是投影到 r 个正交基后，再用这 r 个正交基按投影分量线性组合，以此近似原数据。

二、实验过程记录与分析

1. 任务一：在 matlab 中加载并画出数据集

- **任务要求：**熟悉 matlab 加载数据与绘图的函数，画出人脸数据集中的数据。

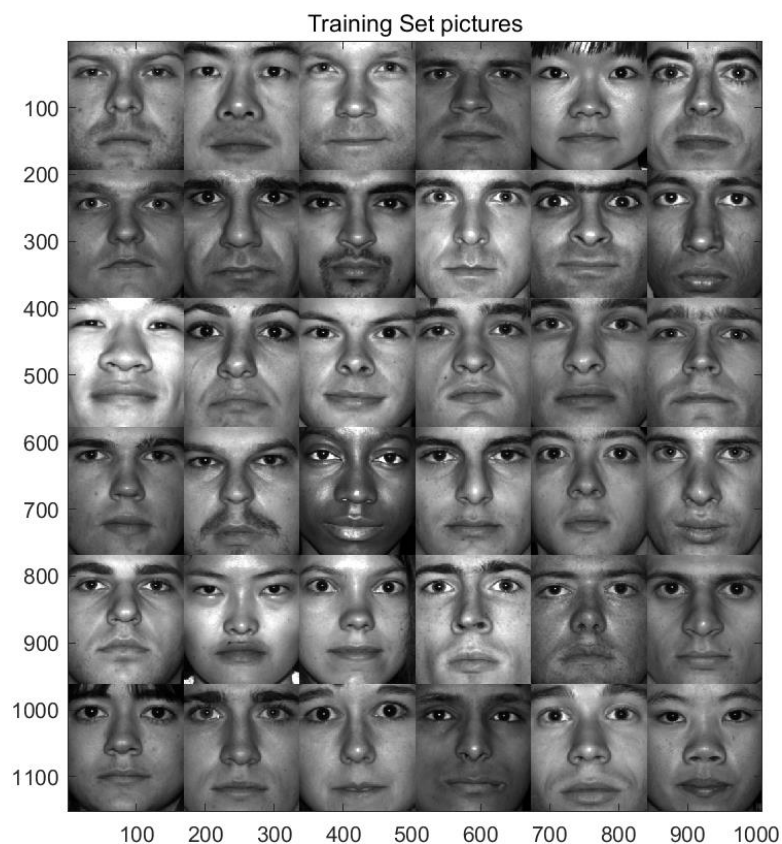
提示：数据集包含 38 个人的人脸，每张图片的大小为(192, 168)

- **原理分析：**略，见代码注释

- 代码实现：（可复制的代码文本见附录）

```
1      %% Task1 加载并画出数据集
2      %下面画出前36个人的第一张照片，并拼成一个6*6的图
3      load('YaleBExtend.mat');
4      TrainSetPic=zeros(192*6,168*6);%TrainSetPic是要画的6*6拼图
5      for i=1:6
6          for j=1:6
7              x_t=cell2mat(X(1,6*(i-1)+j));
8              %取元胞数组中第一项矩阵，即一个人的所有照片
9              x_t=reshape(x_t(:,1),192,168);
10             %取这个人的第一张照片，即第一个列向量，并矩阵化
11             TrainSetPic(192*(i-1)+1:192*i,168*(j-1)+1:168*j)=x_t;
12             %把第i行第j列的图拼到TrainSetPic上
13         end
14     end
15     figure('Name','Training Set pictures');
16     imagesc(TrainSetPic),colormap gray;
17     title('Training Set pictures');
18
```

- 运行结果：



2. 任务二：用 SVD 处理数据，计算特征脸

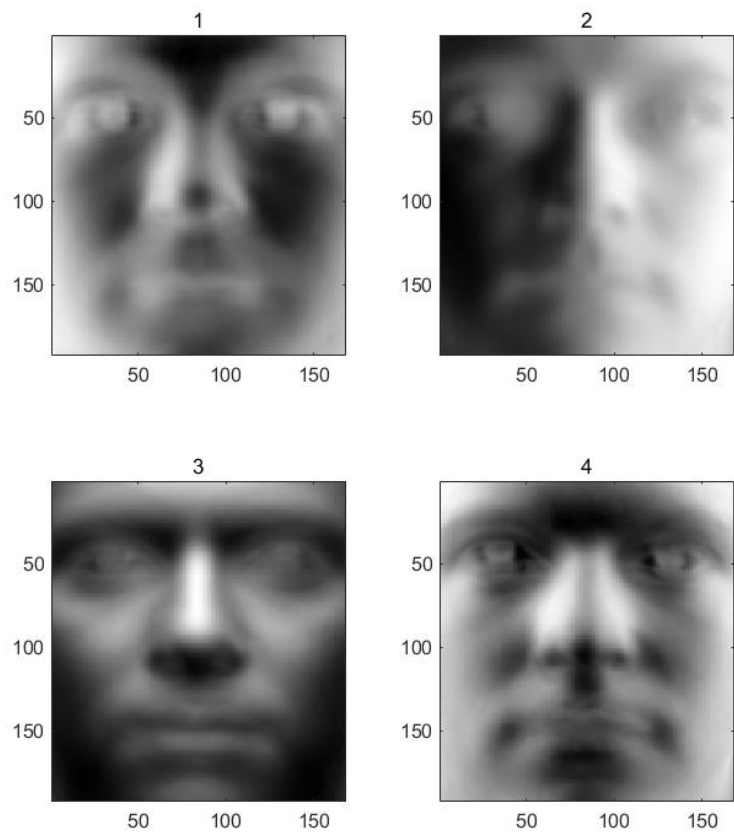
- **任务要求：**使用SVD 对数据进行分析，提取并可视化前 4 张特征脸（左奇异向量）。首先将人脸数据去均值，方便后面数据的分析。这里我们将数据集划分为包含前 36 个人的所有数据的训练集，和包含最后 2 个人所有数据的测试集。

再用SVD 对去均值后的人脸数据进行处理，画出特征脸。

- **原理分析：**实际上就是对汇总了所有训练集照片的 32256×2286 的矩阵进行奇异值分解。前 4 张特征脸即为前 4 个主成分。
- **代码实现：**（可复制的代码文本见附录）

```
19 %% Task2 用SVD处理数据，计算特征脸
20 %首先把36个人的共2286个列向量拼起来的训练集矩阵A
21 A=zeros(32256, 2286);
22 j=0;
23 for i=1:36
24     x_t=cell2mat(X(1, i));
25     size_t=size(x_t);
26     A(:, j+1:j+size_t(1, 2))=x_t;%把每个人的照片列向量依次拼到A上
27     j=j+size_t(1, 2);
28 end
29 Mean=mean(A, 2);%按列向量求均值
30 for i=1:2286
31     A(:, i)=A(:, i)-Mean;
32 end
33 [u, s, v]=svd(A, 'econ');%u是左奇异向量矩阵，v是右奇异向量矩阵，s是奇异值对角阵
34 %此处因为economic size, u为32256*2286, s为2286*2286
35 figure('Name', 'The first 4 singular face')
36 for i=1:4 %画出前四张特征脸
37     subplot(2, 2, i)
38     imagesc(reshape(u(:, i), 192, 168)), colormap gray;
39     title(i);
40 end
41
```

- **运行结果：**



3. 任务三：人脸近似

- **任务要求：**利用SVD 得到的奇异向量进行低秩矩阵近似(包括一张测试集中的人脸和一张你自己挑选的图片)，直观理解秩的大小对矩阵近似效果的影响。

人脸和一张你自己挑选的图片)，直观理解秩的大小对矩阵近似效果的影响。

利用投影 $\hat{x}_{test} = UU^T x_{test}$ ，用学习到的特征脸来顾及测试集中的数据。注意这里的测试图片不属于训练集

- **原理分析：** 投影到 r 个正交基后，再用这 r 个正交基按投影分量线性组合，以此近似原数据
- **代码实现：**（可复制的代码文本见附录）

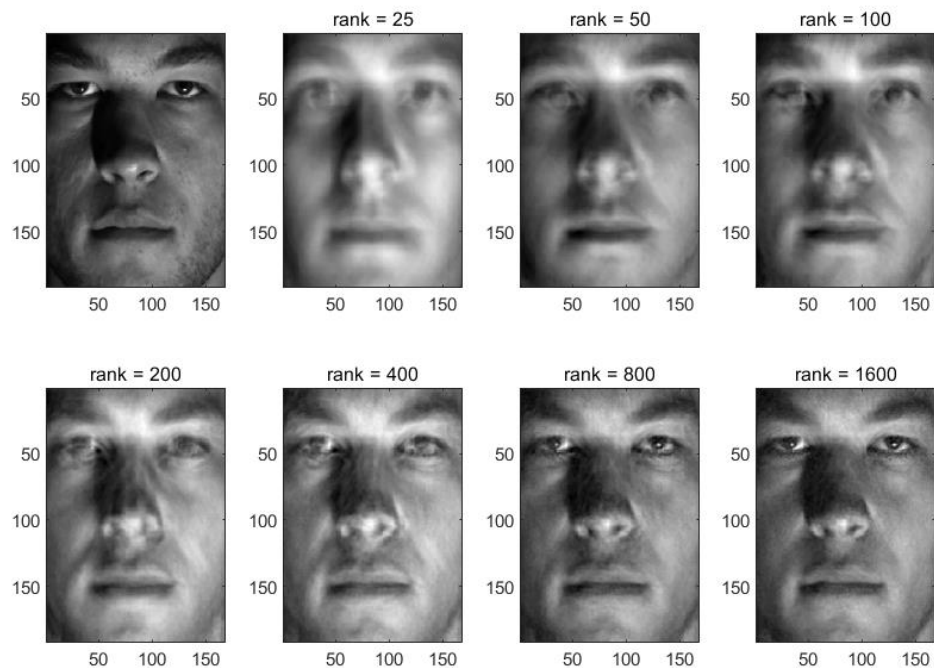
```
42 %% Task3 人脸近似
43 - x1=cell2mat(X(1,37));
44 - x1=x1(:,43);
45
46 - x2=imread('haha.jpg');
47 - x2=rgb2gray(x2);%原图像有3通道，给它灰度化
48 - x2=im2double(x2);%将图象数组转换成double精度类型
49 - x2=reshape(x2,32256,1);%列向量化
50
```

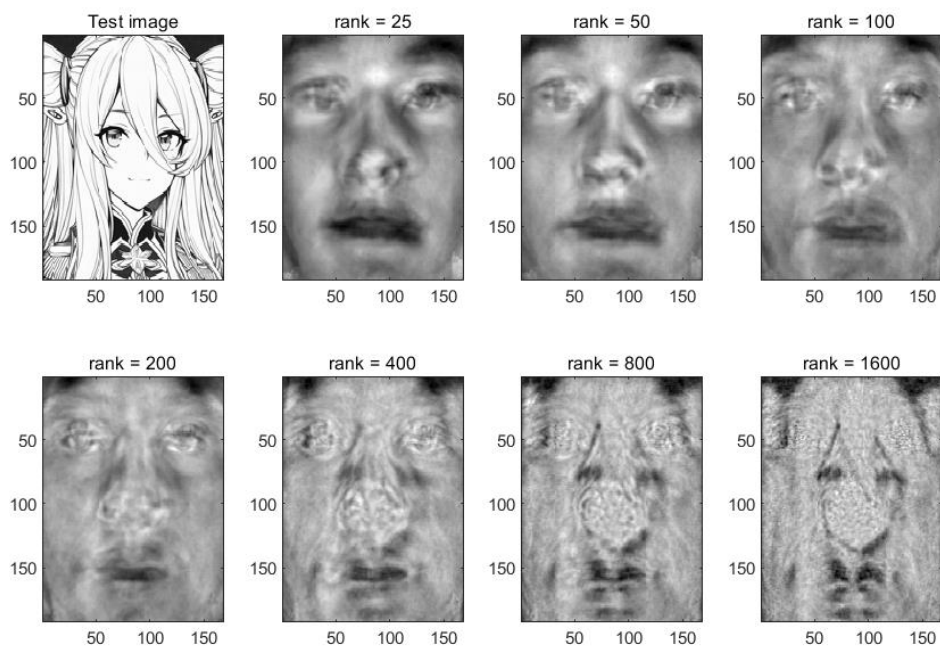
```

51
52 — r=[25, 50, 100, 200, 400, 800, 1600];
53 — figure('Name','Test image 1');
54 — subplot(2,4,1);
55 — title('Test image');
56 — imagesc(reshape(x1,192,168)), colormap gray;
57 — for i=1:7
58 —     u_t=u(:,1:r(1,i));
59 —     x_appr=u_t*u_t.'*x1;
60 —     subplot(2,4,i+1);
61 —     imagesc(reshape(x_appr,192,168)), colormap gray;
62 —     title(['rank = ',int2str(r(i))]);
63 — end
64
65 — figure('Name','Test image 2');
66 — subplot(2,4,1);
67 — imagesc(reshape(x2,192,168)), colormap gray;
68 — title('Test image');
69 — for i=1:7
70 —     u_t=u(:,1:r(1,i));
71 —     x_appr=u_t*u_t.'*x2;
72 —     subplot(2,4,i+1);
73 —     imagesc(reshape(x_appr,192,168)), colormap gray;
74 —     title(['rank = ',int2str(r(i))]);
75 — end

```

● 运行结果:





可见随着秩的增加,近似效果逐步改善。可见对第 37 人的照片近似效果较好,而对于动漫风格的图像近似效果较差,推测原因是图像与训练集数据特征偏离较远。从矩阵的角度理解, 192×168 的灰度图像有 32256 维的自由度,但实际上只有 2286 个基向量,这 2286 个基向量能完全表达 36 个人的照片所构成的实向量空间(是 32256 维实向量空间的子空间),即训练空间。而在此之外,在 32256 维实向量空间中,与训练集空间相差越远的数据,所能近似的效果就越差。

4. 任务三：人脸识别及分类预处理

- **任务要求：**将两个不同的人的脸投影到特征空间并可视化(可以参考下面的示例)，说明如何将 PCA 应用于人脸识别或分类任务。尝试用特征脸来作为坐标系，构成一个特征脸空间。将不同的人脸投影到这个特征空间，观察它们的特点。
- **代码实现：**(可复制的代码文本见附录)

```

78  %% Task4 人脸识别及分类预处理
79  x_vec=u(:, 123);
80  y_vec=u(:, 33);
81  p1=cell2mat(X(1, 33));
82  p2=cell2mat(X(1, 7));
83  [~, L1]=size(p1);
84  [~, L2]=size(p2);
85  feature_1=zeros(2, L1);
86  feature_2=zeros(2, L2);
87  figure('Name', 'Feature Space Plot(person 123&33 -> singular vector 33&6)');

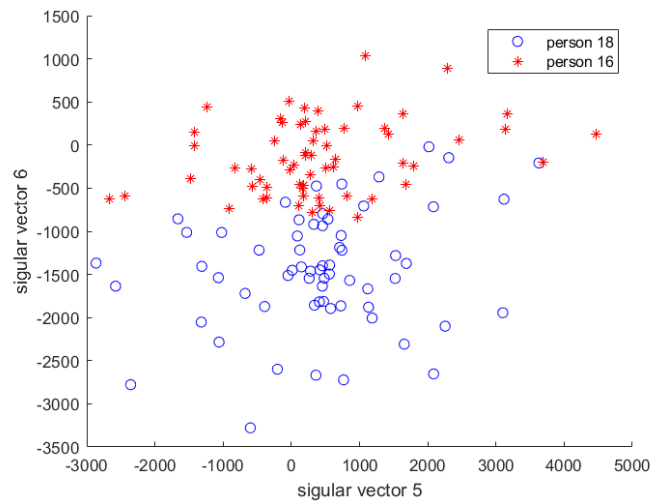
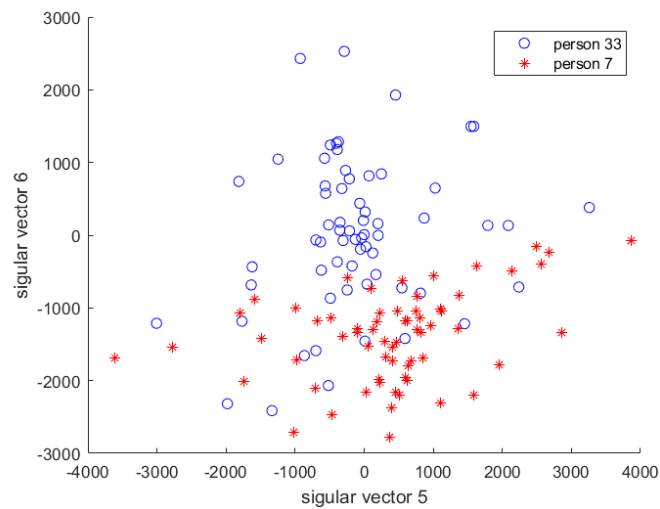
```

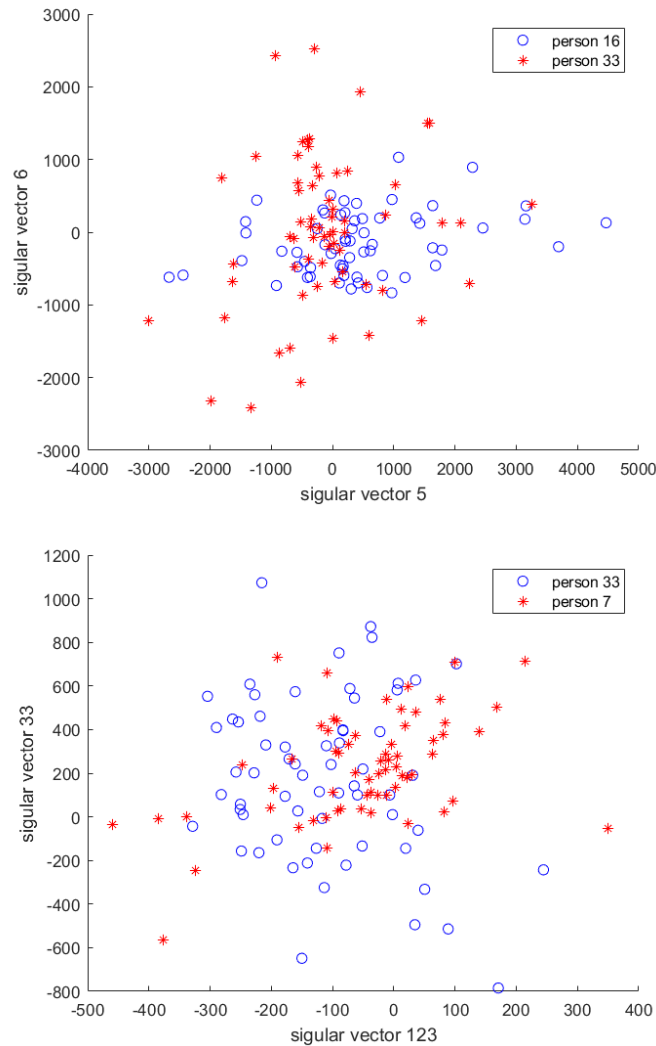
```

88 — for i=1:L1
89 —     feature_1(:,i)=[x_vec.'*p1(:,i),y_vec.'*p1(:,i)].';
90 — end
91 — scatter(feature_1(1,:),feature_1(2,:),[],'b','o');
92 — hold on;
93 — for i=1:L2
94 —     feature_2(:,i)=[x_vec.'*p2(:,i),y_vec.'*p2(:,i)].';
95 — end
96 — scatter(feature_2(1,:),feature_2(2,:),[],'r','*');
97 —
98 — xlabel('singular vector 123');
99 — ylabel('singular vector 33');
100 — legend('person 33','person 7')

```

● 运行结果:





前两张图可以看出，经过投影的数据点有较好的线性可分性，但后两张图展示了投影的数据点有较多重叠，并不能显著地判断出不同的人。可见随机选取 2286 维中的两个维度进行分类，并不能保证这两个维度仍然保留了足以用以分类的特征信息，但较为直观。

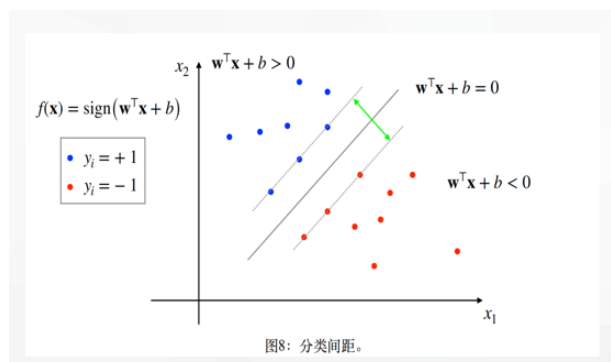
作为人脸识别和数据分类的预处理，下一步可采用 SVM(支持向量机) 进行二分类。SVM 算法的原理仅简述如下，即在 n 维空间中找一个(超)平面作为分类平面

$$\omega^T x + b = 0$$

使得来源于两个人的数据

分布在平面两侧。而找这个平面的方法是从支持向量出发，即离分类平面最近的样本点，这些样本点构成两个支持平面，支持平面到分类平面的距离称为分类间距 ρ ，而 SVM 的核心思想就是最大化分类间距 ρ 。

使用数学语言可以描述为以下优化问题：



目标函数 $\text{Margin}(\omega, b) = \min_{\omega, b, x_i} \text{distance}(\omega, b, x_i), i = 1, \dots, N$

$$= \min_{\omega, b, x_i} \frac{1}{\|\omega\|} |\omega^T x_i + b|, i = 1, \dots, N$$

约束条件（实际是描述目标函数到分类结果 $y_i \in \{+1, -1\}$ 的映射）:

$$y_i(\omega^T x_i + b) > 0$$

最优化问题公式变为:
$$\begin{cases} \max_{\omega, b} \min_{x_i} \frac{1}{\|\omega\|} |\omega^T x_i + b| \\ \text{s.t. } y_i(\omega^T x_i + b) > 0 \end{cases}$$

上述最优化问题可用拉格朗日法化为无约束优化问题, 并使用 KKT 条件简化求解, 此为 Hard-Margin SVM 算法。

进一步地, 还可以引入 hinge loss 来允许一定的错误分类, 让分类平面并不作为一个绝对的界限, 以此应对两数据组有部分重叠的情况。还可以在二分类 SVM 的基础上构建多分类 SVM。

三、附加题

在前面的基础上, 结合试验结果观察分析 PCA 的特点。思考能否找到一种比 PCA 更适用于图像分类的降维方法, 并说明原因。

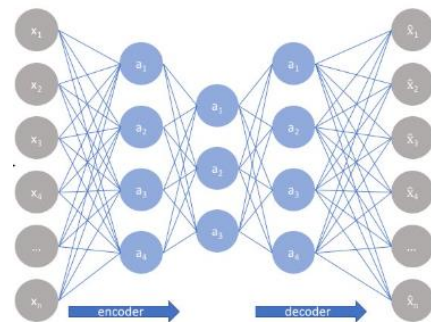
PCA 可以很好地去线性相关来降维, 但对于高阶相关性就没有办法了。对于线性可分性很差的数据 (在向量空间中, 两组数据较好的分界面可能是一个超曲面而非超平面), PCA 降维不合适。

- Kernel PCA 方法

KPCA 的大致思路是: 对于输入空间(Input space)中的矩阵 X , 我们先用一个非线性映射把 X 中的所有样本映射到一个高维的空间 (称为特征空间, Feature space), (使其线性可分), 然后在这个高维空间进行 PCA 降维。

- Auto-Encoder

自编码器通过神经网络实现, 从输入到节点最少的隐藏层部分的网络称为编码器, 从最小的隐藏层再到输出为解码器。训练的目标是, 使得编码器的输入和解码器的输出尽可能一致, 因此损失函数衡量输入和输出向量的差距。



当训练完成，输入输出一致时，此时最小的隐藏层则可以作为数据降维的结果。编码的过程即降维的过程，解码的过程即数据复原的过程。

由于神经网络层与层之间可采用多种非线性的激活函数，因此可以实现非线性降维。

四、代码附录

```
%% Task1 加载并画出数据集
%下面画出前 36 个人的第一张照片，并拼成一个 6*6 的图
load('YaleBExtend.mat');
TrainSetPic=zeros(192*6,168*6);%TrainSetPic 是要画的 6*6 拼图
for i=1:6
    for j=1:6
        x_t=cell2mat(X(1,6*(i-1)+j));%取元胞数组中第一项矩阵，即一个人的所有照片
        x_t=reshape(x_t(:,1),192,168);%取这个人的第一张照片，即第一个列向量，并矩阵化
        TrainSetPic(192*(i-1)+1:192*i,168*(j-1)+1:168*j)=x_t;%把第 i 行第 j 列的图拼到 TrainSetPic 上
    end
end
figure('Name','Training Set pictures');
imagesc(TrainSetPic),colormap gray;
title('Training Set pictures');

%% Task2 用 SVD 处理数据，计算特征脸
%首先把 36 个人的共 2286 个列向量拼起来的训练集矩阵 A
A=zeros(32256,2286);
j=0;
for i=1:36
    x_t=cell2mat(X(1,i));
    size_t=size(x_t);
    A(:,j+1:j+size_t(1,2))=x_t;%把每个人的照片列向量依次拼到 A 上
    j=j+size_t(1,2);
end
Mean=mean(A,2);%按列向量求均值
for i=1:2286
    A(:,i)=A(:,i)-Mean;
end
```

```

[u,s,v]=svd(A,'econ');%u 是左奇异向量矩阵, v 是右奇异向量矩阵, s 是奇异值对
角阵
%此处因为economic size, u 为32256*2286, s 为2286*2286
figure('Name','The first 4 singular face')
for i=1:4 %画出前四张特征脸
    subplot(2,2,i)
    imagesc(reshape(u(:,i),192,168)),colormap gray;
    title(i);
end

%% Task3 人脸近似
x1=cell2mat(X(1,37));
x1=x1(:,43);

x2=imread('haha.jpg');
x2=rgb2gray(x2);%原图像有3 通道, 给它灰度化
x2=im2double(x2);%将图索数组转换成 double 精度类型
x2=reshape(x2,32256,1);%列向量化

r=[25,50,100,200,400,800,1600];
figure('Name','Test image 1');
subplot(2,4,1);
title('Test image');
imagesc(reshape(x1,192,168)),colormap gray;
for i=1:7
    u_t=u(:,1:r(1,i));
    x_appr=u_t*u_t.'*x1;
    subplot(2,4,i+1);
    imagesc(reshape(x_appr,192,168)),colormap gray;
    title(['rank = ',int2str(r(i))]);
end

figure('Name','Test image 2');
subplot(2,4,1);
imagesc(reshape(x2,192,168)),colormap gray;
title('Test image');
for i=1:7
    u_t=u(:,1:r(1,i));
    x_appr=u_t*u_t.'*x2;
    subplot(2,4,i+1);
    imagesc(reshape(x_appr,192,168)),colormap gray;
    title(['rank = ',int2str(r(i))]);
end

```

```
% Task4 人脸识别及分类预处理
x_vec=u(:,123);
y_vec=u(:,33);
p1=cell2mat(X(1,33));
p2=cell2mat(X(1,7));
[~,L1]=size(p1);
[~,L2]=size(p2);
feature_1=zeros(2,L1);
feature_2=zeros(2,L2);
figure('Name','Feature Space Plot(person 123&33 -> singular vector
33&6)');
for i=1:L1
    feature_1(:,i)=[x_vec.*p1(:,i),y_vec.*p1(:,i)].';
end
scatter(feature_1(1,:),feature_1(2,:),[],'b','o');
hold on;
for i=1:L2
    feature_2(:,i)=[x_vec.*p2(:,i),y_vec.*p2(:,i)].';
end
scatter(feature_2(1,:),feature_2(2,:),[],'r','*');

xlabel('singular vector 123');
ylabel('singular vector 33');
legend('person 33','person 7')
```