山东大学

信息科学与工程学院

2022－2023 学年第一学期

实 验 报 告

课程名称：信息基础 II

专 业 班 级 崇新学堂

学 生 学 号

学 生 姓 名

课 程 报 告 主成分分析（PCA）

1. 作业要求

(1) 不区分花的类别，使用 PCA 得到 1st 主成分，以及相应的重构误差.(Reconstruction error)。

(2) 计算 Iris setosa 的 1st 主成分，以及相应的重构误差.(Reconstruction error)

2. 理论部分

2.1 数据向量格式说明

$$X=\begin{bmatrix} x_1^{(1)} & \cdots & x_1^{(m)} \\ \vdots & \vdots & \vdots \\ x_n^{(1)} & \cdots & x_n^{(m)} \end{bmatrix} \qquad U=\begin{bmatrix} u_1^{(1)} & \cdots & u_1^{(m)} \\ \vdots & \vdots & \vdots \\ u_n^{(1)} & \cdots & u_n^{(m)} \end{bmatrix} \qquad Z=\begin{bmatrix} u_1^{(1)} & \cdots & u_1^{(m)} \\ \vdots & \vdots & \vdots \\ u_k^{(1)} & \cdots & u_k^{(m)} \end{bmatrix}$$

X 是输入的数据，维度为 $n\times m$，其中每个样本的特征为 n 维，样本个数为 m。U 矩阵是协方差矩阵的特征向量，维度为 $n\times n$。Z 矩阵是经过 PCA 处理之后的数据，维度为 $k\times m$。

2.2 数据预处理

在进行 PCA 处理之前，需要对数据进行一定的预处理，包括以下两个过程：

（1）平移数据点，使得满足（1）中的条件：

$$\mu_j=\frac{1}{m}\sum_{i=i}^{n}x_j^{(i)}=0 \tag{1}$$

对于数据处理的方法为，求取所有样本每个特征的均值，令每个样本的对应特征减去其对应的均值，如下所示（2）（3）所示：

$$\mu_j=\frac{1}{m}\sum_{i=i}^{n}x_j^{(i)} \tag{2}$$

$$x_j^{(i)}=x_j^{(i)}-\mu_j \quad for\ all\ \ i \tag{3}$$

（2）特征缩放，求取每个特征的方差，之后将每个特征除以对应特征的标准差，对应的数学表达（4）：

$$x_j^{(i)}=\frac{x_j^{(i)}}{\sigma_j} \quad for\ all\ \ i \tag{4}$$

2.3 PCA 过程

2.3.1 PCA 降维

首先，利用预处理完毕的数据求取协方差矩阵 $n\times n$，维度为其矩阵化的求解方式如下：

$$\Sigma=\frac{1}{m}X\bullet X^T \tag{5}$$

在获得协方差之后，对协方差矩阵进行奇异值分析，求取协方差矩阵的特征向量矩阵 U，选取矩阵 U 的前 k 列（k 在题目要求中为 1，但在本次实验中会进行一定的探讨），组成一个矩阵 U_reduce，其维度为 $k\times n$，在获得矩阵 U_reduce 之后，利用如下表达即可完成 PCA 过程：

$$Z = U_{recuce}^{T} \cdot X \tag{6}$$

Z 矩阵为 PCA 处理之后的特征矩阵。

2.3.2 PCA 重构

PCA 重构的过程是与 PCA 相反的过程，在获得到新的特征矩阵 Z 之后，将其映射到原先的维度内，获得的矩阵为 X_{approx}，其代表着通过重构产生的点，重构坐标的计算，如（7）所示：

$$X_{approx} = U_{reduce} \cdot Z \tag{7}$$

获取到重构点之后，我们定义重构误差，重构误差代表重构点到原数据点之间的距离，其数学表达如（8）所示：

$$\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} - x_{approx}^{(i)} \|^2 \tag{8}$$

2.4 有关 K 的选取问题

在第 2.3 节中提到，选取 k 个特征值最大的特征向量组成矩阵进行 PCA 过程，本题中要求 k 为 1，但在作业中进行了一定的探索，涉及特征保留度的问题，首先我们从误差说起，PCA 处理后常常需要满足以下条件：

$$\frac{\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} - x_{approx}^{(i)} \|^2}{\frac{1}{m}\sum_{i=1}^{m} \| x^{(i)} \|^2} \le \alpha \tag{9}$$

其中 α 代表损失系数，即通过 PCA 处理之后，其重构之后的误差与原数据的比值需要小于某个系数，一般来说人们会将 α 取为 0.05，0.1，0.15 等，其代表了 PCA 处理之后的特征保留程度，而利用奇异值分解之后，我们可以获得矩阵 S ,矩阵 S 的维度为 $n \times n$，其表达形式如下：

$$S = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_n \end{bmatrix} \tag{10}$$

矩阵 S 的主对角线元素为协方差矩阵的特征值，而其余元素都为 0，那么（9）即可转化为（11）：

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \le 1 - \alpha \tag{11}$$

观察（9）的表达，分母为所有特征值求和，分子是前 k 特征值求和，这便是我们选取 k 的标准，根据认为的需求，确定 α ，之后利用 α 确定 k 的取值。

3. 代码实现

3.1 数据集导入函数

该函数导入了鸢尾花数据集，并且对数据集做了格式上的处理，将其转化为 array 格式，并且对特征矩阵进行了转置处理，返回值为数据特征向量以及标签值。

```
def DataSet():
    iris = datasets.load_iris()                    # 导入鸢尾花数据集
    data = iris.data
    label = iris.target
    data = np.array(data)                          # 转为为numpy格式
    data = data.T                                  # 对特征向量进行转置
    label = np.array(label)
    return data, label                             # 输出特征矩阵及其标签值
```

3.2 数据预处理函数

该函数实现数据的预处理，在函数中首先求取每个特征的均值，之后每个特征减去其均值，完成中心化的过程。再求取每个特征的方差，每个特征除以其标准差即可。

```
def Data_transform(data):
    m = len(data[0])                               # 定义样本数量
    n = len(data)                                  # 定义样本维度
    b = np.sum(data, axis=1, keepdims=True)        # 对m个样本求和
    b /= m                                         # 求取每个特征的均值
    for i in range(n):
        for j in range(m):
            data[i][j] -= b[i][0]                  # 将每个样本进行平移
    a = data ** 2
    sigma = np.sum(a, axis=1, keepdims=True)
    sigma /= m                                     # 求取方差
    sigma = sigma ** 0.5                           # 求取每个样本的标准差
    for i in range(n):
        for j in range(m):
            data[i][j] /= sigma[i][0]              # 完成数据标准化工作

    return data                                    # 返回标准化数据
```

3.3 协方差矩阵求取函数

该函数利用 2.3.1 中提到的方法，求取数据的协方差矩阵。

```
def Cov_matrix(data):
    m = len(data[0])                               # m为样本数量
    cov_matrix = np.dot(data, data.T)              # 求取协方差矩阵
    cov_matrix /= m
    return cov_matrix                              # 协方差矩阵
```

3.4 特征向量矩阵及转换矩阵求取

在该函数中，利用 svd 方法完成了奇异值分解，获得了特征向量矩阵，并且通过截取前 k 列，获得了转换矩阵。

```
def Feature(cov_matrix):
    k = 2                                          # 设定k值
    u, s, v = np.linalg.svd(cov_matrix)            # 奇异值分解
    u_reduce = u.T[:k]                             # 求取转换矩阵
    return u_reduce, s
```

3.5 PCA 函数

在该函数中利用在第 2 节中提出的公式完成代码实现即可，返回映射后特征向量及重构

之后的特征向量。

```
def PCA(data, u_reduce):
    z = np.dot(u_reduce, data)
    x_approx = np.dot(u_reduce.T, z)
    return z, x_approx
```

3.6 K 值选择函数（非作业要求）

该函数主要实现的是 k 的选取，分母为所有特征值的求和值，分子为前 k 个特征值的求和，找到满足约束条件的最小的 k 值，作为最终选择的 k 值。

```
def Select_k(s):
    s_lamda = 0                                    # 定义表达式分母
    n_lamda = 0                                    # 定义表达式分子
    a = 0.05
    k = 0
    for i in range(len(s)):
        s_lamda += s[i][i]                         # 求取所有特征值的和
    for i in range(len(s)):                        # 求取前k个特征值的和
        n_lamda += s[i][i]
        if n_lamda/s_lamda > 1 - a:
            k = i
            break
    return k
```

4. 结果分析

4.1 作业结果

（1）不区分花的类别，使用 PCA 得到 1st 主成分，以及相应的重构误差.(Reconstruction error)。

首先展示每个特征都除以标准差的结果：

```
第 1 主成分向量为 [-0.52106591  0.26934744 -0.5804131  -0.56485654]
重构误差为 40.5563318800502
特征保有率为 0.7296244541329989
```

以下为只进行中心化，而不除以标准差的结果：

```
第 1 主成分向量为 [-0.36138659  0.08452251 -0.85667061 -0.3582892 ]
重构误差为 12.840646450201334
特征保有率为 0.9246187232017269
```

（2）计算 Iris setosa 的 1st 主成分，以及相应的重构误差.(Reconstruction error)

我们依然率先展示除以标准差的结果：

```
第 1 主成分向量为 [-0.6044164  -0.57561937 -0.37543478 -0.40297876]
重构误差为 24.268247311147356
特征保有率为 0.5146350537770531
```

以下为不除以标准差的结果：

```
第 1 主成分向量为 [-0.6690784  -0.73414783 -0.0965439  -0.06356359]
重构误差为 0.8911677965883511
特征保有率为 0.7647237023065536
```

通过对比可以发现，不除以标准差，重构误差更小，特征保有率更高，因此我们在此作业中并不需要对每个特征做除以标准差的操作，在后续所有讨论中，我们将不在对数据除以标准差。以上为作业要求的所有结果，接下来我们将进行一定的分析。

4.2 探索分析

4.2.1 修改 k 带来的影响

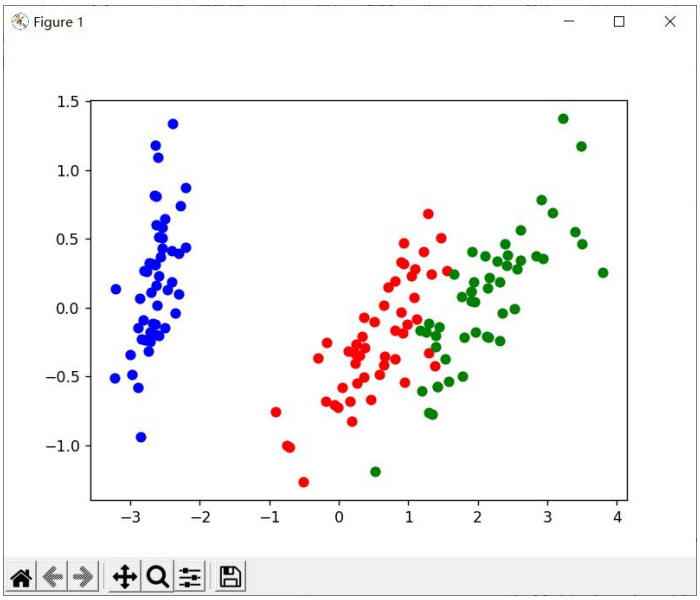
（1） 针对所有鸢尾花数据

```
第 1 主成分向量为 [-0.36138659  0.08452251 -0.85667061 -0.3582892 ]
重构误差为 12.840646450201334
特征保有率为 0.9246187232017269
```

以上为 k =1 的情况下，所有鸢尾花数据进行 PCA 处理之后的一些参数，观察到重构误差较大，特征保有率有 92.46%，接下来我们展示 k=2 的情况：

```
第 1 主成分向量为 [-0.36138659  0.08452251 -0.85667061 -0.3582892 ]
第 2 主成分向量为 [-0.65658877 -0.73016143  0.17337266  0.07548102]
重构误差为 3.801161089859738
特征保有率为 0.9776852063187949
```

当修改 k=2 之后，我们获得了两个主成分，重构误差有明显的减少，并且特征保有率也达到了 97.76%，满足一般的需求，我们将其分类可视化如下：



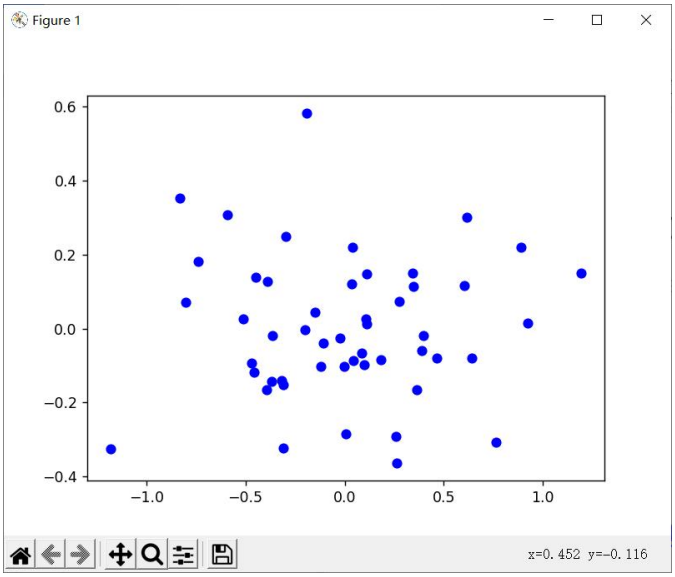
（2） 针对 setosa 数据

```
第 1 主成分向量为 [-0.6690784  -0.73414783 -0.0965439  -0.06356359]
重构误差为 0.8911677965883511
特征保有率为 0.7647237023065536
```

以上为 k =1 的情况下，所有鸢尾花数据进行 PCA 处理之后的一些参数，特征保有率仅有 76.47%，接下来我们展示 k=2 的情况：

```
第 1 主成分向量为 [-0.6690784  -0.73414783 -0.0965439  -0.06356359]
第 2 主成分向量为 [ 0.59788401 -0.62067342  0.49005559  0.13093791]
重构误差为 0.43891332494999696
特征保有率为 0.884122942393242
```

修改 k=2,获得了两个主成分，其重构误差进一步减少，特征保有率提高到了 88.41%，满足一般任务的需求，其可视化分布如下图：



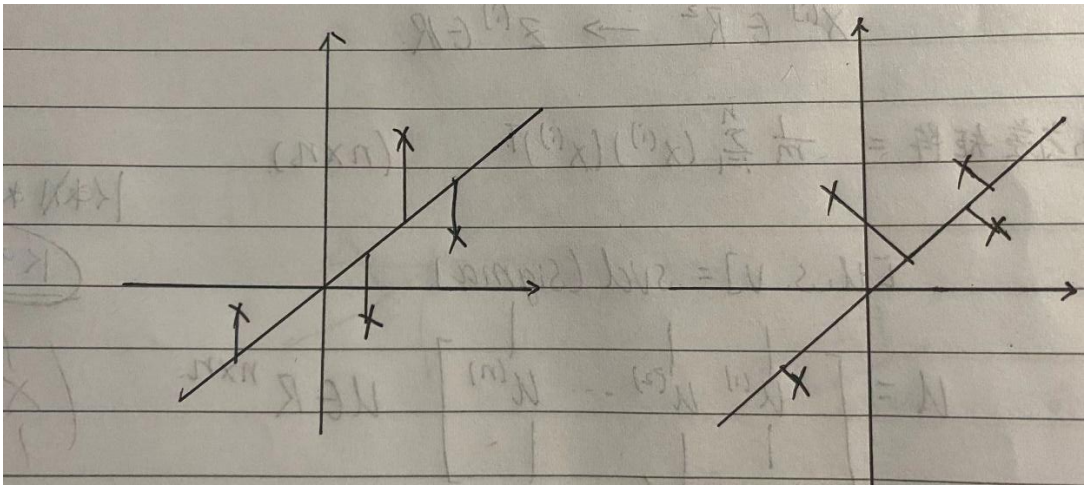
由此我们可以总结出：

（1）当 k<n 时，随着 k 的增大，也就是说降低维度的减少，特征的保有率越高且重构误差越小，这是符合我们的认知的.

（2）当 PCA 用于多个不同类别的数据时，其重构误差相较于对于单一类别的数据进行分类时更大。

（3）对于数据预处理中是否需要对数据特征除以标准差，需要根据具体的表现进行决定，在本次作业中，经过实验，最终决定不需要除以标准差。

（4）PCA 与线性回归的区别：以二维为例，线性回归中计算的距离是沿着 y 轴的距离，而 PCA 中计算的距离是样本点与投影点之间的距离，这是二者之间细微的不同，以下为直观的示意：（左侧为线性回归，右侧为 PCA）



5. 实验总结

在本次实验中 PCA 的实现过程并不困难，主要困难的在于数学推导，但不要求全部理解，在实现过程中，发现了诸多的规律，当我按照周老师上课讲的数据预处理方法处理之后，获得了实验的结果，我又将除去标准差的过程省略之后，又跑了一次代码，发现不除以标准差的实验结果更加优秀，因此在后续的探索实验当中，我暂时的省略了除去标准差的过程，后续的一些过程进展比较顺利。