机器学习课程实验报告

PCA 模型

学　　号　1180301007

姓　　名　赵锦涛

实验时间　2020 年 10 月

一、 实验目的:

实现一个 PCA 模型，能够对给定数据进行降维（即找到其中的主成分）。

二、 实验要求:

（1）首先人工生成一些数据（如三维数据），让它们主要分布在低维空间中，如首先让某个维度的方差远小于其它唯独，然后对这些数据旋转。生成这些数据后，用你的 PCA 方法进行主成分提取。

（2）找一个人脸数据（小点样本量），用你实现 PCA 方法对该数据降维，找出一些主成分，然后用这些主成分对每一副人脸图像进行重建，比较一些它们与原图像有多大差别（用信噪比衡量）。

三、 实验环境:

Python 3.8, Windows 10

四、 实验原理:

PCA 算法有两种形式：最大方差形式和最小误差形式。给定一组数据 x_n，样本集合的均值表达为：

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

投影数据的方差为

$$\frac{1}{N}\sum_{n=1}^{N}\{\boldsymbol{u}_1^T\boldsymbol{x}_n - \boldsymbol{u}_1^T\overline{\boldsymbol{x}}\}^2 = \boldsymbol{u}_1^T\boldsymbol{S}\boldsymbol{u}_1$$

考虑 M 维投影空间的一般情形，那么最大化投影数据方差的最优线性投影由数据协方差矩阵 S 的 M 个特征向量 $u_1, ..., u_m$ 定义，对应于 M 个最大的特征值 $\lambda_1, ..., \lambda_m$.

对于最小误差形式，不失一般性，M 维线性空间可以前 M 个基向量表，因此我们可以下式来近似每个数据点 x_n

$$J = \frac{1}{N}\sum_{n=1}^{N}\|\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n\|^2$$

进一步，我们可以得到失真度量的对应的值为：

$$J = \sum_{i=M+1}^{D} \lambda_i$$

于是，我们可以通过将这些特征向量选择成 D-M 个最小的特征值对应的特征向量，来得到 J 的最小值，因此定义了主子空间的特征向量是对应于 M 个最大特征值的特征向量。

PCA 算法可分为以下几个步骤：

- Step 1: 求平均值以及做 normalization
- Step 2: 求协方差矩阵 (Covariance Matrix)，在实验中使用的是散度矩阵 (Scatter Matrix)
- Step 3: 求协方差矩阵的特征根和特征向量
- Step 4: 选择主要成分
- Step 5: 转化得到降维的数据

五、 代码实现

本次实验共有 3 个文件，其名称和作用分别为：

- *datagen.py*　生成训练数据
- *pca.py*　PCA 算法实现
- *cv_test.py*　使用 PCA 算法对图片进行降维

其中 PCA 算法的实现为

```
def pca(x, k):
    x_mean = np.mean(x)
    x_norm = x - x_mean
    s_Cov = np.dot(np.transpose(x_norm), x_norm)
    eig_val, eig_vec = np.linalg.eig(s_Cov)
    index = np.argsort(-eig_val)
    index = index[0:k]
    pc = eig_vec[:, index]
    new_data = np.dot(np.dot(x - x_mean, pc), pc.T) + x_mean
    return new_data, pc, x_mean
```

代码 1: PCA

六、 实验结果与分析

使用 PCA 对满足二维高斯分布的点进行降维，结果如图所示：

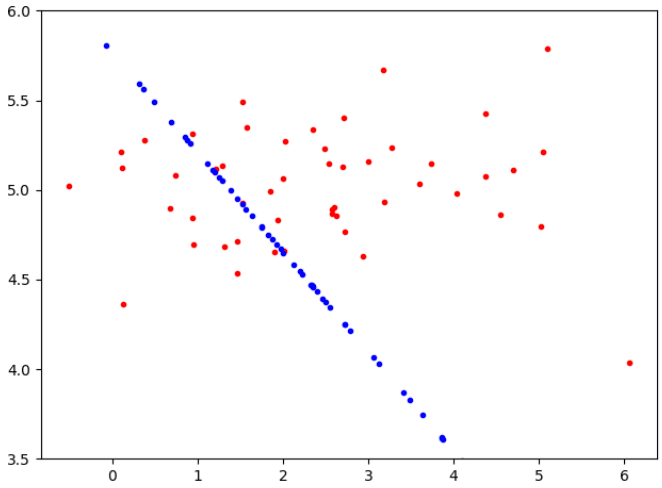

图 1: 二维数据降维

对三维数据提取主成分，结果如图所示：

```
[[ 0.89652117 -0.15263941 -0.41587378]
 [ 0.19723761  0.97811845  0.0661938 ]
 [-0.39667003  0.14137009 -0.90701013]]
```

图 2: 三维数据主成分

使用 PCA 对图像进行降维有 (原图像为 250×250，转换为灰度图像降到 80 维)：

图 3: 降维图片第一部分

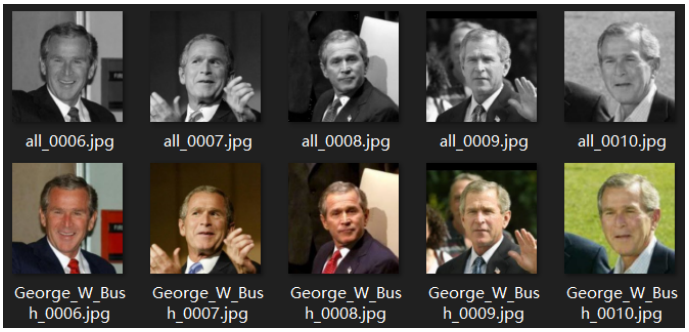

图 4: 降维图片第二部分

计算得到的信噪比为：

```
PSNR of picture 1: 52.34190558519863
PSNR of picture 2: 51.47968623792859
PSNR of picture 3: 51.28740025488449
PSNR of picture 4: 50.670789960124615
PSNR of picture 5: 52.45429189777967
PSNR of picture 6: 50.778059061042086
PSNR of picture 7: 51.998002620657275
PSNR of picture 8: 51.89669532628496
PSNR of picture 9: 52.18722345459852
PSNR of picture 10: 50.37424410794149
```

图 5: 信噪比

从实验结果可以看出，PCA 算法可以很好地保留原有数据的主要成分。

参考文献

[1] Bishop C M. Pattern recognition and machine learning[M]. springer, 2006.