计算机科学与技术学院实验报告

课程名称：机器学习

课程类型：必修

实验题目：PCA

学号：1171800323

姓名： 杨富祥

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、实验要求及实验环境

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维度，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）。

实验环境： Windows10 + matlabR2018a

三、设计思想

1．算法原理

整体思路： 给定D维数据集：$\{x_1, x_2, \dots, x_N\}$，要寻找$M(M < D)$个主成分，只需取数据的协方差矩阵S的前M个特征值最大的特征向量：$\{u_1, u_2, \dots, u_M\}$.

将数据集从D维空间投影到M维空间中：

$$\begin{bmatrix} \mathbf{u}_1^\top \\ \cdots \\ \mathbf{u}_M^\top \end{bmatrix} [\mathbf{x}_1 \cdots \mathbf{x}_N] = [\mathbf{z}_1 \cdots \mathbf{z}_N]$$

最大化方差： 考虑降到$M = 1$维空间的情况，定义这个空间的方向为$u_1 \in R^{D\times 1}$，并且$u_1^T u_1 = 1$. 我们希望的是让投影到这个M维空间的数据的分布尽可能分散，方差最大，从而有：

$$\frac{1}{N}\sum_{n=1}^{N}\{u_1^T x_n - u_1^T \bar{x}\}^2$$

$$= \frac{1}{N}\sum_{n=1}^{N}(u_1^T x_n - u_1^T \bar{x})(u_1^T x_n - u_1^T \bar{x})^T$$

$$= \frac{1}{N}\sum_{n=1}^{N}(u_1^T x_n - u_1^T \bar{x})\,(x_n^T u_1 - \bar{x}^T u_1)$$

$$= \frac{1}{N}\sum_{n=1}^{N}u_1^T (x_n - \bar{x})\,(x_n^T - \bar{x}^T) u_1$$

$$= u_1^T S u_1$$

其中均值$\bar{x}$和数据协方差矩阵S为：

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

$$S = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})(x_n - \bar{x})^T$$

最大化投影数据的方差，并且$||u_1|| = 1$，可以得到优化目标：

$$\max u_1^T S u_1$$

$$s.t. u_1^T u_1 = 1$$

使用拉格朗日乘子法即为最大化：$u_1^T S u_1 + \lambda(1 - u_1^T u_1)$，对$u_1$求导并设为 0 可以得到：

$$S u_1 = \lambda_1 u_1$$

因此u_1必须是S的一个特征向量，上式左边同乘u_1^T：

$$u_1^T S u_1 = \lambda_1$$

最小化误差：引入 D 维单位正交基向量集合：$\{u_1, u_2, \ldots, u_D\}$，并且满足：

$$u_i^T u_j = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases}$$

每个数据点可以精确地表示为基向量的一个线性组合，即：

$$x = \sum_{n=1}^{D} \alpha_{ni} u_i, \quad \alpha_{ni} = x_n^T u_i.$$

我们的目标是通过投影到 M 维子空间（加上一些失真）来表示数据点，用基向量的前 M 个表示 M 维线性子空间：

$$\hat{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i.$$

其中z_{ni}取决于特定的数据点，而b_i是常数。

从而得到优化目标：$\min J$，其中J为

$$J = \frac{1}{N}\sum_{n=1}^{N} ||x_n - \hat{x}_n||^2.$$

对J分别关于z_{nj}和b_j求导并设为 0，可以得到：

$$z_{ni} = x_n^T u_j$$

$$b_j = \bar{x}^T u_j.$$

代入J可以得到：

$$J=\frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D}(x_n^T u_i-\bar{x}^T u_i)^2=\sum_{i=M+1}^{D}u_i^T S u_i.$$

而$Su_i=\lambda_i u_i$

所以$J=\sum_{i=M+1}^{D}\lambda_i$，最小化$J$即是选择$D-M$个最小特征值对应的特征向量，则主成分为前 M 个最大的特征值对应的特征向量。

2. 算法的实现

降维：

```
function [mu,Y,P]=PCA(X, d)

% 设有m条n维数据，则X应为n行m列
% d，为降维后的维度
% mu，为X的每一行求均值形成的n维列向量
% Y，是降维到d维后的数据
% P，d个主成分

[~,m] = size(X);
mu = mean(X,2);
if size(X,2) > 1
    X = X - repmat(mu,1,m);  % 零均值化
end
sigma = 1/m * (X * X');  % 协方差矩阵
[U,S,~] = svd(sigma);    % 特征值分解，U(:,i)为特征向量，S(i,i)为特征值
% disp(S);
% disp(U);
P = U(:,1:d);  % 取U前d个最大的特征值对应的特征向量（svd分解后U默认按特征值降序排列）
Y = P' * X;    % 降维到d维后的数据

J = 0;
for i=1:d
    J = J+S(i,i);
end
J = J / sum(sum(S)); % 最小化误差角度，此处J应该越大，1-J越小，重建后失真越小
disp(J);
```

如果要重建，根据上图的$Y=P'X$，只需$P*Y$，再加上均值向量mu即可，对人脸的图片降维与重建指令如图：

```
function [X,Y,P,Z,U]=command()

X=readPicture();
% 降维，原始256*256，形成15 * 256矩阵
[mu,Y,P]=PCA(X, 15);
% 重建，利用15*256矩阵重建
result = P*Y+repmat(mu,1,size(X,2));
imshow(result);

% 再次降维，15*256的矩阵，形成8*15矩阵
[nmu,Z,U]=PCA(Y', 8);
% 重建，利用8*15矩阵、第一次降维后的均值向量mu和主成分P
figure(2);
imshow(P*(U*Z + repmat(nmu,1,size(Y,1)))'+repmat(mu,1,size(X,2)));
```

由于图片为 256*256 的矩阵，如果将其构造为一个列向量，再降维，中间计

算会涉及一个 65536*65536 的大矩阵（占用内存 32GB），不可计算。故在此仅仅将这个 256*256 原始矩阵直接作为输入，降维形成 15*256 的矩阵，并进行重建。第二次降维形成了 8*15 的矩阵，并利用该矩阵重建。

另一种解决方法是将图片缩放，如下图，缩放到 64*64.

```
function psnr()

img=imread('2.tiff');
[h, w]=size(img);

% 将原始图片缩放到64*64
img=imresize(img,[floor(h/4) floor(w/4)]);
img = im2double(img);
[h, w]=size(img);
% 绘制缩放后的图片
imshow(img);
```

然后再构建一个列向量 4096*1，进行降维与重建，如下图，降到 1 维：

```
12
13 X = img(:);
14 % 降到一维
15 d = 1;
16 [~,Y,P]=PCA(X, d);
17 % 重建
18 result = P*Y;
19 imgn = zeros(size(img));
20 theEnd = 0;
21 for i = 1:size(img)
22     begin = theEnd+1;
23     theEnd = i * size(img,1);
24     imgn(:,i) = result(begin:theEnd,1);
25 end
26 % 绘制重建后的图片
27 figure(2);
28 imshow(imgn);
29
30 B=8;                    %编码一个像素用多少二进制位
31 MAX=2^B-1;              %图像有多少灰度级
32 MES=sum(sum((img-imgn).^2))/(h*w);     %均方差
33 disp(MES);
34 PSNR=20*(log10(MAX)-log10(sqrt(MES)));          %峰值信噪比
35 disp(PSNR)
```

上图 30-35 行计算峰值信噪比。峰值信噪比经常用作图像压缩等领域中信号重建质量的测量方法，它常简单地通过均方差（MSE）进行定义。两个$m \times n$单色图像I和K，如果一个为另外一个的噪声近似，那么它们的的均方差定义为：

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||I(i,j) - K(i,j)||^2$$

峰值信噪比定义为：

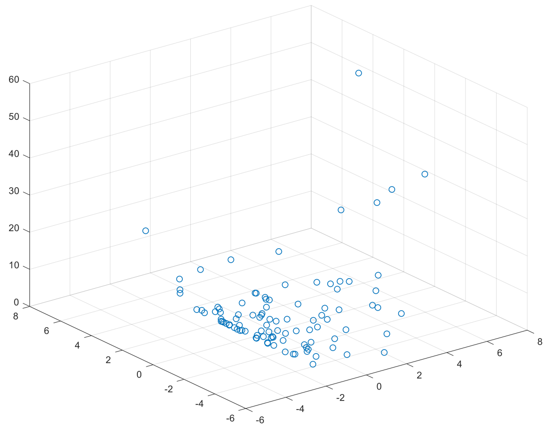
$$PSNR = 10 \cdot log_{10}\frac{MAX_I^2}{MSE} = 20 \cdot \log_{10}\frac{MAX_I}{\sqrt{MSE}}$$

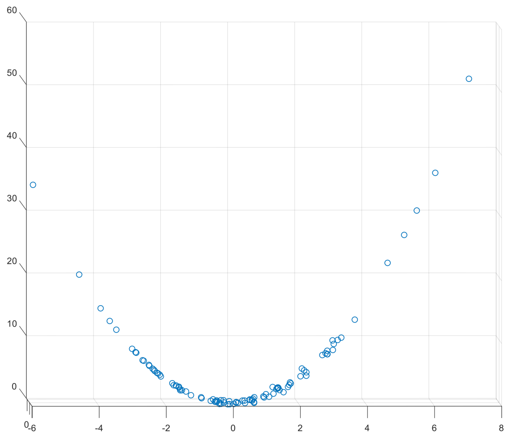
其中，MAX_I是表示图像点颜色的最大数值，如果每个采样点用 8 位表示，那么就是 255。

四、实验结果与分析

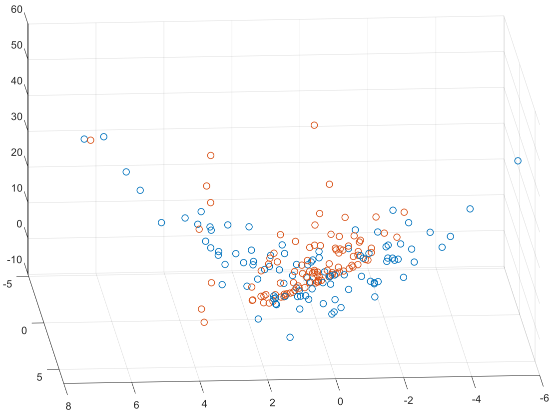
利用高斯分布生成二维数据，再变换得到三维数据 X.

```
function X = productData()

number=100;
mu=[0 0];
sigma=diag([4 5]);

X = mvnrnd(mu, sigma, number);
scatter3(X(:,1),X(:,2),X(:,1).^2);
X = [X X(:,1).^2];
X = X';
```





可以看出数据点分布在一个抛物面上



橙色为重建后的数据

人脸数据：

```
function X = readPicture()
I = imread('1.tiff');
I = im2double(I);
X=I;
```

```
function [X,Y,P,Z,U]=command()

X=readPicture();
% 降维，原始256*256，形成15 * 256矩阵
[mu,Y,P]=PCA(X, 15);
% 重建，利用15*256矩阵重建
result = P*Y+repmat(mu,1,size(X,2));
imshow(result);

% 再次降维，15*256的矩阵，形成8*15矩阵
[nmu,Z,U]=PCA(Y', 8);
% 重建，利用8*15矩阵、第一次降维后的均值向量mu和主成分P
figure(2);
imshow(P*(U*Z + repmat(nmu,1,size(Y,1)))'+repmat(mu,1,size(X,2)));
```

两次降维的$\frac{\sum_{i=1}^{d} s_{ii}}{\sum_{i=1}^{n} s_{ii}}$ (s_{ii} *为特征值*)分别为 0.9594 和 0.9657

原始图256*256　　根据15*256重建　　根据8*15重建

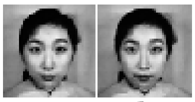
两次重建后的图片信噪比依次为：76.7925dB 和 74.3431dB

当 PSNR 值大于 30dB 的时候，可以认为去噪或压缩后的图像质量较好，低于 20dB 表示图像质量不可接受。

另外，如果将原始图片从 256*256 缩放到 64*64，作为本次实验的原始数据：

```
% 将原始图片缩放到64*64
img=imresize(img,[floor(h/4) floor(w/4)]);
img = im2double(img);
[h, w]=size(img);
% 绘制缩放后的图片
imshow(img);
```

计算需要构造一个 4096*1 列向量，会对 4096*4096 大矩阵求特征值（虽然可以计算，但是仍会占用较大内存，奇异值分解速度非常慢）。



a　　b

图 a 为缩放后的结果，图 b 为根据图 a 降维到 1 维并重建的结果，图 b 的信噪比是 362.7392dB

令人惊讶的是，从 4096 维降维到 1 维，重建后已经有非常大的信噪比。

略显异常的是，再增加低维空间的维数，信噪比也不会增加，甚至会降低（降维至 10 维重建，信噪比 358.9706dB，4000 维重建，信噪比 343.7154dB）。

观察奇异值分解得到的特征值，第一个特征值和其余特征值数量级相差在 10^{16} 以上，所以图 a 的主成分仅需取第一个特征向量，就能有很大的信噪比：

S ×

4096x4096 double

	1	2	3	4	5	
1	1.2659e+03	0	0	0	0	
2	0	3.5094e-13	0	0	0	
3	0	0	2.7042e-13	0	0	
4	0	0	0	1.2647e-13	0	
5	0	0	0	0	1.2647e-...	
6	0	0	0	0	0	1.2
7	0	0	0	0	0	
8	0	0	0	0	0	
9	0	0	0	0	0	
10	0	0	0	0	0	

由以上分析可知，如果降维到 1 维，就舍弃了 4095 个最小的特征值对应的特征向量，由于较小特征值对应的往往与噪声相关，通过 PCA 一定程度起到了降噪效果。

五、结论和思考

作为一个非监督学习的降维方法，它只需要特征值分解，就可以对数据进行压缩。PCA 将数据从 n 个特征降维到 d 个，用来进行数据压缩，例如 100 维的向

量最后可以用 10 维来表示，那么压缩率为 90%。

为避免损失信息过多，实际操作中选择主成分的数量 d，一般要满足下式：

$$\frac{\sum_{i=1}^{d} s_{ii}}{\sum_{i=1}^{n} s_{ii}} \geq \gamma(s_{ii}\text{为特征值，}\gamma \geq 95\%\text{重建效果比较好})$$

PCA 算法舍弃了$n-d$个最小的特征值对应的特征向量，由于较小特征值对应的往往与噪声相关，通过 PCA 一定程度起到了降噪效果。

PCA 算法仅仅需要以方差衡量信息量，不受数据集以外的因素影响。

PCA 各主成分之间正交，可消除原始数据成分间的相互影响的因素。

PCA 方差小的非主成分也可能含有对样本差异的重要信息，因降维丢弃可能对后续数据处理有影响。

六、参考文献

1. 周志华.(2016).机器学习.清华大学出版社.北京
2. 刘杨.(2019).机器学习课件 PCA.哈尔滨工业大学.哈尔滨
3. 吴恩达.网易云课堂机器学习课程章节 15
4. 博客：https://www.cnblogs.com/pinard/p/6239403.html
5. 博客：http://blog.codinglabs.org/articles/pca-tutorial.html
6. 博客：https://blog.csdn.net/weixin_42137289/article/details/86146535
7. 博客：https://blog.csdn.net/sac761/article/details/76547615
8. 博客：http://www.ai-start.com/ml2014/html/week8.html#header-n233

七、附录：源代码（带注释）

此处略，前文已经展示所有代码。