哈尔滨工业大学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： 选修

实验题目： PCA 模型试验

学号：1190201115
姓名： 陈宇豪

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

2.1 实验要求

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。

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）。

2.2 实验环境

Windows 10, matlab2016

三、设计思想（本程序中的用到的主要算法及数据结构）

降维的主要作用有两个，一是数据压缩，使用较少的计算机内存或磁盘空间，此外还能加速我们的学习算法。主成分分析(PCA)是最常见的降维算法。 在 PCA 中，我们要做的是找到一个方向向量，当把所有的数据都投射到该向量上时，希望投射平均均方误差能尽可能地小。具体实现方法如下：

数据产生算法

设计两个验证集，一是二维降一维，一个是三维降二维，保证数据中有一个维度的方差足够小就可以了

```
if dimension==2
    mean=[2,2];
    cov=[5,0;0,0.01];
    data=mvnrnd(mean, cov, num);
end
if dimension==3
    mean=[3,3,3];
    cov=[5,0,0;0,5,0;0,0,0.01];
    data=mvnrnd(mean, cov, num);
end
```

基于特征值分解协方差矩阵实现 PCA 算法

输入：数据集 X={x1,x2,x3...xn}，需要降到 k 维。

1）去平均值(即去中心化)，即每一位特征减去各自的平均值。、

```
mean=sum(data)/row;
centre_data=data-repmat(mean,row,1);
```

2）计算协方差矩阵,这里除或不除样本数量 n-1,对求出的特征向量没有影响。

```
covdata=(centre_data'*centre_data)./(row-1);
```

3）用特征值分解方法求协方差矩阵的特征值与特征向量。

```
[vector,value]=eig(covdata);
```

4）对特征值从大到小排序，选择其中最大的 k 个。然后将其对应的 k 个特征向量分别作为行向量组成特征向量矩阵 P。

```
vector = fliplr(vector);
if k>0
    vector=vector(:,1:k);
```

5）将数据转换到 k 个特征向量构建的新空间中。

```
score=centre_data*vector;
```

（6）最后还要将数据还原，其实进行第五步的逆运算，再加上第一步减去的平均值就行了。

```
tempdata=centre_data*vector;
score=tempdata*vector' +repmat(mean,row,1);
```

图像信噪比计算

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：

方差定义为：

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

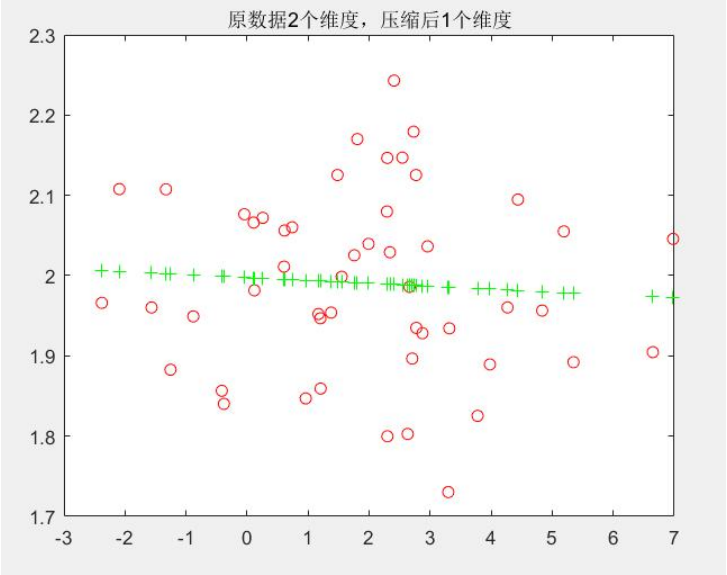
峰值信噪比定义为：

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

MAXI 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。

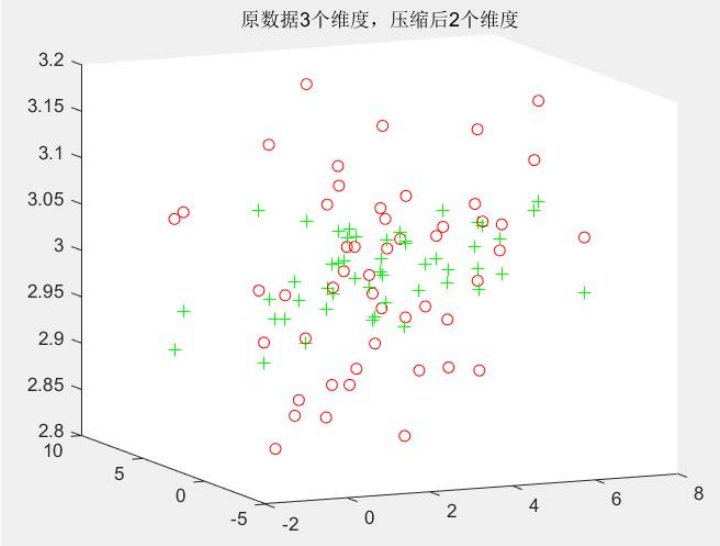
四、实验结果与分析

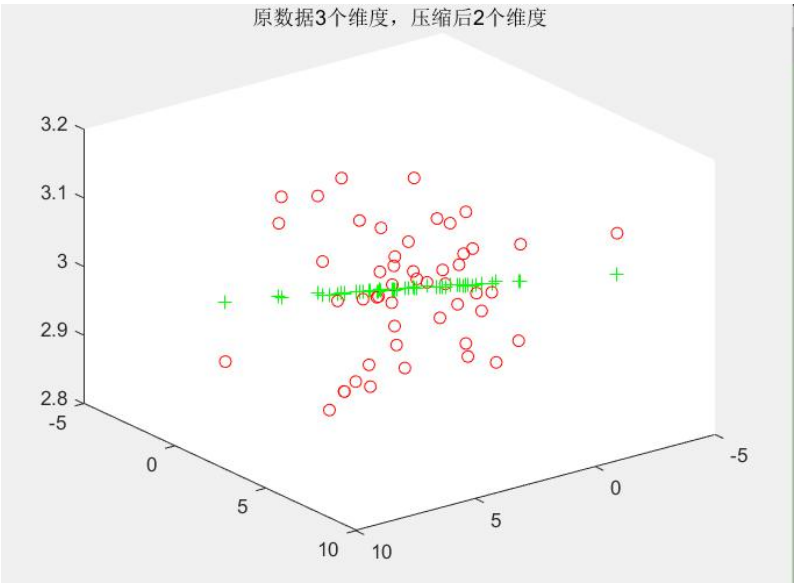
二维降到一维



三维降到二维

不同视角截图如下





可以看到三维数据中，纵轴的变化是很小的，也就是信息量较少的方向，再降维后每个点都投影在了近似垂直与纵轴的平面上。

人脸图像处理

采用的是 112*92 的灰度图


Figure 1
文件(F 编辑(E 查看(V 插入(I 工具(T 桌面(D 窗口(W 帮助(H
压缩后1个维度,信噪比为47.31

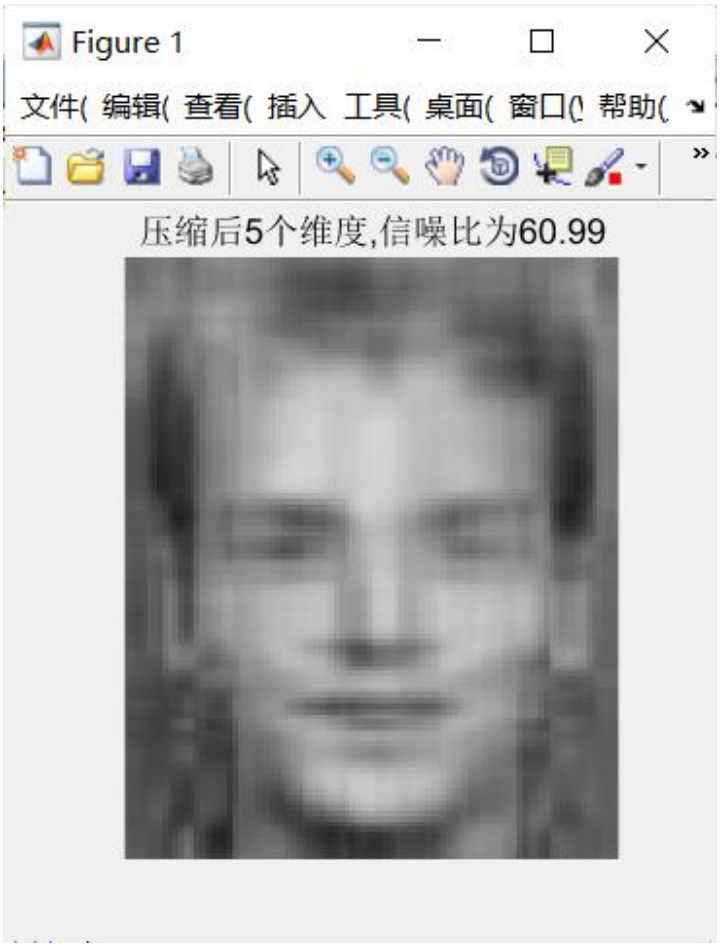


Figure 1
文件(编辑(查看(插入 工具(桌面(窗口(帮助(
压缩后5个维度,信噪比为60.99

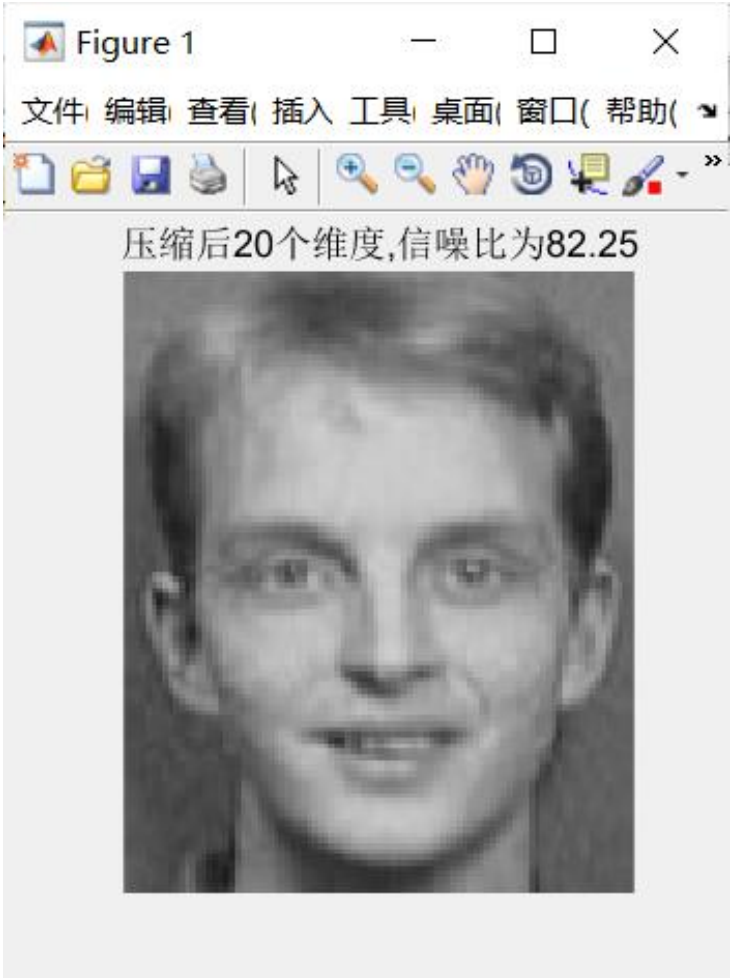
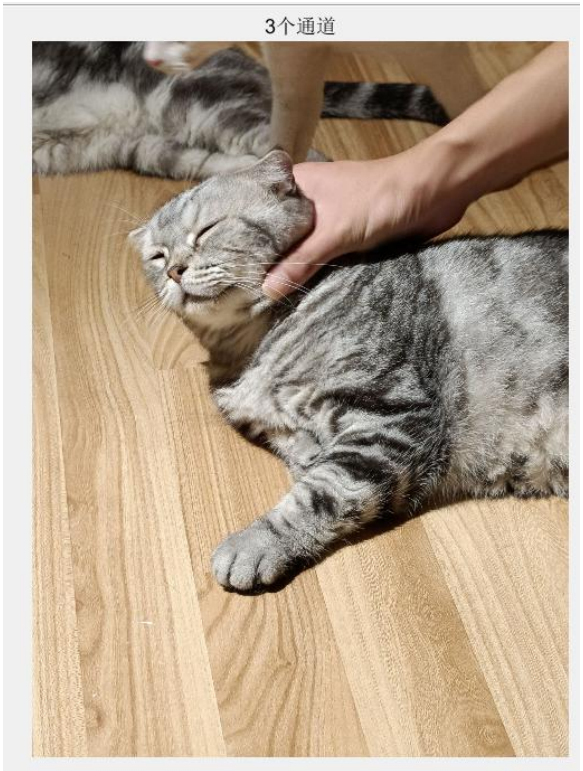


Figure 1
文件 编辑 查看 插入 工具 桌面 窗口 帮助
压缩后20个维度,信噪比为82.25

换一个角度考虑，当处理对象是彩色图象时，还可以将通道数视为维度，对其进行降维。原图如下，为三通道彩色图



降为二维的变化仍然不明显

降为 1 维后，就变成了灰度图。

五、结论

PCA 降维能够舍弃掉一些从总体上来说无关紧要的数据，降低数据存储和处理压力

六、参考文献

https://veal98.gitee.io/cs-wiki/#/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0/%E5%90%B4%E6%81%A9%E8%BE%BE/7-%E8%81%9A%E7%B1%BB+%E9%99%8D%E7%BB%B4

七、附录：源代码（带注释）

github 仓库地址：

https://github.com/1190201115/HIT-machineLearningLab/tree/main/lab4

Main 函数

```
data=create_data(2,50);
PCA(data,1);
```

create_data 函数

```
function [ data ] = create_data(dimension,num)
%产生数据
% 二维或三维
if dimension==2
    mean=[2,2];
    cov=[5,0;0,0.01];
    data=mvnrnd(mean, cov, num);
end
if dimension==3
    mean=[3,3,3];
    cov=[5,0,0;0,5,0;0,0,0.01];
    data=mvnrnd(mean, cov, num);
end
end
```

PCA 函数：

```
function [vector,centre_data,mean] = PCA( data,k )
%data 为数据，k 为降后的维数
[row,col]=size(data);
```

```
mean=sum(data)/row;
centre_data=data-repmat(mean,row,1);
covdata=(centre_data'*centre_data)./(row-1);
[vector,value]=eig(covdata)
vector = fliplr(vector)
if k>0
    vector=vector(:,1:k);
    draw(data,'or');
    tempdata=centre_data*vector;
    score=tempdata*vector'+repmat(mean,row,1);
    draw(score,'g+');
    str=sprintf('原数据%d%s，压缩后%d%s',col,'个维度',k,'
个维度');
    title(str);
end
```

photo 函数，用于图像处理

```
function [ output_args ] = photo( input_args )
%UNTITLED2 此处显示有关此函数的摘要
%   此处显示详细说明

n=1;
ph=imread('s40_2.bmp');
[row ,col ,bands]=size(ph);
ph=double(ph)/255;
[vector,centre_data,mean]=PCA(ph,0);
vector=vector(:,1:n);
new_ph=centre_data*vector;
pic=new_ph*vector'+repmat(mean,row,1);
imshow(pic);
noise=cal_noise(ph,pic);
figure(1);
imshow(pic,'InitialMagnification','fit');
str=sprintf('压缩后%d%s,信噪比为%.2f',n,'个维度',noise);
title(str);

%}
```

```
%%通道压缩
%{
n=1;
ph=imread('cat.png');
ph=double(ph)/255;
[row ,col ,bands]=size(ph);
mul = reshape(ph, [row*col,bands]);
[vector,centre_data,mean]=PCA(mul,0);
re = mul*vector(:,1:n)*vector(:,1:n)';
[r,c,bands] =size(ph);
comp = reshape(re,[r,c,bands]);
str =sprintf('%d%s,信噪比为%.2f',n,'个通道');
figure;imshow(comp);title(str);

%{
ph=im2double(ph);
vector=PCA(ph,80);
imshow(newph);
%}
%}
end
```

draw 函数，画出随机数据图和降维后的图像

```
function [ output_args ] = draw(data,type)
%绘图，对自己产生的数据
[row,col]=size(data);
for i=1:row
    if col==3
        plot3(data(i,1),data(i,2),data(i,3),type);
        hold on;
    end
    if col==2
        plot(data(i,1),data(i,2),type);
        hold on;
    end
    if col==1
        plot(data(i),type);
        hold on;
```

```
    end
end
```

cal_noise 函数，计算信噪比

```
function [ psnr ] = cal_noise(ph,pic)
%UNTITLED3 此处显示有关此函数的摘要
%   此处显示详细说明
[row,col]=size(ph);
mse=0;
for r=1:row
    for c=1:col
        mse=mse+(ph(r,c)-pic(r,c))^2;
    end
end
mse=mse/(row*col);
psnr=20*log(1/sqrt(mse))

end
```