实验报告——随机过程之人脸识别

无 58 马洋 2015011181 2017 年 10 月 25 日

摘要：本实验通过 PCA（Principal Component Analysis）实现训练样本和测试样本的降维，计算测试集数据的协方差矩阵得到任意两个像素之间的相关程度，并通过间接方法求解此高维矩阵的非零特征值及相应的特征向量，选取若干数值较大的特征值所对应的特征向量，然后分别将训练集数据与测试集数据用这些特征向量展开，通过比较两展开系数的相似程度来判断测试集数据是否与训练集数据是一类数据，即是否是人脸。

关键词：PCA，协方差，特征分解，人脸识别。

一、 实验目的

给定一组（35 幅人脸图像）训练样本，通过提取训练样本的特征判断测试样本是否是人脸。

二、 实验原理

基于 PCA（Principal Component Analysis）的人脸识别方法

1. 特征中心化，即每一维的数据都减去该维的均值，本实验中每个像素为一维。首先将每个测试样本化为列向量，再将各列向量排列成训练样本矩阵 X，特征中心化即：
$$\mathrm{X}(\mathrm{i},:) = \mathrm{X}(\mathrm{i},:) - \mathrm{mean}\big(\mathrm{X}(\mathrm{i},:)\big) \qquad i = 1,2,3 \dots M$$
其中 M 为每张训练图像像素个数，本实验中为 108*75=8100。

2. 计算协方差矩阵$\mathrm{R} = \frac{1}{N}\mathrm{X}X^T$，其中 N 为训练样本数量，本实验中为 35。本实验的核心目的是实现数据的降维，即将 8100 维的图像数据通过选择合适的基底降低维数。而如何判断基底选择的好坏呢？协方差矩阵。协方差矩阵衡量了测试集中任意两个像素之间的平均相关程度，如果一个数据集的协方差矩阵是对角化的，就意味着这个数据集更精简。于是我们想寻找一种变换 P，使得 PX 的协方差矩阵是对角化的，进而通过删减对角矩阵中数值较小的特征值降低 PX 的维度。通过线性代数知识易知，P 即为对 R 做特征分解后的特征向量矩阵 V 的转置（$P = V^T$）。

3. 特征分解：
$$\mathrm{RV} = \mathrm{VD}$$
其中 D 为特征值对角阵，$\mathrm{V} = [v_1 \ \ v_2 \dots v_M]$，$v_i$为特征值$\lambda_i$对应的特征向量。将特征值从大到小排列，并得到相对应的特征向量，根据需要选择特征值最大的 p 个特征向量，排列成新矩阵V_p 。

4. 将训练集和测试集降维：
$$\underset{p*35}{X_p} = \underset{p*8100}{V_p^T} * \underset{8100*35}{X}$$
对于第 i 列，有$V_p V_p^T \mathrm{X}(i) = V_p X_p(i)$，即$X_p$的一列代表了用挑选出的 p 个特征向量线性组合成$V_p V_p^T \mathrm{X}(i)$的各个系数，这意味着给定$V_p$后，通过上述映射，任何图像都可以用它投影到各个特征向量上的系数来大概描述，而这组系数的个数只有 p 个，相比原来的 N 个像素（本实验 N=8100），维度大为下降。

5. 人脸识别。经过上述步骤，我们已经得到了测试集和训练集各自的降维数据，本实验所采用的判定人脸方法为：对于给定的测试图像，降维后与训练集降维数据的平均值做内积，人为设定合适阈值，来判断给定图像是否为人脸。

高维矩阵特征分解

考虑到本实验中协方差矩阵$\mathrm{R}=\frac{1}{N}XX^T$为 8100*8100 的高维矩阵，直接做特征分解耗时长，

于是采用如下方法实现特征向量的计算：对低维矩阵$\mathrm{C}=\frac{1}{N}X^TX$（35*35）进行特征分解，得

到特征向量矩阵 E，XE各列为$\mathrm{R}=\frac{1}{N}XX^T$的特征向量，证明如下：

对于$\mathrm{C}=\frac{1}{N}X^TX$的某个特征向量$e_i$：　$\frac{1}{N}X^TXe_i=\lambda_ie_i$

左乘 X：　$\frac{1}{N}XX^TXe_i=\lambda_iXe_i$

令$v_i=Xe_i$：　$\frac{1}{N}XX^Tv_i=\lambda_iv_i$

即$v_i=Xe_i$为 R 的特征向量，对应的特征值为λ_i。又由线性代数相关结论，R 和 C 的非零特征值相同，于是通过这种方法可以计算出 R 的所有非零特征值及其所对应的特征向量，这也是我们所需要的特征向量。

三、　实现方法及成果

1. 训练过程

```matlab
% training
X_ori = zeros(108*75,35);   % training images(each column of X_ori)
for i=1:35
    filename = sprintf('%s%03d%s','imgs/clip_image',i,'.jpg');
    image = double(imread(filename));
    for x = 1:108
        for y = 1:75
            X_ori((x-1)*75+y,i) = image(x,y);
        end
    end
end
X = X_ori;
for i = 1:8100
    m = mean(X(i,:));
    X(i,:) = X(i,:) - m;
end
C = X'*X/35;
[E,D] = eig(C);
V = X*E;                % each column is an eigen vector
image_p = V'*X_ori;     % descending dimention images:
                        % projection of original images on V
averg = zeros(35,1);
for i = 1:35
    averg(i) = mean(image_p(i,:));
end
```

图 1 训练部分 matlab 代码

X_ori 为原始训练数据矩阵，每列为一幅人脸图像，训练数据集中所有图片大小相同，均为 108*75；X 为特征中心化后的矩阵；V 的每列为一个特征向量；本实验中所选 p=34（非零特征值个数为 34），averg 为训练图像用这 p 个特征向量所描述的系数向量的平均值，此即从训练集提取出的宝贵的人脸特征。

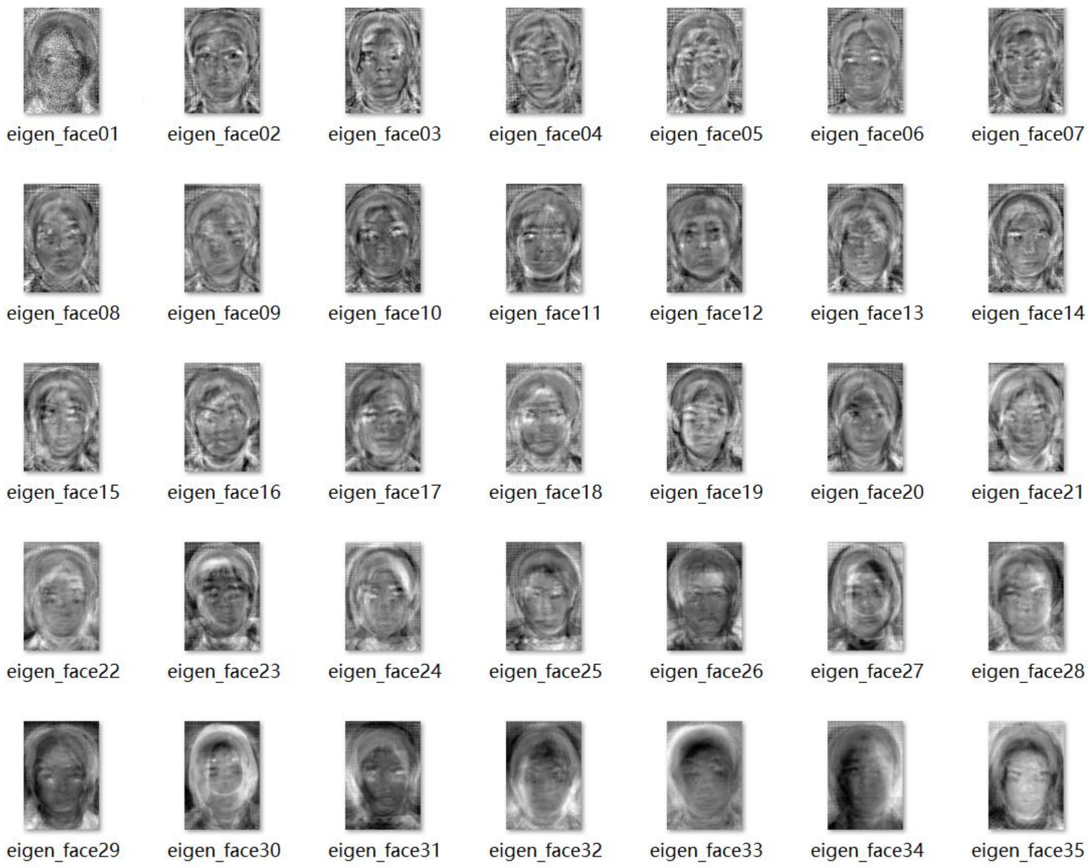

图 2 映射到[0,255]的 35 个特征向量（矩阵 V 的每一列，按特征值从小到大排列）

图 3 平均脸（由训练集平均系数 averg 确定）

2. 测试过程

```matlab
% testing
test_image = imread('horse.jpg');
test_vector = zeros(8100,1);
for x = 1:108
    for y = 1:75
         test_vector((x-1)*75+y) = test_image(x,y);
%          if x>105
%              test_vector((x-1)*75+y) = 0;
%          else
%              test_vector((x-1)*75+y) = test_image(x,y);
%          end
    end
end
test_p = V(:,35-p+1:35)'*test_vector;  % descending dimention test_image:
                                        % projection of test_image on V
sim = averg'*test_p;
if sim > 1.42e+16                     % choose a proper threshold
    fprintf('%s\n','判定结果：人脸');
else
    fprintf('%s\n','判定结果：非人脸');
end
```

图 4 测试部分 matlab 代码

首先将测试样本做基变换投影到各个特征向量上，得到相应的系数 test_p，随后将测试样本的系数 test_p 与训练样本的平均系数 averg 做内积，得到结果 sim，用以衡量二者的相关程度，sim 越大即测试样本与人脸样本相似程度越大，人为确定阈值 1.42e+16。

```
命令行窗口
判定结果：人脸
sim=14407394555075120.000000
fx >>
```

```
命令行窗口
判定结果：非人脸
sim=13996961940667164.000000
fx >>
```

图 5 测试结果

分别输入测试样本人脸和马脸，结果与预期相符。

四、　　实验总结

通过本次实验，我对协方差，基变换等线性代数的基础知识有了更深刻地认识，同时感受到了数学的美妙与强大，当实现想法的过程中遇到困难时，总是会发现数学早已给出了精确完美的解释。

在这次试验过程中，我感受到最为核心的一点就是要在数学推导的同时思考其物理意义，不但使推导更为顺畅，还能使思路更加丰富，有助于问题的解决。

五、　　参考文献

[1] PCA 的数学原理--张洋 http://blog.codinglabs.org/articles/pca-tutorial.html