**作业报告**

童逸轩

201830320294

（1）基本要求

Task #1: in pca.m

%% Task 1: Try to write this piece of code (the trick on page570 in Pattern.Recognition.and.Machine.Learning)

%task1

scatterMat=DATA'\*DATA;

[V,D]=eig(scatterMat);

trans=DATA\*V;

l2=(sum(trans.^2)).^0.5;

%a=l2./size(DATA,2);wi模为1，a模就是D模

%[~,IX] = sort(a,'descend');

[~,IX] = sort(diag(D),'descend');

projMat=trans./repmat(l2,size(DATA,1),1);

projMat=projMat(:,IX(1:para.pcaDim));

Task #2: in lda\_core.m

for i=1:length(subjectVec)

%% Task 2: Try to obtain Sw and Sb by yourself (on page62 of the Machine Learning textbook)

label=subjectVec(i);

idx=find(labelVec==label);

n\_sub=length(idx);

DATA\_sub=DATA(:,idx);

mf\_sub=mean(DATA\_sub,2);

Sb=Sb+n\_sub\*(mf\_sub-meanData)\*(mf\_sub-meanData)';

tmp\_mat=DATA\_sub-repmat(mf\_sub,1,n\_sub);

Sw=Sw+tmp\_mat\*tmp\_mat';

end

Task #3: 尝试基于pcaProj矩阵画出特征脸, in main.m

%% Task #3: try to draw eigenfaces by yourself

num\_eigenfaces\_show=9;

for i = 1:num\_eigenfaces\_show

subplot(3, 3, i)

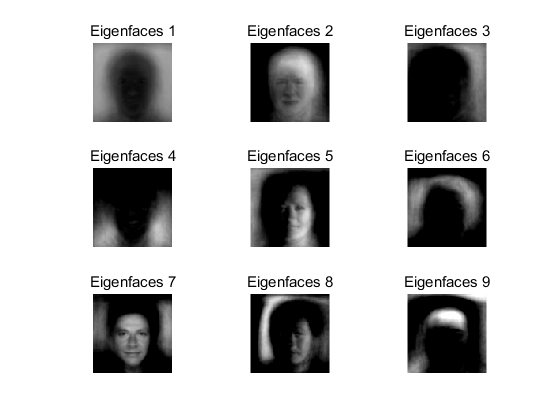
eigenface=40\*reshape(pcaProj(:,i),[100,100]);

imshow(eigenface);

title(['Eigenfaces ' num2str(i)]);

End

**Eigenfaces**



**Eigenfaces of fliped training set**



可以看到两者确有区别，通过左右翻转图像扩展训练集生成的eigenfaces明显更加对称。

**训练结果**

**参数调整和人脸识别率最佳参数threshold=300 pca.Dim=80**

**正确率：0.8455 改进后正确率：0.8545**

（2）扩展任务

**方法一：将训练集左右翻转，标签值不变，增加训练集。**

load TrainDATA.mat;

train2=fliplr(TrainData);

TrainData=cat(3,TrainData,train2);

trainLabelVec=repmat(trainLabelVec,2,1);

稍有效果准确率提升到0.8545，多分对一个样本。

有16个样本分错。

**方法二：尝试bagging的方法。**利用SVM和LDA两个分类器，将分类的结果组合起来达到更好的效果。（没有成功）

1. 利用matlab的fitcsvm和predict函数求出SVM分类和预测

调参

参数范围：

多项式核函数，阶次2,3,4。

高斯核函数，KernelScale：10-3-103

是否标准化

不适用核函数

调参结果：标准化，不使用核函数时，有最好的正确率0.8273

其他适用核函数情况，最高正确率均未超过0.7。其中适用高斯核，进行标准化且KernelScale=68时有适用核函数的最高正确率0.6273。

利用predict函数生成预测和得分。得分通过正负号判断正负样本。

2）将LDA的输出减去threshold，然后放缩

if accuracy02>accuracy01

prediction=prediction(2,:);

lda\_score=(output+threshold)./(threshold);

fore=-1;

else

prediction=prediction(1,:);

lda\_score=(output-threshold)./(threshold);

fore=1;

end;

目的是让其方差和SVM生成的得分的方差相近。

1. 最终得分以a\*（LDA得分）+b\*（SVM得分）加权求和的形式得到。

确定a=1，b通过调参得到其值为0.3。

组合后正确率为0.8455。有下降。

观察发现：

SVM出错序号

3 4 14 17 18 24 25 29 30 31 36 37 47 48 49 50 53 55 95

LDA出错序号

1. 25 29 30 36 47 48 49 68 71 78 82 91 92 95 110

LDA出错序号对应预测值

组合后出错序号

1. 24 25 29 30 31 36 47 48 49 53 71 78 82 92 95 110

相同错误

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 14 | 25 | 29 | 30 | 36 | 47 | 48 | 49 | 95 |
| LDA | -0.62 | -0.17 | -0.53 | -0.41 | -0.78 | -0.75 | -0.59 | -0.23 | 0.82 |
| SVM | -2.11 | -0.61 | -0.64 | -1.06 | -0.93 | -0.99 | -0.70 | -0.01 | 0.27 |

不同错误

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 3 | 4 | 17 | 18 | 24 | 31 | 37 | 50 | 53 |
| LDA | 0.33 | 0.13 | 0.19 | 0.36 | 0.04 | 0.09 | 0.53 | 0.28 | 0.04 |
| SVM | -0.0011 | -0.30 | -0.29 | -0.10 | -0.59 | -0.62 | -0.38 | -0.27 | -1.14 |
|  | 55 | 68 | 71 | 78 | 82 | 91 | 92 | 110 |
| LDA | 0.14 | 0.09 | 1.05 | 0.41 | 0.70 | 0.01 | 0.80 | 0.62 |
| SVM | -0.36 | -0.32 | -0.74 | -0.32 | -1.24 | -1.55 | -0.32 | -0.33 |

1. SVM和LDA出错序号不同
2. 组合结果在LDA结果基础上修改了两个原有错误，加入了3个新错误。
3. LDA倾向于产生多的正预测，SVM倾向于产生多的负预测。
4. 但是两者均在各自预测错误的部分样本上得到了非常高的错误的置信度。例如：SVM在17、24、31、37、50、53、55都得到了绝对值和LDA预测相当或更大的错误预测值。LDA在71、78、92、110同样如此。这使得bagging方法效果并不理想。

**第三个方法：（没有实施）采用RBG三通道彩色图像的预测结果和Retinex图像增强算法生成的图像的预测结果组合为最终结果。**

观察训练集和测试集发现，本次分类目标是找到一个金发白人（正样本全为彩色），而其他图像有彩色有黑白有亚洲人有非洲人。因此图像颜色是很重要的信息。所以设想加入RGB三通道彩色图像进行预测。

观察正样本发现：有许多不同光照下的照片，因此想消除光照影响，所以考虑加入利用MSR(Multi-Scale Retinex)生成的图像作为分类的另一依据。因为这种图像增强算法常用来去雾或消除光照影响。[1][2]

最终的结果也许可以通过两者分类结果归一化后加权求和得到。

**Reference：**

1. https://blog.csdn.net/ajianyingxiaoqinghan/article/details/71435098?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522160335469519724813212202%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=160335469519724813212202&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_v2~rank\_v28-2-71435098.first\_rank\_ecpm\_v3\_pc\_rank\_v2&utm\_term=MSR&spm=1018.2118.3001.4187

[2]H. Chen, G. Hu, Z. Lei, Y. Chen, N. M. Robertson and S. Z. Li, "Attention-Based Two-Stream Convolutional Networks for Face Spoofing Detection," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 578-593, 2020, doi: 10.1109/TIFS.2019.2922241.