基于 PCA 的人脸识别

17341146 王程钥

1 简介

本次数字图像处理的大作业是需要完成一个基于主成分分析(PCA)的人脸识别模型。模型首先使用 PCA 算法对训练数据进行降维处理，将图像特征从高维空间映射到低维空间。后对测试集在低维空间上使用贪心算法，寻找距离最近的训练样本，并将该样本的标签赋给测试样本，得到预测结果。

我使用剑桥大学 ORL 人脸数据库，将一半的数据作为训练集，一半的数据作为测试集，最终在测试集上取得了 92%的准确率。

2 相关工作

2.1 主成分分析(PCA)

PCA 是一种经典的无监督数据降维算法，可以将数据从高维空间映射到低维空间。定义 N 组 D 维训练集数据$X \in R^{N*D}$，协方差矩阵$C \in R^{D*D}$，低维空间基向量矩阵$V \in R^{K*D}$，PCA 算法的步骤如下。

(1) 对数据进行标准化处理，根据公式(1)将 X 中的每个数据点减去数据均值。

(2) 根据公式(2)构造 D*D 维的协方差矩阵 C。

(3) 对协方差矩阵进行特征值分解，得到特征值和特征向量。

(4) 提取出前 K 大的特征值对应的特征向量，作为低维特征空间的基向量。基向量构成矩阵 V，作为数据集的低维近似估计。

(5) 根据公式(3)将训练集投影到低维空间上，得到 N*K 维的向量X'作为向量 X 的低维近似表示。

$$\overline{X_i} = X_i - \frac{1}{N}\sum_{I=1}^{N} X_I \qquad (1)$$

$$C = \frac{1}{N-1}\sum_{I=1}^{N} \overline{X_i}\,\overline{X_i}^T \qquad (2)$$

$$X' = X * V^T \qquad (3)$$

PCA 算法可以从高维特征空间中提取主要信息，构建维度更低的低维空间。和高维空间相比，低维空间上的属性信息量更大，也更复杂。

2.2 距离度量(metric)

在测试的时候，需要使用公式(3)将测试集数据从 D 维空间映射到 K 维空间，后在 K 维空间上寻找距离最近的向量，将其标签作为预测结果，本次实验我测试了一下三种距离度量方式，分别是 L1 范数(4)，L2 范数(5)和相关系数(6)。不同距离度量的实验结果见 4.3 节。

$$L1(x_i, x_j) = ||\alpha_p - \alpha_l||_2 = \sum_{k=1}^{K} |x_{i,k} - x_{j,k}| \qquad (4)$$

$$L2(x_i, x_j) = ||\alpha_p - \alpha_l||_1 = \sum_{k=1}^{K}(x_{i,k} - x_{j,k})^2 \qquad (5)$$

$$Cor(x_i, x_j) = \frac{\alpha_p * \alpha_l}{||\alpha_p|| * ||\alpha_l||} = \frac{\sum_{k=1}^{K} x_{i,k} * x_{j,k}}{(\sum_{k=1}^{K} {x_{i,k}}^2) * (\sum_{k=1}^{K} {x_{j,k}}^2)} \qquad (6)$$

3 方法描述与实现

3.1 方法描述

Algorithm: Face Recognization

```
Input: train_x, train_y, test_x, test_y, K
Output: predict_y
eig_vec = PCA(train_x, K)
train_project = eig_vec * train_x
for img in test_x
    project = eig_vec  * img
    vec, label = nearest vector with project in train_project
predict_y.append(label)
```

以上为人脸识别的伪代码，先使用 PCA 算法获取训练集的特征向量，后将训练集和测试集根据特征向量投影到低维空间。对于测试集，在低维空间上选择距离最近的向量，将它对应的 label 作为预测结果。

3.2 实现过程

```
[train_x, train_y, test_x, test_y] = Load_Dataset('.\dataset');
mean_face = mean(train_x, 1);
train_x = train_x - mean_face;
[coeff, score, latent, tsquared, explained] = pca(train_x);
accuracy_L1 = Test(test_x, test_y, coeff, score, mean_face, train_y, K,
'L1');
```

以上为主代码逻辑，先获取数据，后将其标准化，后进行 PCA，最后进行测试。

```
function accuracy = Test(test_x, test_y, coeff, score, mean_face,
train_y, K, metric)
    train_len = size(train_y,2);
    test_len = size(test_y,2);
    coeff = coeff(:,1:K);
    score = score(:,1:K);
    accuracy = 0;
    for i=1:test_len
        inputs = test_x(i,:);
        labels = test_y(i);
        feature = (inputs - mean_face) * coeff;
        similarity = zeros(1,train_len);
        for i=1:train_len
```

```
            if metric == "Cor"
                val = corrcoef(score(i,:), feature);
                if K>1
                    val = val(1,2);
                end
            elseif metric == "L1"
                val = -norm(score(i,:) - feature, 1);
            elseif metric == "L2"
                val = -norm(score(i,:) - feature, 2);
            end
            similarity(i) = val;
        end
        [best_val, best_idx] = max(similarity);
        if train_y(best_idx) == labels
            accuracy = accuracy + 1;
        end
    end
    accuracy = accuracy / test_len;
end
```

以上为测试部分的代码，将训练集和测试集投影到 K 维空间，寻找最近向量即可。

```
function Show_Eigen_Face(coeff , A, B)
    H=112; W=92; length=A*B;
    for i=1:length
        img = coeff(:,i);
        img = reshape(img,H,W);
        img = MIN_MAX(img);
        subplot(A,B,i);
        imshow(img);
        title(['eigen ',num2str(i)]);
    end
end
```

以上为显示特征脸的代码。将特征向量 reshape 到图像空间，并拉伸到[0,255]，显示即可。

4 实验结果与分析

4.1 数据集与训练设置

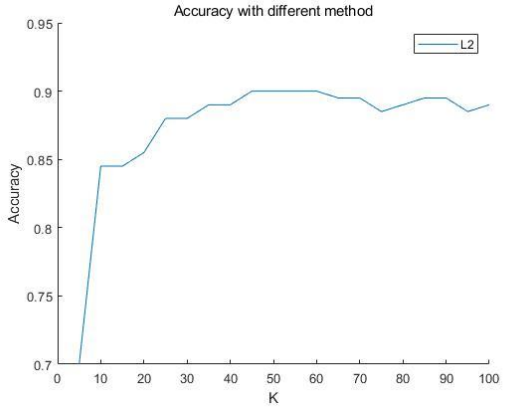
本次实验使用了剑桥大学 ORL 人脸数据库，该数据库包含 40 中人脸，每种人脸包含 10 张图像。根据作业要求。训练集由每张人脸的前 5 张图组成，共有 200 张图。测试机由每张脸的后五张图组成，同样也有 200 张图。

4.2 不同的维度

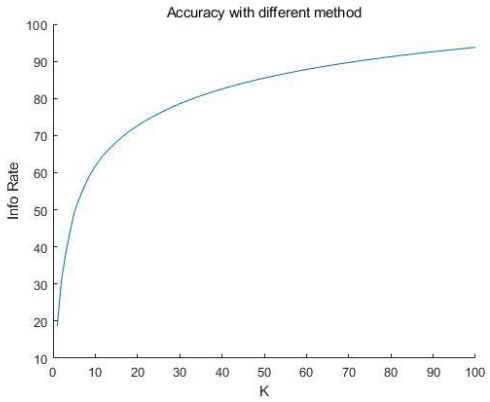
图 1 和表 1 为在不同维度下，使用 L2 范数作为距离度量尺度的结果。

可以看到，随着维度的上升，准确率先上升后下降。当维度较小时，大量有用的信息被过滤掉，导致分类准确率不高。当维度较大时，保留了过多的冗余信息，信息杂质较多，准确率也会略微下降。

表 1 的第二行和图 2 为不同维度下特征向量的信息保持量的比例，这可以根据特征值的比例算出。如图表可以看到，10 个特征就可以保存下 62%的信息，在 K=50 的时候信息保留率大约有 85%，准确率最高。



(图 1)



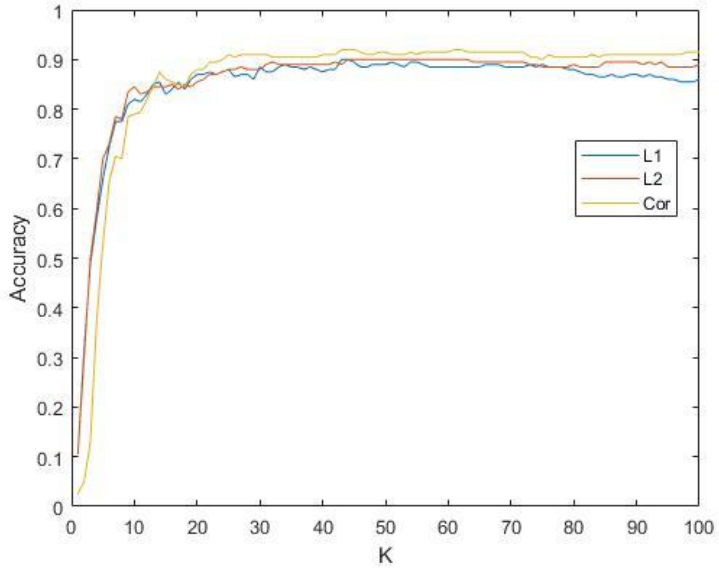
(图 2)

Attribute\K	10	30	50	80
Accuracy	82.00%	88.00%	90.00%	89.00%
Rate	61.73%	78.56%	85.48%	91.24%

(表 1)

4.2 不同的距离度量

我分别使用 L1 范数，L2 范数和相关系数作为距离度量，从 1 到 100 枚举 PCA 降维的维度。如图 3 所示，当 K 较小时 L1 范数和 L2 范数的准确率较高，不过当 K 提高后，相关系数的准确率更高，可以达到 92%。表 2 为部分准确率结果，也验证了这个结论。



(图 3)

Metric\K	10	30	50	80
L1	84.50%	88.50%	89.00%	88.00%
L2	82.00%	88.00%	90.00%	89.00%
Cor	79.00%	91.00%	91.50%	90.50%

(表 2)

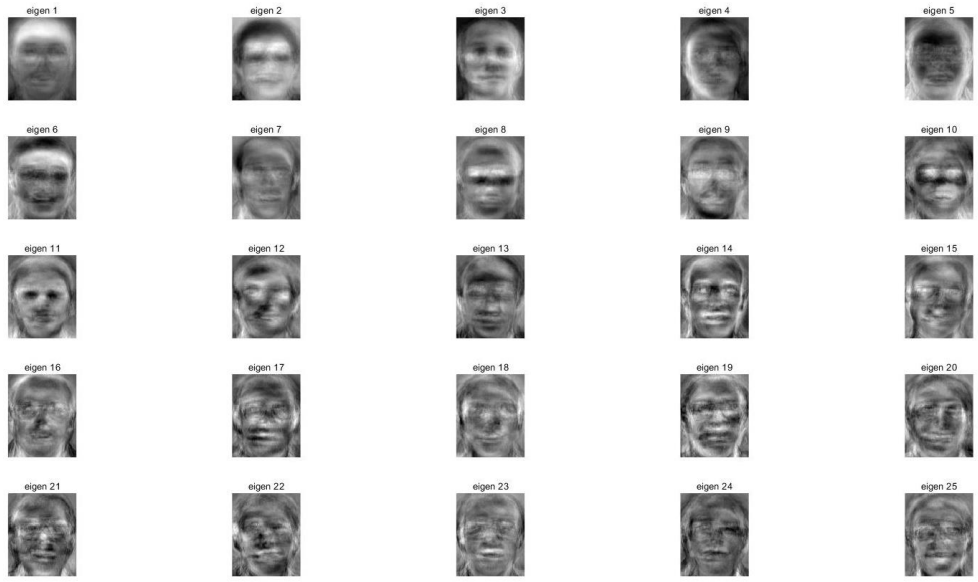
4.4 实验结果分析

表 3 为三种尺度下最高准确率和其对应的维度 K(准确率相同选最小 K)。可以看到使用相关系数作为距离度量的准确率最高，当 K=43 的时候准确率可以达到 92%。使用 L1 和 L2 范数的最高准确率都是 90。可以发现，当 K 取 50 左右模型的性能最好。

Metric	K	Accuracy
L1	43	90.00%
L2	44	90.00%
Cor	43	92.00%

(表 3)

随后我又将模型产生的特征脸进行可视化，特征脸如图 4 所示。



(图 4)

4 总结

本次实验实现了一个基于 PCA 的人脸识别模型。实验结果表明 PCA 算法可以提取图像的关键特征，并筛去图像的冗余特征。最终在使用基本的 L2 范数度量下可以获得 90%的准确率，使用基于相关系数的度量，在 K=43 的时候模型可以取得 92%的准确率。

通过本次实验，我对主成分分析(PCA)算法有了更深入的理解，对 matlab 的使用也更加熟练。

参考文献

[1] Eigenfaces face recognition (MATLAB):
https://blog.cordiner.net/2010/12/02/eigenfaces-face-recognition-matlab/
[2] Eigenfaces(wikipedia):
https://zh.wikipedia.org/wiki/%E7%89%B9%E5%BE%81%E8%84%B8