《大数据分析方法》课程实验报告

学号：2020204246

姓名：王浩

专业：计算机科学与技术

班级：20 图灵

实验五 SVC 分类器的设计与应用

实验目的：

1. 熟悉 SVC 的基本设计原理。
2. 掌握 SVC 的使用方法。
3. 利用 SVC 实现人脸识别。

实验内容：

1. 数据库的选择

可选取 ORL 人脸数据库作为实验样本，总共 40 个人，每人 10 幅图像，图像大小为

112*92 像素。图像本身已经经过处理，不需要进行归一化和校准等工作。实验样本分为训练样本和测试样本。首先设置训练样本集，选择 40 个人前 5 张图片作为训练样本，进行训 练。然后设置测试样本集，将 40 个人后 5 张图片作为测试样本，进行选取识别。

2. 实验基本步骤

人脸识别算法步骤概述：

a) 读取训练数据集；

若 flag=0，表述读取原文件的前五幅图作为训练数据，若 flag=1，表述读取原文件

的后五幅图作为测试数据，数据存入 f_matrix 中，每一行为一个文件，每行为 112*92

列。

b) 主成分分析法降维并去除数据之间的相关性；

c) 数据规格化；

d) SVC 训练（选取径向基和函数）得到分类函数；

e) 读取测试数据、降维、规格化；

f) 用步骤 d 产生的分类函数进行分类（多分类问题，采用一对一投票策略，归位得票

最多的一类）；

g) 计算正确率。

实验要求：

1. 分别使用 PCA 降维到 20,50,100,200，然后训练分类器，对比分类结果，画出对比曲线；
2. 变换 SVC 的 kernel 函数，如分别使用径向基函数和多项式核函数训练分类器，比分类结果，画出对比曲线
3. 使用交叉验证方法，变换训练集及测试集，分析分类结果。

一. 问题描述

实验流程 通过 PCA 降维，然后进行 SVC 预测，评估分类结果

比较不同 PCA 所要保留的主成分个数，以及不同 SVC kernel 算法对分类结果精确度的影响

二. 解决思路

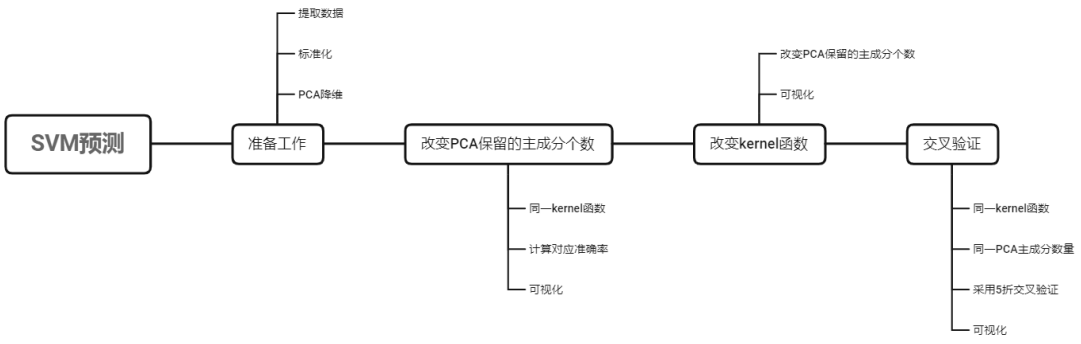

图 1 问题解决流程图

2.1 数据集处理

2.1.1 生成数据

采用opencv提取图片，并加上对应的标签，将原来112*92的数据转为1*10304的数据，方便运算

```
    t_img = cv2.imread(imgPath,0)
    t_img = t_img.reshape(1,t_img.size)
```

	0	1	2	3	4	5	6	7	8	9	...	10294	10295	10296	10297	10298	10299	10300	10301	10302	10303
0	48.0	49.0	45.0	47.0	49.0	57.0	39.0	42.0	53.0	49.0	...	39.0	44.0	40.0	41.0	49.0	42.0	44.0	47.0	46.0	46.0
0	60.0	60.0	62.0	53.0	48.0	51.0	61.0	60.0	71.0	68.0	...	27.0	35.0	28.0	33.0	31.0	31.0	37.0	32.0	34.0	34.0
0	39.0	44.0	53.0	37.0	61.0	48.0	61.0	45.0	35.0	40.0	...	23.0	30.0	36.0	32.0	28.0	32.0	31.0	29.0	26.0	29.0
0	63.0	53.0	35.0	36.0	33.0	34.0	31.0	35.0	39.0	43.0	...	173.0	169.0	166.0	161.0	158.0	169.0	137.0	41.0	10.0	24.0
0	64.0	76.0	80.0	53.0	34.0	72.0	60.0	66.0	66.0	50.0	...	31.0	28.0	34.0	32.0	35.0	34.0	35.0	35.0	37.0	39.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
0	123.0	121.0	126.0	122.0	127.0	127.0	123.0	124.0	123.0	127.0	...	29.0	47.0	34.0	36.0	42.0	34.0	39.0	40.0	35.0	42.0
0	129.0	127.0	133.0	124.0	131.0	129.0	130.0	129.0	127.0	132.0	...	91.0	92.0	93.0	90.0	90.0	92.0	89.0	93.0	93.0	93.0

图 2 提取数据图

2.1.2 数据标准化

采用sklearn中的StanderScaler方法进行标准化

```
StdData(X_train,X_test):
stdScaler = StandardScaler().fit(X_train)
X_train = stdScaler.transform(X_train)
X_test = stdScaler.transform(X_test)
```

```
]: array([[-1.05125993, -1.02502268, -1.15078496, ..., -0.63687931,
        -0.67224364, -0.66490378],
       [-0.71590752, -0.71676625, -0.67275578, ..., -0.95670487,
        -0.94235661, -0.93789482],
       [-1.30277423, -1.16513923, -0.92583005, ..., -1.02066998,
        -1.12243192, -1.05164108],
       ...,
       [ 1.10058468,  0.93660913,  1.07064477, ..., -0.87141805,
        -0.82980954, -0.8013993 ],
       [ 0.93290848,  0.96463244,  0.95816732, ...,  0.25863227,
         0.40820822,  0.22231707],
```

图 3 标准化后的数据

2.1.2 PCA 降维

采用sklearn中的PCA进行降维

```
def PcaDimRedu(X_train,X_test,comNum = 20):
    pca = PCA(n_components = comNum).fit(X_train)
    X_train = pca.transform(X_train)
    X_test = pca.transform(X_test)
    return X_train,X_test
```

```
array([[-44.98686497,   0.79780748,  57.65660109, ..., -16.21149117,
         -1.21559258,   6.58445167],
       [-73.12715966,  30.48094331,  -2.7942089 , ...,  -6.84819268,
         -2.73291628,  19.89296261],
       [-55.55827916,  32.51079789,  36.09077798, ...,  -6.42957408,
          8.64824995,  11.09140112],
       ...,
       [-31.40229174,  -4.984277  , -39.65869395, ..., -11.20441053,
         -0.31612789, -14.25088414],
       [-51.81593777,  -2.73537389, -21.98322655, ...,  -7.0996897 ,
         -8.92654812,   7.22901968],
       [-20.27568188,  -0.10651979, -51.22132165, ...,  -1.41691806,
         -1.04359892,  -2.26642646]])
```

图 4 降维后的数据

三.　模型构建及结果分析

3.1 模型结果

改变PCA保留的主成分数量，计算准确率的变化以及运行时间的变化，可视化如图5所示。

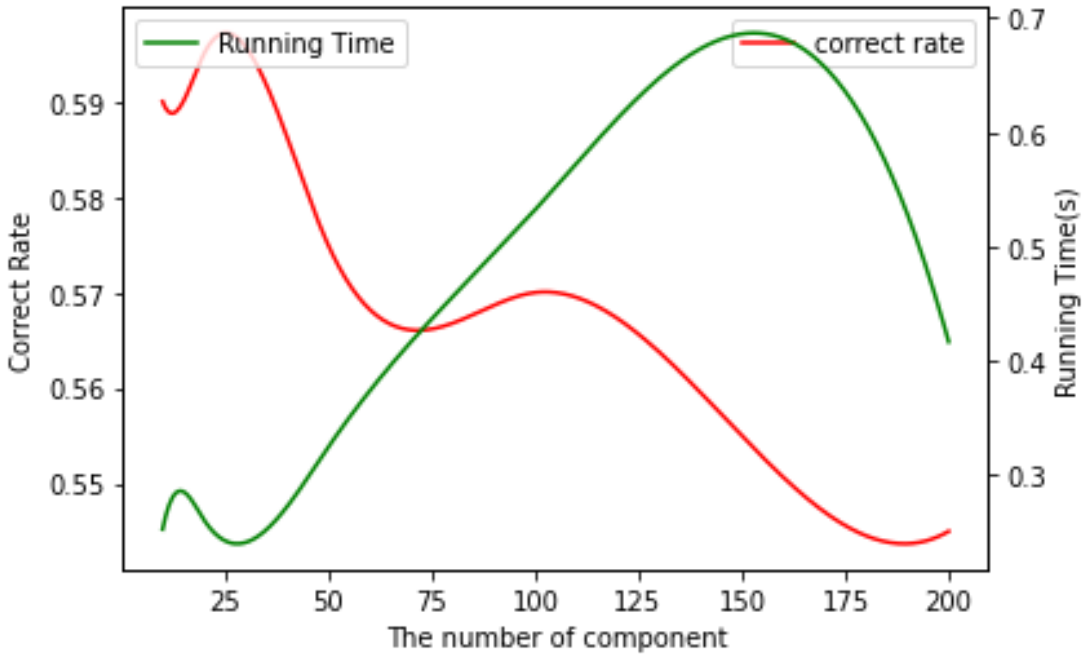

图 5 准确率-PCA 维数图

改变不同的 kernel 函数，不同 PCA 保留的主成分数量，计算准确率的变化，可视化如图 6 所示。

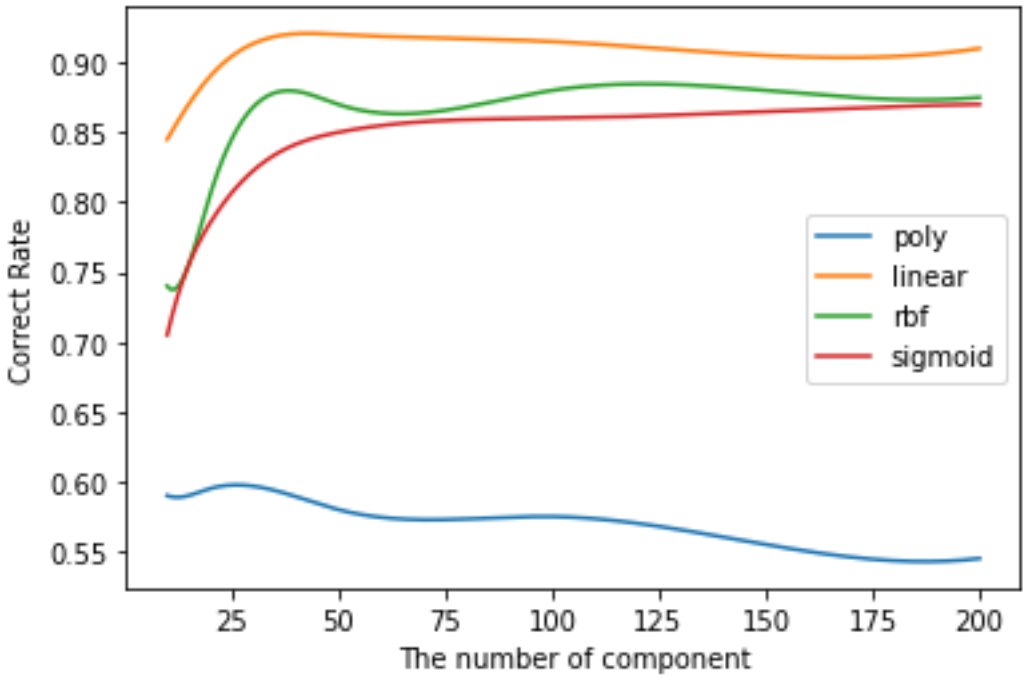

图 6 准确率-kernel 图

采用交叉验证的方式，变换训练集及测试集，分析分类结果，kernel函数采用默认的rbf函数，采用cross_val_score函数对分类结果进行评估。

```
]: Cross_validation()
   # 5折交叉验证下的准确率
```

```
[0.975  0.95   0.9875 0.975  0.9375]
```

图 7 交叉验证评估结果

3.2 实验中存在的问题及解决方法

对数据进行交叉验证时一开始采用了accuracy评分方法，结果全0，采用f1_samples评分方法，结果为nan，最后选择正确的recall_macro评分方法，得到了合理的评分结果

对于SVC的参数不太了解，比如如何使用径向基函数等相关知识，通过查阅相关资料，解决了相关问题

。