《人工智能与模式识别》

实 验 报 告 书

班级： 计科1801

学号： 汤礽禾

姓名： 181604124

指导教师： 翟婷婷

2020-2021 学年 第 二 学期

**实验名称： 分类算法在人脸图像分类中的应用**

**实验时间： 2021 年 5 月 8 日 第 10 周 星期 六**

一、实验目的

1．熟悉matlab中的统计和机器学习工具箱（statistic and machine learning toolbox）和计算视觉工具箱(computer vision system toolbox)，对其中的关键函数有一定了解。

2. 熟悉各种分类方法的原理，包括二分类和多分类的支持向量机、K近邻、朴素贝叶斯和决策树，能够灵活运用来解决给定的人脸图像分类问题。

3．掌握设计一个模式识别系统需要的重要设计环节：数据搜集和预处理、特征提取与选择、分类器设计、预测与评估。

二、实验预习（预备知识的问题及回答）

1. 求解一个多分类问题，可以通过将其转化为多个二分类问题来解决，简述one-vs-all 和one-vs-one的转化策略是如何解决多分类问题的。（1分）

**解答：**

**one-vs-all：对于训练集中的每一类实例，训练一个线性判别函数，将属于该类的实例和不属于该类的实例分开。**

**one-vs-one：对于训练集中任意两个类的实例，训练一个线性判别函数，将这两类的实例分开。**

2. 在使用K近邻方法进行分类前，需要先选择距离度量函数，常用的距离函数有哪些，请分别写出它们的计算公式。（请至少写出4个）（1分）

**解答：**

**闵可夫斯基：**

**欧式距离： **

**曼哈顿距离： **

**各维距离最大值： **

三、实验内容（包含实验所用命令或相关程序源代码）

本实验所使用的**人脸图像数据集**为：ORL人脸数据集，该数据集共包含40个不同人的人脸图像，每个人有10张图像，一共400张图像，是在1992年4月至1994年4月期间由英国剑桥的Olivetti研究实验室创建。所有的图像是以PGM格式存储，灰度图，图像大小宽度为92，高度为112。每张图像是在不同的时间、不同的光照、不同的面部表情(睁眼/闭眼，微笑/不微笑)和面部细节(戴眼镜/不戴眼镜)环境下采集的。

1. 二分类应用：

人脸图像性别分类数据集orl\_faces\_gender是ORL人脸数据集的一个子集，其中根据人脸图像的性别，将图像分为男性和女性的人脸图片。要求使用matlab，在该数据集上构建一个SVM分类器，用来预测一张人脸图像是男性的还是女性的，评估该分类器的分类准确率。

(a)准备工作：使用**图像的像素矩阵**作为每一张图像的特征：

构建**训练**数据集和**测试**数据集，训练数据集用来训练分类模型，测试数据集用来评估分类模型的精度。具体地，使用orl\_faces\_gender数据集中每个人的前7张图像作为训练使用的图像，后3张图像作为测试使用的图像。对这些图像依次提取像素特征，可以得到训练数据矩阵和测试数据矩阵，其中矩阵的每一行表示一张图像。还要对每张图像进行标记，表明它是男性还是女性的人脸，可以分别得到训练数据和测试数据的标记向量，该向量的第i个元素的值表示第i张的人脸图像是男性的还是女性的人脸。

**matlab代码：（3分）**

clear;clc;close all;

%% a

testSet = [];

trainSet = [];

test\_label = [];

%遍历 女

faceDataPath = 'E:\扬大\模式识别\实验\实验3 5.8\trh\orl\_faces\_gender\female\';

faceDataDir = dir(faceDataPath);

for i = 1:length(faceDataDir)

if(isequal(faceDataDir(i).name,'.')||isequal(faceDataDir(i).name,'..')|| ~faceDataDir(i).isdir)

continue;

end

faceImgDir = dir([faceDataPath, faceDataDir(i).name, '/\*.pgm']);

for j = 1 : 7

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

trainSet = [trainSet; im2double(reshape(temp, 1, [])), 0];

end

for j = 8 : 10

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

testSet = [testSet; im2double(reshape(temp, 1, []))];

test\_label = [test\_label; 0];

end

end

%遍历 男

faceDataPath = 'E:\扬大\模式识别\实验\实验3 5.8\trh\orl\_faces\_gender\male\';

faceDataDir = dir(faceDataPath);

for i = 1:length(faceDataDir)

if(isequal(faceDataDir(i).name,'.')||isequal(faceDataDir(i).name,'..')|| ~faceDataDir(i).isdir)

continue;

end

faceImgDir = dir([faceDataPath, faceDataDir(i).name, '/\*.pgm']);

for j = 1 : 7

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

trainSet = [trainSet; im2double(reshape(temp, 1, [])), 1];

end

for j = 8 : 10

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

testSet = [testSet; im2double(reshape(temp, 1, []))];

test\_label = [test\_label; 1];

end

end

(b) 使用fitcsvm函数训练一个**二分类的线性SVM模型**，要求SVM的优化算法采用Sequential Minimal Optimization (SMO)，使用predict函数对训练出的模型在测试数据集上进行预测，将得到的类标预测值与测试数据真实的类标进行比较，计算测试数据中被正确分类的样本所占的比例，称为测试数据集上的分类准确率，并在matlab的命令窗口输出。

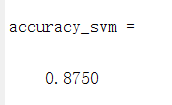
**matlab代码和运行截图：（3分）**

%% b

Md1 = fitcsvm(trainSet(:, 1:10304), trainSet(:, 10305), 'SOLVER', 'SMO');

predict\_svm = predict(Md1, testSet);

accuracy\_svm = sum(predict\_svm == test\_label) / length(test\_label)



(c) 使用fitcsvm函数训练一个**二分类的非线性SVM模型，**要求**核函数采用rbf核**，依次

设置rbf核函数的参数'KernelScale'为2, 4, 6, 8, 10，分别计算训练得到的分类器在测试数据集上的分类准确率，在matlab的命令窗口输出。**(3分)**

**matlab代码和运行截图：**

%% c

for i = 2 : 2 : 10

Md1 = fitcsvm(trainSet(:, 1:10304), trainSet(:, 10305), 'KernelFunction','RBF',...

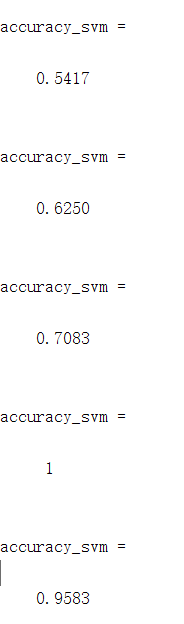
'KernelScale', i);

% 使用测试图像的特征向量预测样本标签

predict\_svm = predict(Md1, testSet);

accuracy\_svm = sum(predict\_svm == test\_label) / length(test\_label)

End



2. 多分类应用：

利用ORL人脸数据集中40个人的人脸图像，构建分类器，实现对这40个人的人脸图像进行分类。

(a) 构建**训练**数据集和**测试**数据集，使用**图像的像素矩阵**作为每一张图像的特征：使用ORL人脸数据集中每个人的前7张图像作为训练使用的图像，后3张图像作为测试使用的图像。对这些图像依次提取像素特征，可以得到训练数据矩阵和测试数据矩阵，其中矩阵的每一行表示一张图像。还要对每张图像进行标记，表明它是哪个人的图像，可以分别得到训练数据和测试数据的标记向量，该向量的第i个元素的值表示第i张的人脸图像是哪个人的。

**matlab代码：（3分）**

clear;clc;close all;

%%

testSet = [];

trainSet = [];

test\_label = [];

%遍历

faceDataPath = 'E:\扬大\模式识别\实验\实验3 5.8\trh\orl\_faces\_original\';

faceDataDir = dir(faceDataPath);

for i = 1:length(faceDataDir)

if(isequal(faceDataDir(i).name,'.')||... % 去除系统自带的两个隐文件夹

isequal(faceDataDir(i).name,'..')||...

~faceDataDir(i).isdir) % 去除遍历中不是文件夹的

continue;

end

faceImgDir = dir([faceDataPath, faceDataDir(i).name, '/\*.pgm']);

for j = 1 : 7

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

trainSet = [trainSet; im2double(reshape(temp, 1, [])), str2double(strrep(faceDataDir(i).name,'s',''))];

end

for j = 8 : 10

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

testSet = [testSet; im2double(reshape(temp, 1, []))];

test\_label = [test\_label; str2double(strrep(faceDataDir(i).name,'s',''))];

end

end

(b) 使用fitcecoc函数训练一个**多分类的SVM模型**，观察当多分类转化为二分类的策略分别采用one-vs-one和one-vs-all方式时，训练出的分类器在测试数据集上的分类准确率分别为多少，并在matlab的命令窗口输出。（提示：改变fitcecoc函数的coding属性）

**matlab代码和运行截图：（3分）**

%% b

Md1\_svm = fitcecoc(trainSet(:, 1:10304), trainSet(:, 10305), 'coding', 'onevsone');

predict\_svm = predict(Md1\_svm, testSet);

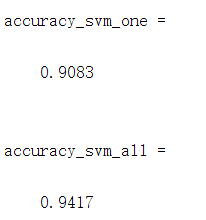
accuracy\_svm\_one = sum(predict\_svm == test\_label) / length(test\_label)

%

Md1\_svm = fitcecoc(trainSet(:, 1:10304), trainSet(:, 10305), 'coding', 'onevsall');

predict\_svm = predict(Md1\_svm, testSet);

accuracy\_svm\_all = sum(predict\_svm == test\_label) / length(test\_label)



(c) 训练一个**K近邻分类模型**，求出该模型在测试数据集上的分类准确率，并输出。**（2分）**

提示：使用fitcknn和predict。

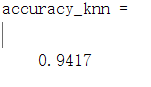
**matlab代码和运行截图：**

%% c

Md1\_knn = fitcknn(trainSet(:, 1:10304), trainSet(:, 10305));

predict\_knn = predict(Md1\_knn, testSet);

accuracy\_knn = sum(predict\_knn == test\_label) / length(test\_label)



(d)训练一个**朴素贝叶斯模型**，求出该模型在测试数据集上的分类准确率，并输出。**（2分）**

提示：使用fitcnb和predict。

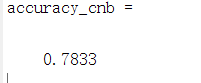
**matlab代码和运行截图：**

%% d

Md1\_cnb = fitcnb(trainSet(:, 1:10304), trainSet(:, 10305));

predict\_cnb = predict(Md1\_cnb, testSet);

accuracy\_cnb = sum(predict\_cnb == test\_label) / length(test\_label)



1. 特征提取应用：

(a) 提取图像的LBP(Local Binary Pattern，局部二值模式)特征来构建图像的特征向量，按照第2题的各个步骤依次训练SVM、KNN和Naïve Bayes分类器，计算并输出测试准确率。

提示：该题与上一题唯一的区别在于，使用图像的LBP而不是像素矩阵来构建特征向量。

利用视觉工具箱中的图像特征提取方法—extractLBPFeatures，提取一幅图像的LBP特征的代码示例：

t = imread(‘a.pgm’);

feature\_vector = extractLBPFeatures(t, 'Upright',true,'cellsize',[14 23]);

注意：用fitcnb时，用kernel density estimation来拟合概率分布，即要改变'DistributionNames' 属性的值为”kernel”。

**matlab代码和运行截图：（3分）**

clear;clc;close all;

%%

testSet = [];

trainSet = [];

test\_label = [];

%遍历当前文件夹下所有子文件夹中所有图片

faceDataPath = 'E:\扬大\模式识别\实验\实验3 5.8\trh\orl\_faces\_original\';

faceDataDir = dir(faceDataPath);

for i = 1:length(faceDataDir)

if(isequal(faceDataDir(i).name,'.')||... % 去除系统自带的两个隐文件夹

isequal(faceDataDir(i).name,'..')||...

~faceDataDir(i).isdir) % 去除遍历中不是文件夹的

continue;

end

faceImgDir = dir([faceDataPath, faceDataDir(i).name, '/\*.pgm']);

for j = 1 : 7

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

temp = extractLBPFeatures(temp, 'Upright',true,'cellsize',[14 23]);

trainSet = [trainSet; im2double(reshape(temp, 1, [])), str2double(strrep(faceDataDir(i).name,'s',''))];

end

for j = 8 : 10

temp = imread([faceDataPath, faceDataDir(i).name, '/', faceImgDir(j).name]);

temp = extractLBPFeatures(temp, 'Upright',true,'cellsize',[14 23]);

testSet = [testSet; im2double(reshape(temp, 1, []))];

test\_label = [test\_label; str2double(strrep(faceDataDir(i).name,'s',''))];

end

end

%% b

Md1\_svm = fitcecoc(trainSet(:, 1:1888), trainSet(:, 1889), 'coding', 'onevsone');

predict\_svm = predict(Md1\_svm, testSet);

accuracy\_svm\_one = sum(predict\_svm == test\_label) / length(test\_label)

%

Md1\_svm = fitcecoc(trainSet(:, 1:1888), trainSet(:, 1889), 'coding', 'onevsall');

predict\_svm = predict(Md1\_svm, testSet);

accuracy\_svm\_all = sum(predict\_svm == test\_label) / length(test\_label)

%% c

Md1\_knn = fitcknn(trainSet(:, 1:1888), trainSet(:, 1889));

predict\_knn = predict(Md1\_knn, testSet);

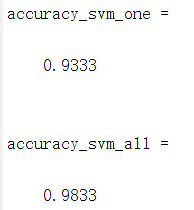
accuracy\_knn = sum(predict\_knn == test\_label) / length(test\_label)

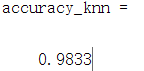
%% d

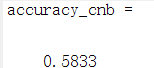
Md1\_cnb = fitcnb(trainSet(:, 1:1888), trainSet(:, 1889), 'DistributionNames', 'kernel');

predict\_cnb = predict(Md1\_cnb, testSet);

accuracy\_cnb = sum(predict\_cnb == test\_label) / length(test\_label)

****

****

****

(b) 观察不同的分类器在哪种特征提取方法下的测试准确率高？这里的特征提取指的是“像素特征”或者“LBP特征”。**（**1**分）**

**解答：**

**多分类的SVM模型：在LBP特征下测试准确率高**

**K近邻分类模型 ：在LBP特征下测试准确率高**

**朴素贝叶斯模型 : 在像素矩阵下测试准确率高**