《人工智能与模式识别》

实 验 报 告 书

班级： 计科 2002

学号： 202801108

姓名： 李蔚

指导教师： 翟婷婷

2022-2023 学年 第 二 学期

**实验名称： 分类算法综合应用**

**实验时间： 2023 年 4 月 7 日 第 8 周 星期 五**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题目 | 二 | 三 | | | | | | 总分 |
| 1(a) | 1(b) | 1(c) | 1(d) | 2(a) | 2(b) |
| 得分 |  |  |  |  |  |  |  |  |

一、实验目的

1. 熟悉各种分类方法的原理，包括多分类的支持向量机、K近邻和朴素贝叶斯。

2. 了解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。

3. 利用Maltab中的库函数训练出SVM多分类器、k近邻分类器和贝叶斯分类器，用来解决一个给定的人脸图像分类问题。

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：**在One-vs-All方法中，对于一个有n个类别的多分类问题，将每个类别与其他所有类别组合起来，形成n个二分类问题。当我们需要对一个新的样本进行分类时，我们将该样本输入到所有二分类器中，并选择具有最高置信度的二分类器的类别作为预测结果。

在One-vs-One方法中，对于一个有n个类别的多分类问题，我们每次选择两个类别进行组合，因此会形成n(n-1)/2个二分类问题。当我们需要对一个新的样本进行分类时，我们将该样本输入到所有二分类器中，并将它所属的类别选择为获得最多票数的类别。

三、实验内容（包含实验所用命令或相关程序源代码）

1. 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。要求利用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)** 利用imageDatastore函数得到orl\_faces\_original目录下所有人脸图像的路径信息，并以图像所在的文件夹名作为图像的类标。利用每张图像的路径信息，依次读取每张人脸图像，

构建**训练**数据集和**测试**数据集。具体地说，使用每个人的前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，将图像的**像素矩阵**转换为行向量作为每一张图像的特征向量，通过这种方式，得到**训练**数据矩阵trainData和**训练**类标向量trainLabel，以及**测试**数据矩阵testData和**测试**类标向量testLabel。**（4分）**

**matlab代码：**

% 使用imageDatastore函数获取所有人脸图像的路径信息

faceData = imageDatastore('D:\orl\_faces\_original\orl\_faces\_original', 'IncludeSubfolders', true, 'FileExtensions', '.pgm');

numImages = numel(faceData.Files);

% 初始化训练和测试数据矩阵以及训练和测试标签向量

trainData = zeros(280, 10304);

testData = zeros(120, 10304);

trainLabel = zeros(280, 1);

testLabel = zeros(120, 1);

trainCnt = 1;

testCnt = 1;

labelCnt = 1;

[~, folderName] = fileparts(fileparts(faceData.Files{1}));

% 逐一读取每张人脸图片，并将其转化为特征向量

for i = 1:numImages

% 读取图片

img = readimage(faceData, i);

% 将图像像素矩阵转换为 double 类型

img = im2double(img);

% 将图像像素矩阵拉伸为行向量

imgVec = reshape(img, 1, []);

% 根据文件名判断该图片的类别

[~, lastFolderName] = fileparts(fileparts(faceData.Files{i}));

if lastFolderName==folderName

label = labelCnt;

else

labelCnt = labelCnt+1;

label = labelCnt;

folderName = lastFolderName;

end

% 将特征向量和类别分别存储到训练或测试数据矩阵以及训练或测试标签向量中

if mod(i, 10) >= 1 && mod(i, 10) <= 7 % 训练数据

trainData(trainCnt,:) = imgVec;

trainLabel(trainCnt) = label;

trainCnt = trainCnt + 1;

else % 测试数据

testData(testCnt,:) = imgVec;

testLabel(testCnt) = label;

testCnt = testCnt + 1;

end

end

% 保存训练和测试数据以及标签

save('genderData.mat', 'trainData', 'testData', 'trainLabel', 'testLabel');

clear; % 清除工作区变量

clc; % 清空命令行

% 加载数据

load('genderData.mat');

**(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）**（2分）**

**matlab代码和运行截图：**

Md1 = fitcecoc(trainData, trainLabel,'Coding','onevsone');

% 对测试数据进行预测

predictedLabels = predict(Md1, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('多分类的SVM模型one-vs-one方式:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);

Md2 = fitcecoc(trainData, trainLabel,'Coding','onevsall');

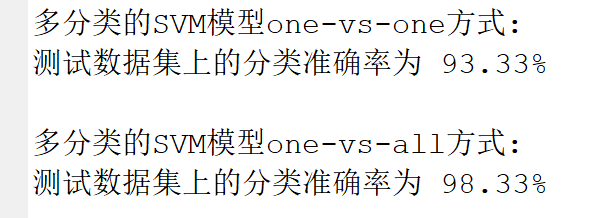
% 对测试数据进行预测

predictedLabels = predict(Md2, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('多分类的SVM模型one-vs-all方式:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);

****

**(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：**

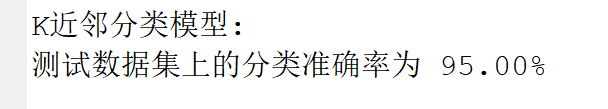
Md3 = fitcknn(trainData, trainLabel);

predictedLabels = predict(Md3, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('K近邻分类模型:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);



**(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：**

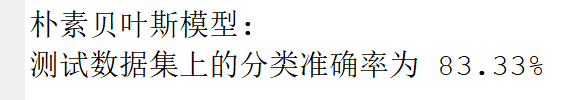
Md4 = fitcnb(trainData, trainLabel);

predictedLabels = predict(Md4, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('朴素贝叶斯模型:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);



2. 利用视觉工具箱中的图像特征提取方法—extractLBPFeatures函数，可以提取一幅图像的LBP特征(Local Binary Pattern，局部二值模式)。

(a) 在上题中，将特征提取的方式变为提取图像的LBP特征来构建图像的特征向量，然后按照第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：本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。**（3分）**

extractLBPFeatures函数使用示例：

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：**

% 使用imageDatastore函数获取所有人脸图像的路径信息

faceData = imageDatastore('D:\orl\_faces\_original\orl\_faces\_original', 'IncludeSubfolders', true, 'FileExtensions', '.pgm');

numImages = numel(faceData.Files);

% 初始化训练和测试数据矩阵以及训练和测试标签向量

trainData = zeros(280, 1888);

testData = zeros(120, 1888);

trainLabel = zeros(280, 1);

testLabel = zeros(120, 1);

trainCnt = 1;

testCnt = 1;

labelCnt = 1;

[~, folderName] = fileparts(fileparts(faceData.Files{1}));

% 逐一读取每张人脸图片，并将其转化为特征向量

for i = 1:numImages

% 读取图片

img = imread(faceData.Files{i});

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

% 根据文件名判断该图片的类别

[~, lastFolderName] = fileparts(fileparts(faceData.Files{i}));

if lastFolderName==folderName

label = labelCnt;

else

labelCnt = labelCnt+1;

label = labelCnt;

folderName = lastFolderName;

end

% 将特征向量和类别分别存储到训练或测试数据矩阵以及训练或测试标签向量中

if mod(i, 10) >= 1 && mod(i, 10) <= 7 % 训练数据

trainData(trainCnt,:) = imgVec;

trainLabel(trainCnt) = label;

trainCnt = trainCnt + 1;

else % 测试数据

testData(testCnt,:) = imgVec;

testLabel(testCnt) = label;

testCnt = testCnt + 1;

end

end

% 保存训练和测试数据以及标签

save('genderData.mat', 'trainData', 'testData', 'trainLabel', 'testLabel');

clear; % 清除工作区变量

clc; % 清空命令行

% 加载数据

load('genderData.mat');

Md1 = fitcecoc(trainData, trainLabel,'Coding','onevsone');

% 对测试数据进行预测

predictedLabels = predict(Md1, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('多分类的SVM模型one-vs-one方式:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);

Md2 = fitcecoc(trainData, trainLabel,'Coding','onevsall');

% 对测试数据进行预测

predictedLabels = predict(Md2, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('多分类的SVM模型one-vs-all方式:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);

Md3 = fitcknn(trainData, trainLabel);

predictedLabels = predict(Md3, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('K近邻分类模型:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);

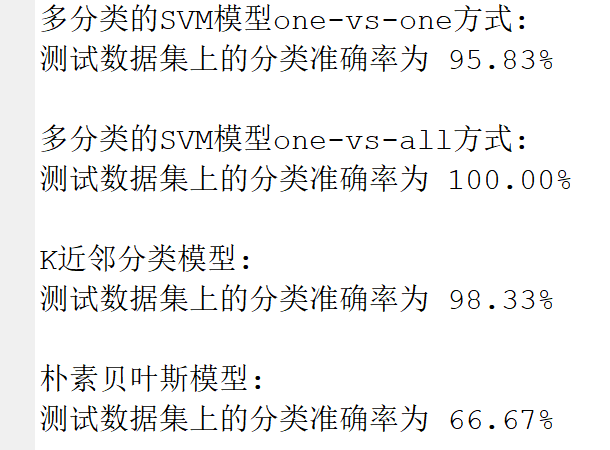
Md4 = fitcnb(trainData, trainLabel,'DistributionNames','kernel');

predictedLabels = predict(Md4, testData);

% 计算分类准确率

accuracy = sum(predictedLabels == testLabel) / numel(testLabel);

fprintf('朴素贝叶斯模型:\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);



(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one-vs-one方式：在LBP特征提取方法下的测试准确率高；

多分类的SVM模型one-vs-all方式：在LBP特征提取方法下的测试准确率高；

K近邻分类模型：在LBP特征提取方法下的测试准确率高；

朴素贝叶斯模型：在像素特征提取方法下的测试准确率高；