《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： SVM算法应用**

**实验时间： 2023 年 3 月 24 日 第 6 周 星期 五**

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

一、实验目的

1. 理解用于二分类的线性和非线性支持向量机算法的原理。

2. 掌握模式识别的重要环节：数据搜集、预处理、特征选择、分类器训练、预测与评估。

3. 利用Maltab中的库函数训练出线性和非线性的SVM分类器以解决一个基于人脸图像判断性别的二分类问题。

二、实验预习（预备知识的问题及回答）

1. 请用一句话概括线性可分的二分类SVM的基本思想是什么？（0.5分）

**解答：**

线性可分的二分类SVM的基本思想是通过寻找一个最优的超平面来将两个类别的数据分开，并使得两个类别到超平面的距离最大化。

1. 简述要利用广义线性化方法实现SVM的非线性分类，关键要解决什么问题，如何解决该问题。（0.5分）

**解答：**

利用广义线性化方法实现SVM的非线性分类需要解决的关键问题是如何将非线性可分问题转化为线性可分问题。

为解决该问题，可以通过引入核函数将输入空间映射到高维特征空间，从而使得原本在输入空间中非线性可分的问题在特征空间中变成线性可分问题，最终得到支持向量机的非线性分类器。

三、实验内容（包含实验所用命令或相关程序源代码）

1. 利用matlab的help命令查明以下函数/命令实现了什么功能。(每个0.5分，共1分)

(a) imgs = imageDatastore(filename,'includesubfolder',true, 'FileExtensions','.jpg);

imgpath= imgs.File{1};

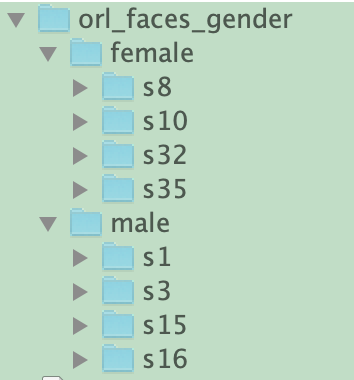
(b) save(‘data.mat’,’test’);

load gong.mat;

(a) save(‘data.mat’,’test’)用于创建一个名为 imgs 的图像数据存储对象，它包含了指定文件夹 filename 下的所有 .jpg 文件，包括子文件夹。imgs.File包含了所有的图像文件路径。imgpath = imgs.File{1}用于提取存储在 imgs 中的第一个图像文件的路径，将其赋值给变量 imgpath。

(b) save(‘data.mat’,’test’)用于将一个变量 test 保存到名为 data.mat 的 MATLAB 数据文件中。load gong.mat用于加载名为 gong.mat 的 MATLAB 数据文件到工作区中。

1. 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，其目录结构如下图所示：



其中，女性和男性的人脸图像各包含4个人，每人10张图，同一个人的图放在相同文件夹中。所有人脸图像均是以PGM格式存储的灰度图，图像大小宽度为92，高度为112。

要求使用matlab，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，利用训练数据集来训练SVM分类模型，在测试数据集上评估训练好的分类模型的精度。

**(a)** 利用imageDatastore函数得到orl\_faces\_gender目录下所有人脸图像的路径信息。利用每张图片的路径信息，依次读取每张人脸图片，将图片的unit8类型的像素矩阵转换为double型，将转换后的矩阵拉伸为1\*10304维的行向量，作为该图片的特征向量，并记录该图片的类标。如果第i张图片满足，i对10取余且，则将该图片对应的特征向量和类标分别放入**训练**数据矩阵trainData和**训练**标签向量trainLabel中，否则，分别放入**测试**数据矩阵testData和**测试**类标向量testLabel中。通过这种做法，会将每个人的前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。使用save函数将trainData，testData，trainLabel和testLabel这4个变量都存储在genderData.mat文件中，方便下面使用。

**matlab代码：（5分）**

% 使用imageDatastore函数获取所有人脸图像的路径信息

faceData = imageDatastore('D:\orl\_faces\_gender\orl\_faces\_gender', '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(56, 10304);

testData = zeros(24, 10304);

trainLabel = zeros(56, 1);

testLabel = zeros(24, 1);

trainCnt = 1;

testCnt = 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

img = readimage(faceData, i); % 读取图片

img = im2double(img); % 将图像像素矩阵转换为 double 类型

imgVec = reshape(img, 1, []);% 将图像像素矩阵拉伸为行向量

% 根据文件名判断该图片的类别

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

if startsWith(folderName, 'male')

label = 1; % 男性为正例

else

label = -1; % 女性为反例

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

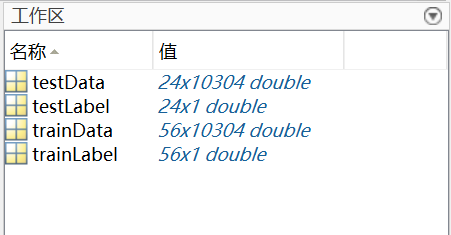
save('genderData.mat', 'trainData', 'testData', 'trainLabel', 'testLabel');

**(b)**先利用clear和clc命令清除工作区变量，清空命令行。利用load函数加载genderData.mat文件，得到trainData，testData，trainLabel和testLabel。**(1分)**

**matlab代码和工作区窗口截图：**

clear; % 清除工作区变量

clc; % 清空命令行

load('genderData.mat'); % 加载数据

**(c)**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，计算测试数据集中被正确分类的实例所占的比例，称为测试数据集上的分类准确率，并在matlab的命令窗口输出。**（3分）**

**matlab代码：**

% 训练SVM模型

svmModel = fitcsvm(trainData, trainLabel, 'KernelFunction', 'linear', 'Solver', 'SMO');

% 对测试数据进行预测

predictedLabels = predict(svmModel, testData);

% 计算分类准确率

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

fprintf('(c)\n测试数据集上的分类准确率为 %.2f%%\n\n', accuracy \* 100);

**(d)** 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**，使用**for循环**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：**

kernelScales = [2, 4, 6, 8, 10]; % 待测试的核函数参数

accuracies = zeros(size(kernelScales)); % 记录分类准确率

fprintf('(d)\n');

% 循环训练和测试不同参数下的SVM模型

for i = 1:length(kernelScales)

% 训练SVM模型

svmModel = fitcsvm(trainData, trainLabel, 'KernelFunction', 'rbf', 'KernelScale', kernelScales(i), 'Solver', 'SMO');

% 对测试数据进行预测

predictedLabels = predict(svmModel, testData);

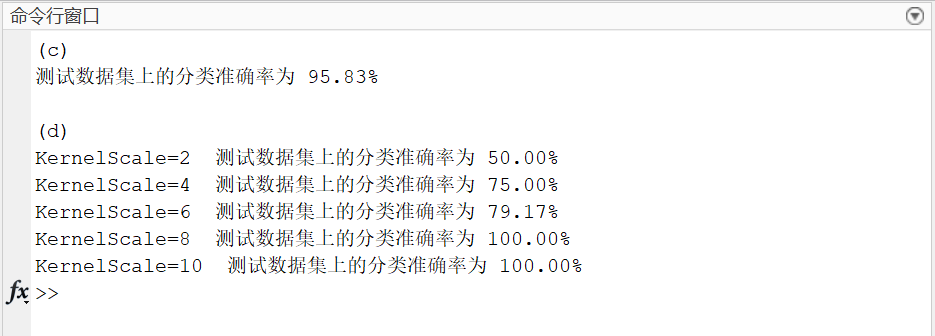
% 计算分类准确率

accuracies(i) = sum(predictedLabels == testLabel) / numel(testLabel);

% 输出结果

fprintf('KernelScale=%d 测试数据集上的分类准确率为 %.2f%%\n', kernelScales(i), accuracies(i) \* 100);

end

**(e)** 给出(c)和(d)运行结果的截图。**（1分）**