编程作业说明：SVM

任务一：分别用线性 SVM 和高斯核 SVM 预测对数据进行分类

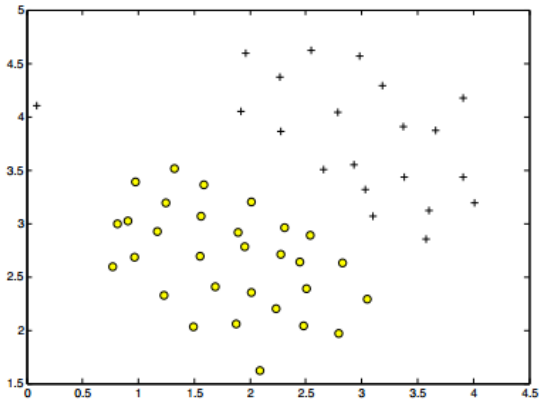
（1）问题：task1_linear.mat 中有一批数据点，试用线性 SVM 对他们进行分类，并在图中画分出决策边界。task1_gaussian 中也有一批数据点，试用高斯核 SVM 对他们进行分类，并在图中画出决策边界。

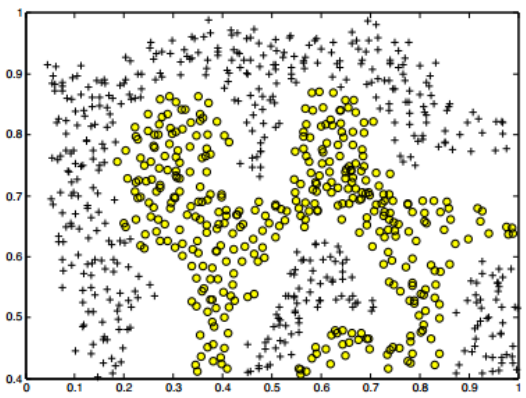
（2）步骤提示：

1.实验所需函数的代码在 SVM_Functions.py 中已经给出，请尝试读懂代码，理解算法思想与步骤，并自行读入数据，执行所需函数，观察实验结果。（也可以自己编写实验代码）

2.附件 PPT 中给出了 SMO 训练算法的详细分析。

3.task1_linear.mat 和 task1_gaussian.mat 图像如下所示：





4.训练过程：

（a）加载与可视化数据 loadData()，plotData()

（b）训练模型

model = SVM_Functions.svmTrain_SMO(X, y, C=1, max_iter=20)

model=SVM_Functions.svmTrain_SMO(X,y,C=1,kernelFunction='gaussian',K_matrix=s.gaussianKernel(X,sigma=0.1))

（c）决策边界可视化 SVM_Functions.visualizeBoundaryLinear(X, y, model)

（3）提交要求：

任务一需要编写实验报告进行简述其原理，代码步骤，并根据训练结果在图

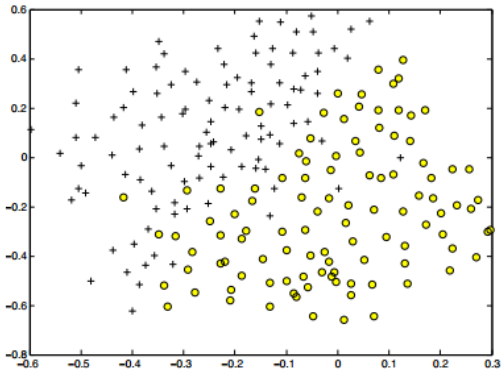
中画出决策边界。

任务二：使用高斯核 SVM 对给定数据集进行分类

（1）数据集讲解：给定数据集（文件 task2.mat），参考 task1 的代码，编程实现一个高斯核 SVM 进行分类。输出训练参数 C, sigma 分别取 0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30 时(共 64 组参数组合)的训练集上的准确率。(程序运行时间 8mins 左右，准确率 = 预测正确样本数/样本总数)

（2）提示：

1. 数据集可视化：



2. 程序运行结果举例（由于 SMO 算法的随机性，你的结果应该跟下面的例子不完全相同）：

```
Training Accuracy:
sigma   0.01   0.03    0.1    0.3      1      3     10     30
C
0.01   0.498  0.502  0.498  0.498  0.498  0.820  0.502  0.502
0.03   0.502  0.502  0.521  0.867  0.498  0.829  0.498  0.810
0.1    0.498  0.502  0.948  0.867  0.834  0.502  0.498  0.498
0.3    0.502  0.981  0.948  0.900  0.867  0.649  0.806  0.502
1      1.000  0.995  0.948  0.934  0.905  0.858  0.502  0.498
3      1.000  1.000  0.943  0.943  0.919  0.863  0.796  0.498
10     1.000  1.000  0.962  0.948  0.924  0.891  0.853  0.498
30     1.000  1.000  0.948  0.938  0.929  0.924  0.867  0.735
```

（3）提交要求：

编写实验报告，报告中需要包含：实验设置，编程思路，实验结果展示以及自己的思考等。其中实验结果展示按照上图的格式在实验报告中给出。

任务三：使用线性 SVM 实现对垃圾邮件分类

（1）数据集讲解：

编程实现一个垃圾邮件 SVM 线性分类器，分别在训练集和测试集上计算准确率。其中训练数据文件：task3_train.mat，要求导入数据时输出样本数和特征维度。测试数据文件：task3_test.mat，要求导入数据时输出样本数和特征维度，测试数据标签未给出。（程序运行时间 10mins 左右）

（2）步骤提示：

1. 对 SMO 算法的实现进行举例说明:

```
简化版SMO算法
1.Initialize alphas向量为0
2.    while 迭代次数小于最大迭代次数:
3.        for each alpha[i] in alphas:
4.            if alpha[i]可优化:
5.                随机选择另一个alpha[j],同时优化这两个向量。
6.            else :
7.                退出本次内循环。
8.        if 所有的alphas都没有被优化:
9.            增加迭代次数，进行下一次外循环。
```

（3）提交要求：

将测试数据预测结果按顺序存储为 txt 文件，每一行为一个样本的标签。将测试集预测结果的 txt 文件发送到邮箱 yxj603@foxmail.com。实验报告中需要写明具体实验流程，思路。