支持向量机

一、实验名称：支持向量机

二、实验目的

实现支持向量机，通过训练获得适当的参数。

三、实验原理

线性不可分的支持向量机需要实现软间隔最大化：

$$\min_{w,b,\zeta}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\zeta_i$$
$$s.t.\quad y_i(w^Tx_i+b)\geq 1-\zeta_i,\quad \zeta_i\geq 0,\quad \forall i=1,\cdots,N$$

这里的 zeta（ζ）⩾0 是松弛变量，是硬间隔到软间隔的变化部分。它保证了在线性不可分的交界处，对越界的数据点进行宽容。

为获得其最小值，对不同参数求偏导，得到它们各自为 0 的条件，再带回，就可以得到该问题的对偶问题：

$$\min_{a}\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}a_ia_jy_iy_jx_i^Tx_j-\sum_{i=1}^{N}a_i$$
$$s.t.\quad \sum_{i=1}^{N}a_iy_i=0,\quad 0\leq a_i\leq C,\quad \forall i=1\cdots,N$$

该对偶问题满足 KKT 条件：

$$\begin{cases}\alpha_i\geq 0,\quad \mu_i\geq 0\\ \xi_i\geq 0,\quad \mu_i\xi_i=0\\ y_i\left(\boldsymbol{w}^{\top}\boldsymbol{X}_i+b\right)-1+L_i\geq 0\\ \alpha_i\left[y_i\left(\boldsymbol{w}^{\top}\boldsymbol{X}_i+b\right)-1+L_i\right]=0\end{cases}$$

利用拉格朗日乘子法求解该问题，增广拉格朗日函数如下：

$$L(\alpha,\lambda)=\frac{1}{2}\alpha^T\hat{X}\hat{X}^T\alpha-1^T\alpha+\lambda y^T\alpha+\frac{\beta}{2}(y^T\alpha)^2$$

在迭代中获得拉格朗日参数：

$$\begin{aligned}\hat{\alpha}^k=&\alpha^k-\eta\nabla L(\alpha^k,\lambda^k)\\ =&\alpha^k-\eta\left(\hat{X}\hat{X}^T\alpha^k-1+\lambda^ky+\beta y^T\alpha^ky\right)\\ \alpha_i^{k+1}=&\begin{cases}\hat{\alpha}_i^k & \text{if } 0\leq\hat{\alpha}_i^k\leq C\\ 0 & \text{if } \hat{\alpha}_i^k<0\\ C & \text{if } \hat{\alpha}_i^k>C\end{cases}\\ \lambda^{k+1}=&\lambda^k+\beta(y^T\alpha^{k+1})\end{aligned}$$

根据拉格朗日参数，带回求解分类界面参数：

$$\boldsymbol{w^*=\sum_{i=1}^{N}\alpha_i^*y_ix_i}$$

$$\boldsymbol{b^*=y_j-\sum_{i=1}^{N}\alpha_i^*y_i(x_i\cdot x_j)}$$

获得分类界面：$\boldsymbol{w^*\cdot x+b^*=0}$，得到分类决策函数：$\boldsymbol{f(x)=sign(w^*\cdot x+b^*)}$。

该问题还可以转化为合页损失函数最小化：

$$\sum_{i=1}^{N}[1-y_i(w\cdot x+b)]_+ +\lambda\|w\|^2$$

针对原始问题和对偶问题，将训练数据集中对应 alpha（α）>0 的样本点称为支持向量即线性可分位于间隔边界上的样本点，线性不可分位于间隔边界上及间隔边界内的样本点。

四、实验步骤

1、生成数据集：利用中心点加高斯噪音，生成两类数据集，样本量 n=100，维度为 2 维，并为两类加不同标签。

2、SVM 模型初始化：初始化系数 w 和 b，并初始化对偶问题变量 alpha（α）=zeros(2*n,1）、拉格朗日参数 lambda（λ）=0.1、惩罚参数 beta（β）=0.1、惩罚参数 C=0.3、步长/学习率 eta（η）=0.00005、迭代次数 m=100000。

3、约束最优解：在迭代中，不断更新α和λ，并对超出[0,C]范围的α设回边界值，具体如下：

```
alpha = alpha - eta*(X_hat*X_hat'*alpha - 1 + lambda*Y + beta*Y'*alpha*Y)
lambda = lambda + beta*(Y'*alpha)
```

4、分类界面参数：根据迭代的结果，按照原理中给出的式子求解分类界面参数。

5、结果展示：在图像中标注数据集、分类超平面以及支持向量，并显示 w、b 以及代码运行时间。

6、函数结果：利用原理中给出的相应函数，求得线性增广拉格朗日函数和合页损失函数的值，作为参考。

7、训练：改动参数（如上标红），进行调试训练，综合考虑，获得合适的参数。

五、代码

主要的训练部分已标红

```
% 数据样本生成
n = 100;                  % 样本量
center1 = [1,1];          % 数据中心（第二类：可分[6,6]，不可分[3,3]）
center2 = [3,3];
X = zeros(2*n,2);         % 数据点（2 维）：高斯噪声
X(1:n,:) = ones(n,1)*center1 + randn(n,2);
X(n+1:2*n,:) = ones(n,1)*center2 + randn(n,2);
Y = zeros(2*n,1);         % 类别标签
Y(1:n) = 1;
Y(n+1:2*n) = -1;
X_hat = X .* Y;

%{
% 图一：数据点
figure(1)
set(gcf,'Position',[1,1,700,600], 'color','w')
set(gca,'Fontsize',18)
plot(X(1:n,1),X(1:n,2),'go','LineWidth',1,'MarkerSize',10);
hold on;
```

```
plot(X(n+1:2*n,1),X(n+1:2*n,2),'b*','LineWidth',1,'MarkerSize',10);
hold on;
xlabel('x axis');
ylabel('y axis');
legend('class 1','class 2');
%}

% SVM 模型
tic()
alpha = zeros(2*n,1);       % 对偶问题变量α
lambda = 0.1;              % 拉格朗日参数λ
beta = 0.1;                % 惩罚参数β
C = 0.3;                   % 惩罚参数 C
eta = 0.00005;             % 步长η
m = 100000;                % 迭代次数 m
for i=1:m                  % 求最优解
    alpha = alpha - eta*(X_hat*X_hat'*alpha-1+lambda*Y+beta*Y'*alpha*Y);
    alpha(alpha>C) = C;
    alpha(alpha<0) = 0;
    lambda = lambda + beta*(Y'*alpha);
end
idx = find(alpha<C&alpha>0);    % 分类界面参数
len = length(idx);
j = idx(randi(len));
w = X_hat' * alpha;
b = Y(j)-sum(Y.*alpha.*X*X(j,:)');
b_all = zeros(len,1);          % 观测不同 b 是否收敛
for i=1:len
    b_all(i) = Y(i)-sum(Y.*alpha.*X*X(i,:)');
end
L1 =
1/2*alpha'*(X_hat*X_hat')*alpha-sum(alpha)+lambda*Y'*alpha+beta/2*(Y'*alpha)
^2; % 线性增广拉格朗日函数
z = 1-Y.*(X*w+b);
z(z<0) = 0;
L2 =
sum(z)+1/2/C*(w'*w);
% 合页损失函数
disp(L1)
disp(L2)
toc()

% 图二：分类器可视图（x1 为横轴，y 为纵轴，1 为分类界面，2、3 为间隔边界）
x1 = -2 : 0.00001 : 7;
```

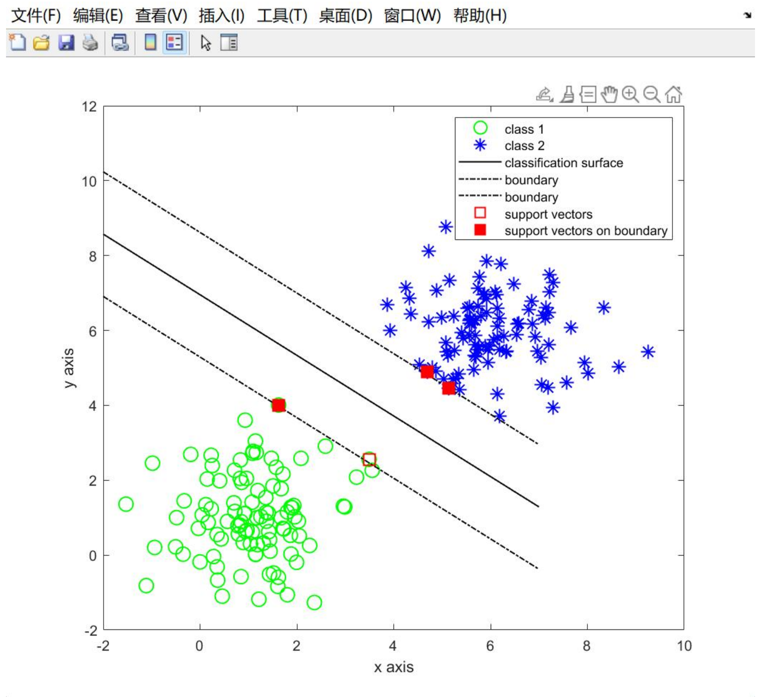
```
y1 = ( -b * ones(1,length(x1)) - w(1) * x1 )/w(2);
y2 = ( ones(1,length(x1)) - b * ones(1,length(x1)) - w(1) * x1 )/w(2);
y3 = ( -ones(1,length(x1)) - b * ones(1,length(x1)) - w(1) * x1 )/w(2);
figure(2)
set(gcf,'Position',[1,1,700,600], 'color','w')
set(gca,'Fontsize',18)
plot(X(1:n,1),X(1:n,2),'go','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(n+1:2*n,1),X(n+1:2*n,2),'b*','LineWidth',1,'MarkerSize',10);
hold on;
plot( x1,y1,'k','LineWidth',1,'MarkerSize',10);
hold on;
plot( x1,y2,'k-.','LineWidth',1,'MarkerSize',10);
hold on;
plot( x1,y3,'k-.','LineWidth',1,'MarkerSize',10);
hold on;
plot(X(alpha>0,1),X(alpha>0,2),'rs','LineWidth',1,'MarkerSize',10);    % 支持向量
hold on;
plot(X(alpha<C&alpha>0,1),X(alpha<C&alpha>0,2),'rs','MarkerFaceColor','r','LineWidth',1,'MarkerSize',10);    % 间隔边界上的支持向量
hold on;
xlabel('x axis');
ylabel('y axis');
set(gca,'Fontsize',10)
legend('class 1','class 2','classification surface','boundary','boundary','support vectors','support vectors on boundary');
```

六、展示对比

以下通过修改不同参数，调试模型。改变的参数分别为样本量 n、惩罚参数 beta（β）、惩罚参数 C、步长/学习率 eta（η）、迭代次数 m。经过调试，源代码中的参数是合理的。另外，中心点的设置将影响数据点是否线性可分。（alpha 和 lambda 作为迭代的参数，其初值只要设置合理即可，并无影响）

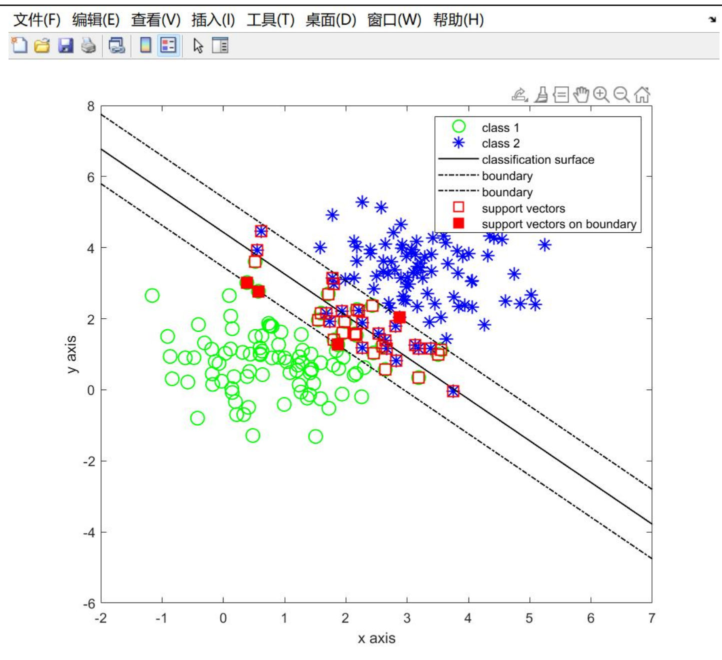
1、将中心点 2 设为（6，6），数据集基本线性可分

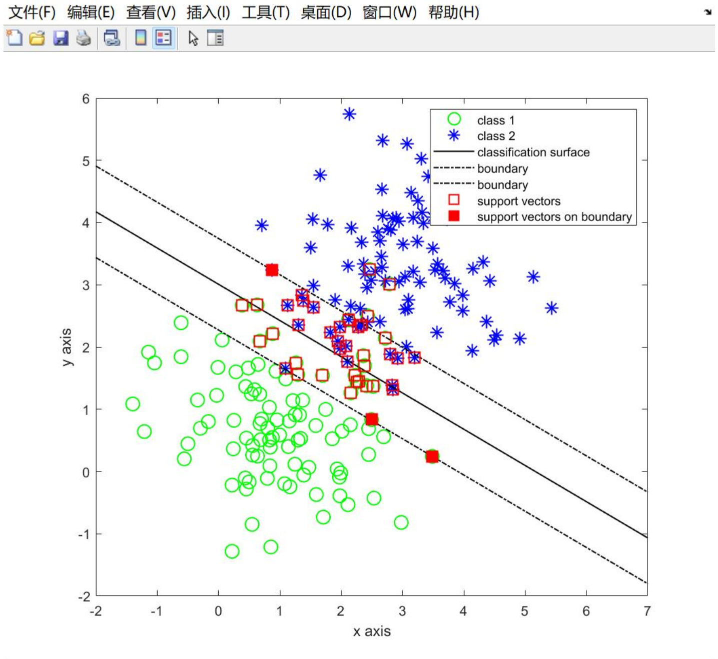
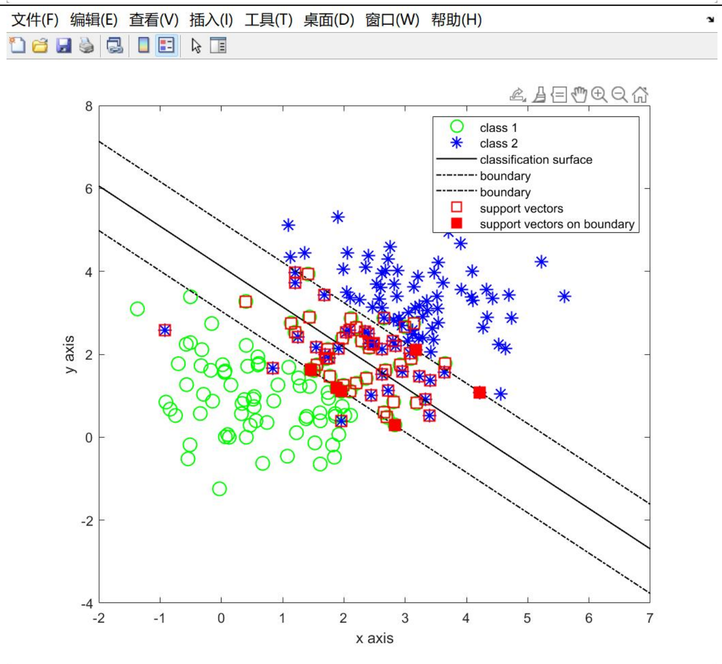
图三（测试）	线性增广拉格朗日函数	合页损失函数	耗时(s)

	0.313105495266735	1.0714322766946552	3.578271

线性可分支持向量机对参数的要求较低，不做过多展示，以下着重展现线性不可分支持向量机参数的调试过程。

2、将中心点 2 设为（3，3），数据集变得线性不可分，连续重复 3 次

次数	图三（测试）	线性增广拉格朗日函数	合页损失函数	耗时(s)
1		-10.2191	34.0718	3.311674

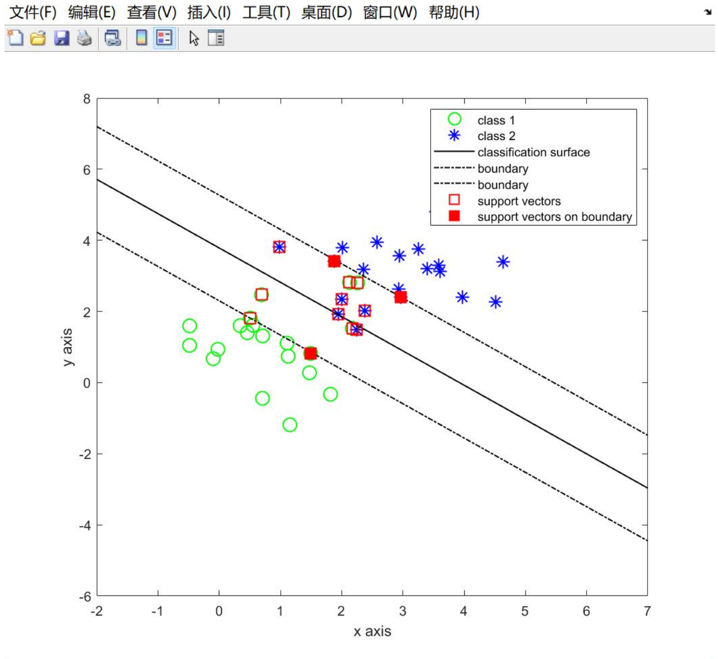
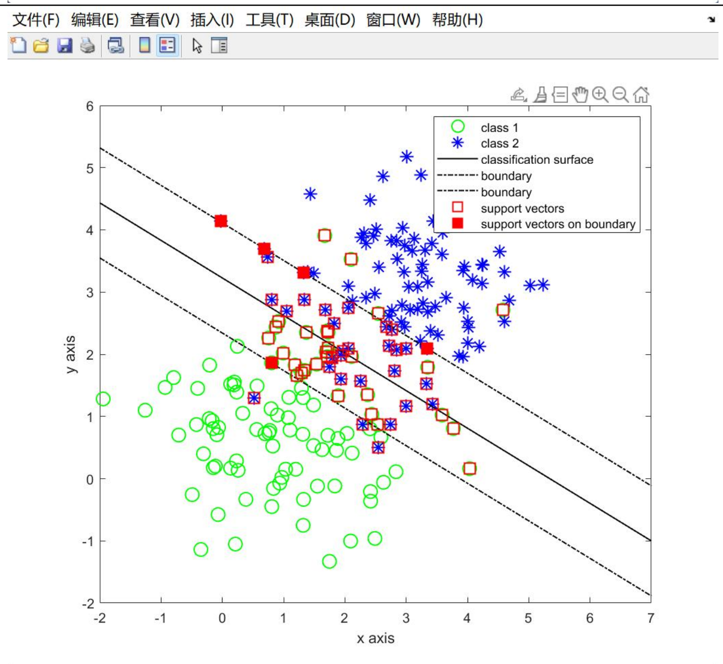
2		−11.3746	37.9191	3.553234
3		−18.4428	61.4939	3.833496

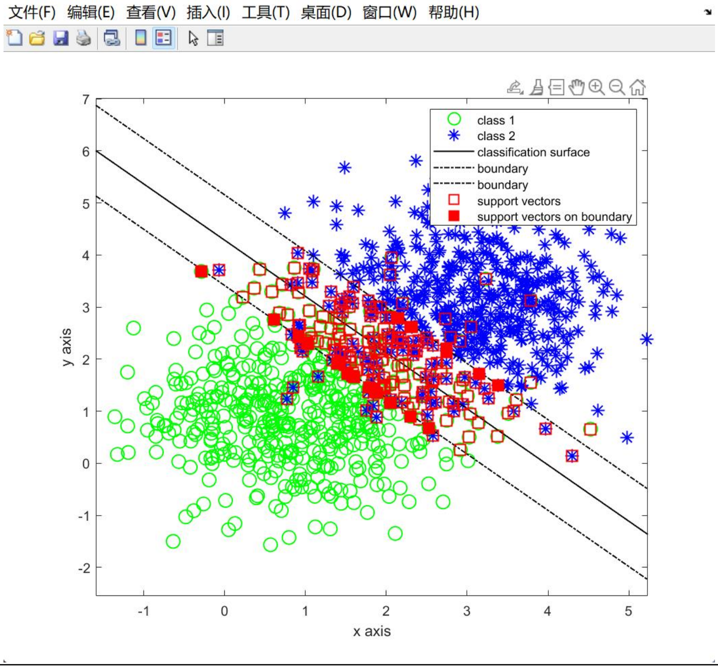
通过观察图像可以发现，SVM 模型稳定地较好地完成了分类任务，并完美地将支持向量尤其是间隔边界上的支持向量绘制了出来。

然而，通过计算发现，在一定范围内调节参数，并无法使每个支持向量计算出的 b 唯一，其总是在一段小的区间内，这说明在保守的参数设置（主要顾及运行时间）下，算法并没有完全收敛。

3、beta=0.1，C=0.3，eta=0.00005，m=100000，改动 n

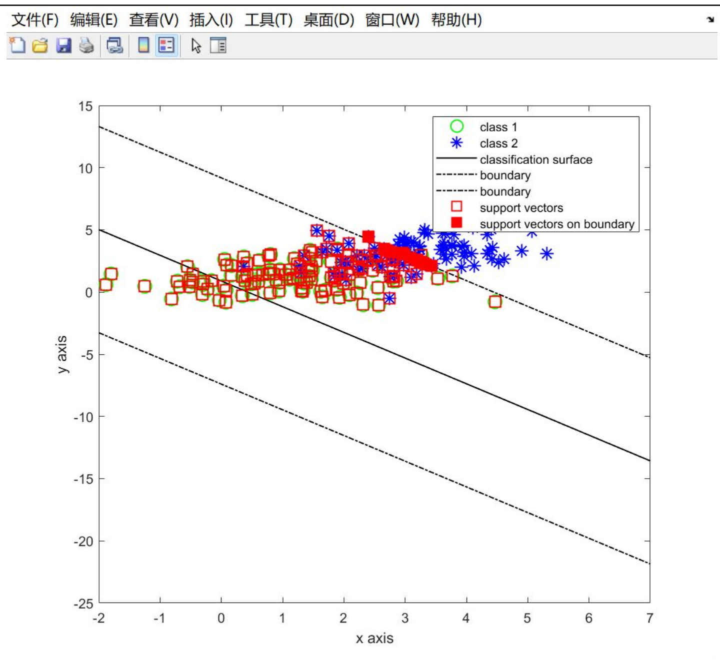
n	图三（测试）	线性增广拉格朗日函数	合页损失函数	耗时(s)

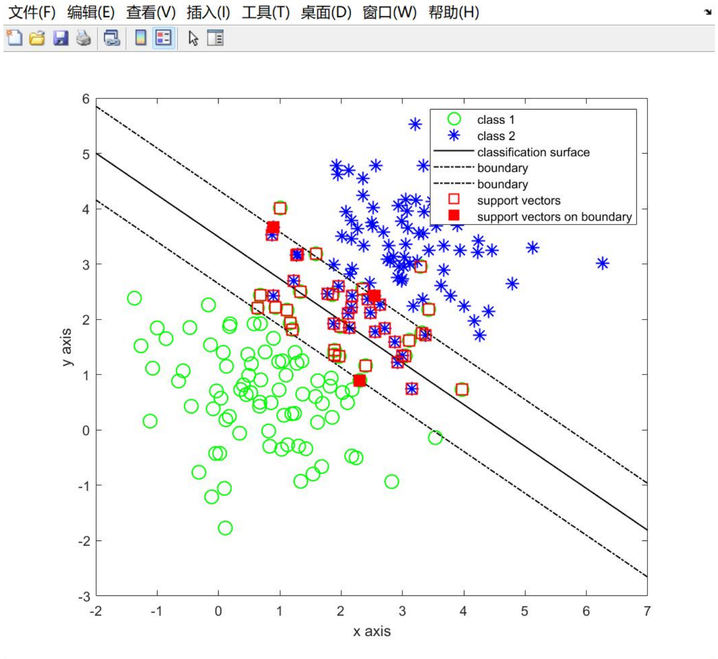
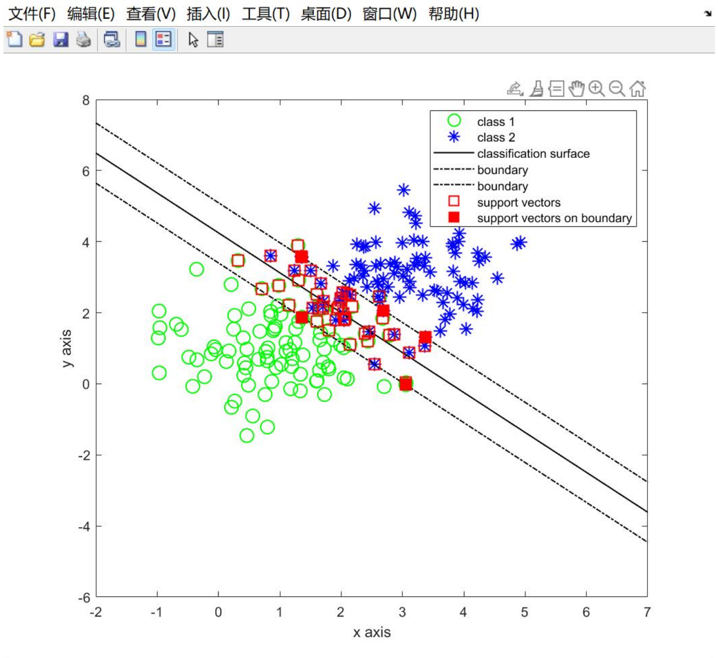
20		−3. 0597	10. 2272	0. 252857
100		−16. 4603	54. 8746	4. 114969

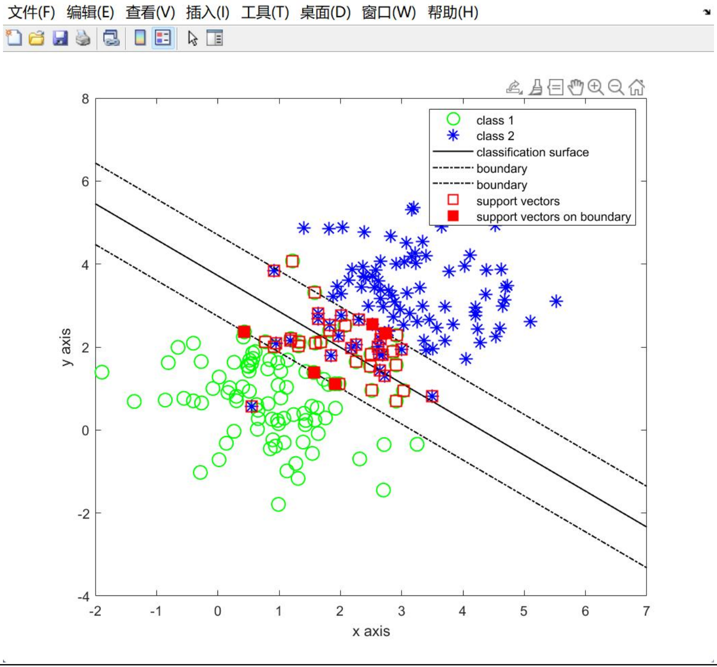
<table>
<tr><td>500</td><td>
</td><td>-60.2266</td><td>200.9428</td><td>190.624911</td></tr>
</table>

通过增加数据样本量可以发现，两个函数的结果绝对值与之正相关，时间上更是几何关系。虽然保持了良好的分类效果，并完美标注了支持向量，但是时间上却和前几个算法不在同一个量级上。

4、 n=100，C=0.3，eta=0.00005，m=100000，改动 beta

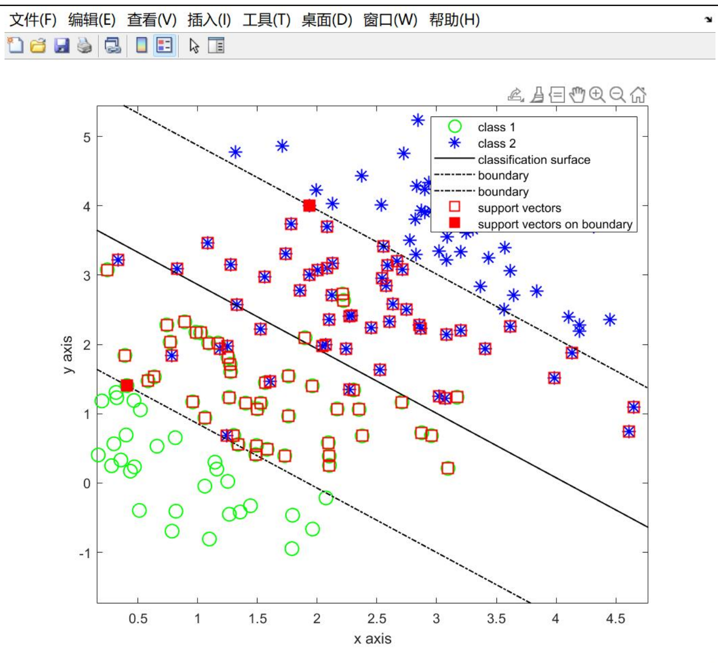
<table>
<tr><td>beta</td><td>图三（测试）</td><td>线性增广拉格朗日函数</td><td>合页损失函数</td><td>耗时(s)</td></tr>
<tr><td>0</td><td>
</td><td>-41.8952</td><td>139.2785</td><td>3.585339</td></tr>
</table>

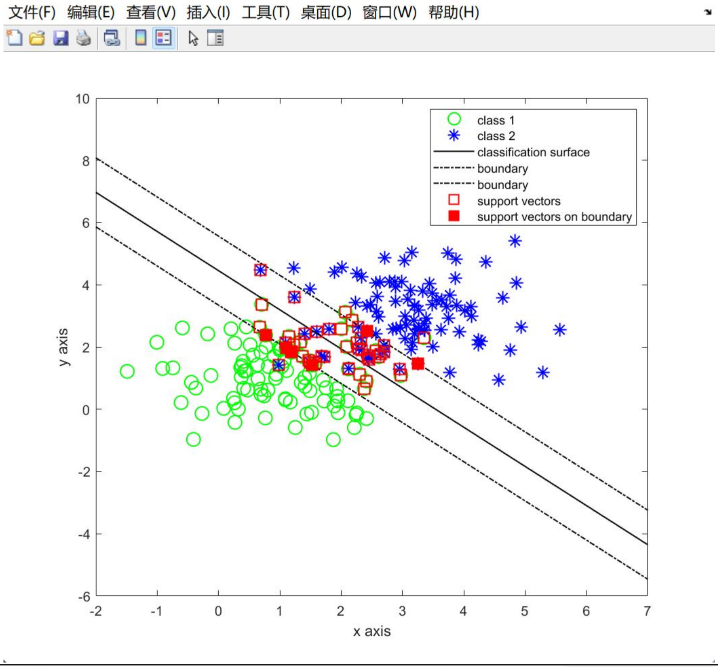
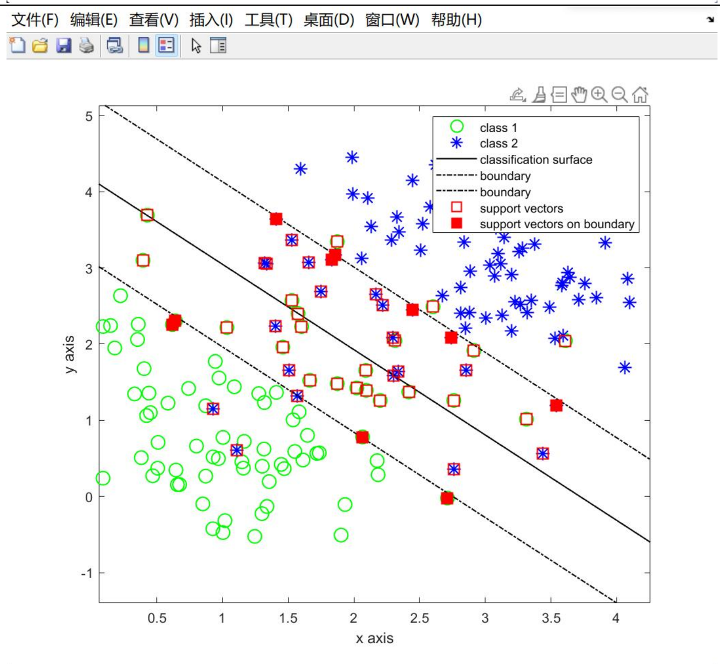
0.01		−12.4621	41.5426	3.539452
0.1		−12.0302	40.1209	3.379429

1		-12.4245	41.4220	3.187416

在惩罚参数 beta 等于 0 时，模型完全无法正常运行，这是由于在迭代中，以其为系数的项并未起到作用。而通过对比可以发现，在惩罚参数 beta 增大的过程中，两个函数值基本不发生变化，这说明在其他参数合适而该参数合理的情况下，其对结果影响不大，而运行时间与其反相关。

5、 n=100，beta=0.1，eta=0.00005，m=100000，改动 C

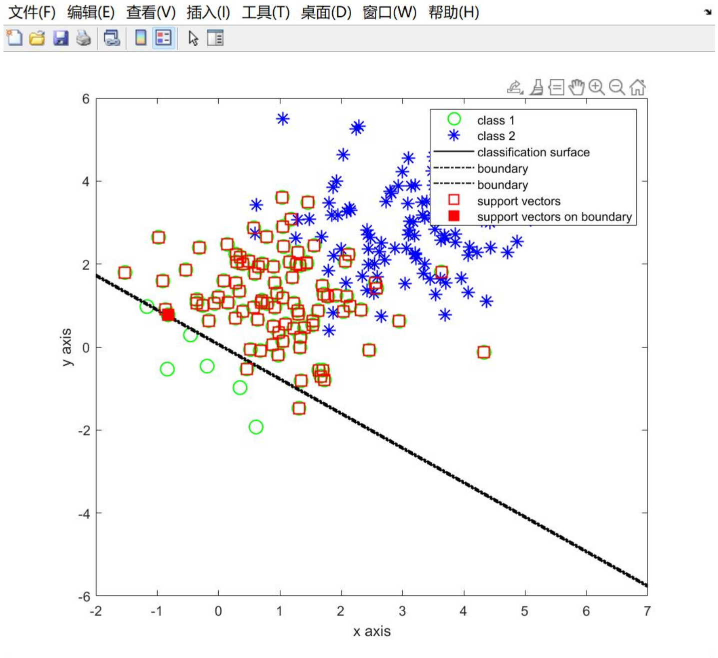
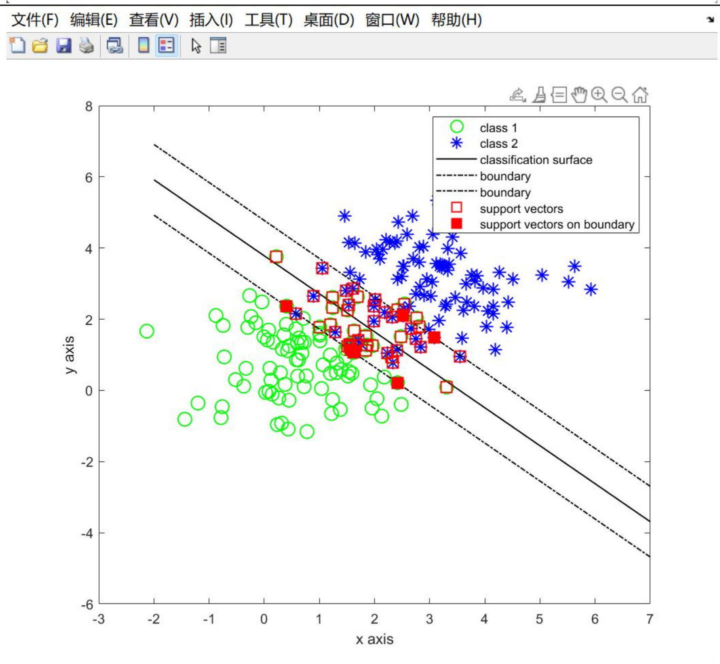
C	图三（测试）	线性增广拉格朗日函数	合页损失函数	耗时(s)
0		-0.7957	79.5680	3.347345

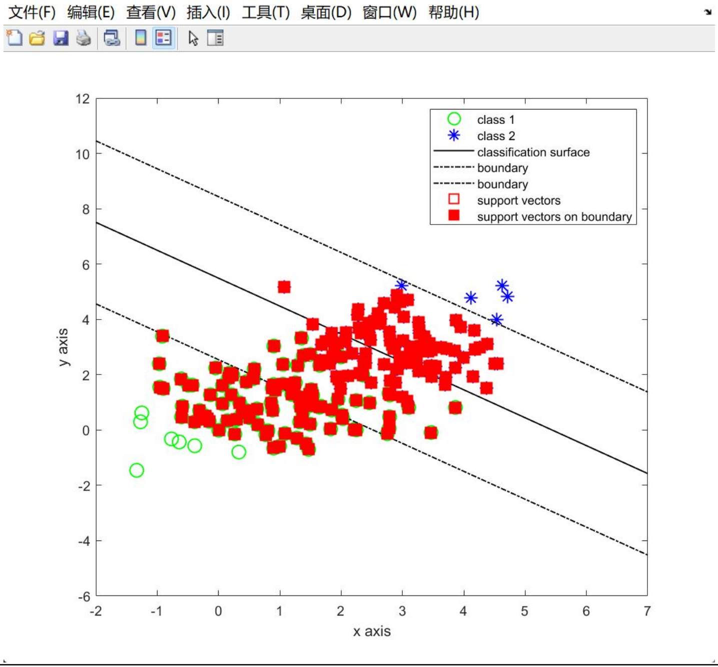
0.3		-12.9021	43.0161	3.385767
1		-42.7737	42.8846	3.659881

通过对比可以发现,在惩罚参数 C=0 时,没有支持向量,代码无法运行;在其过小(C=0.01)时，间隔边界上的支持向量没有全部标注出来；而在其过大（C=1）时，在间隔边界上的支持向量并不都恰好在间隔边界线上。这里特意将图像放大以便识别,通过两个函数的结果也可得出相应结论。

6、n=100，beta=0.1，C=0.3，m=100000，改动 eta

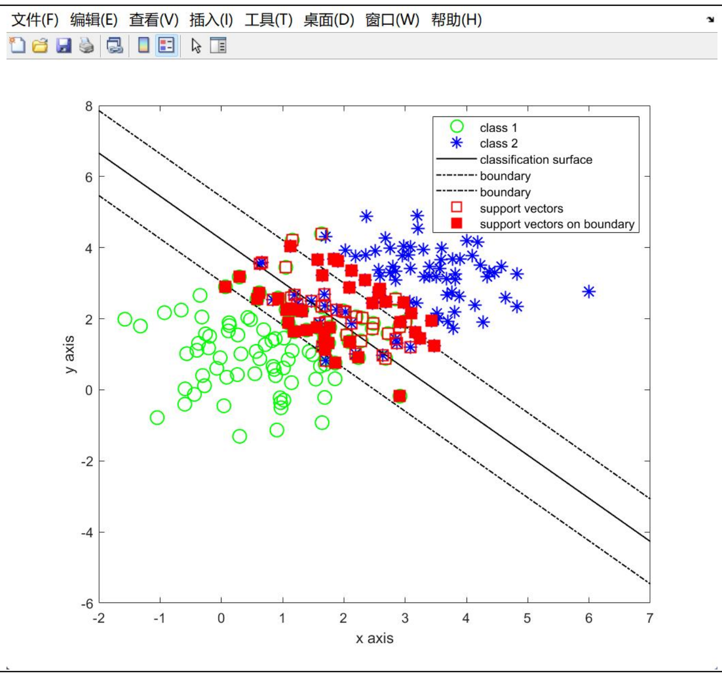
eta	图三（测试）	线性增广拉格朗日函数	合页损失函数	耗时(s)

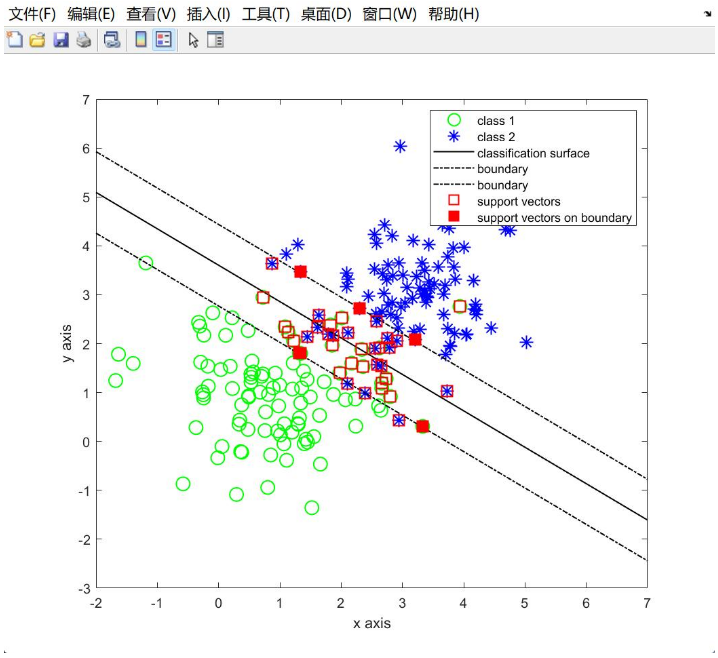
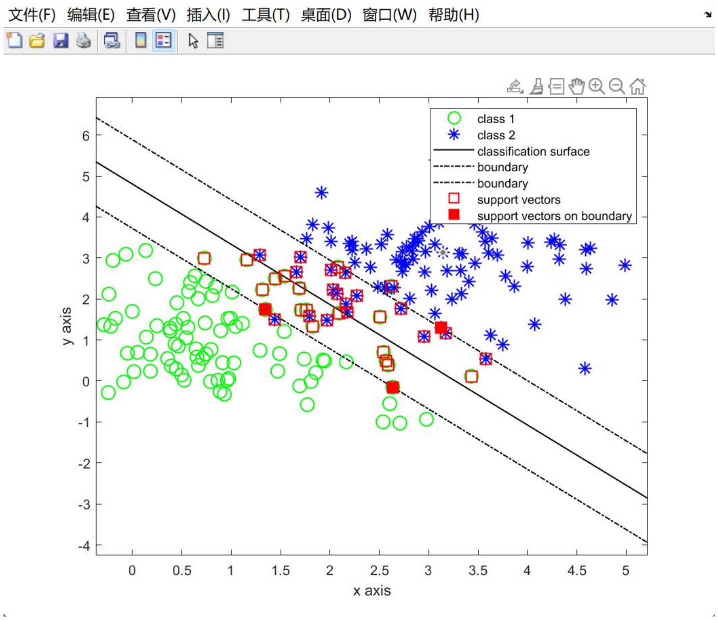
0.05		919.1785	2.1541e+04	3.331630
0.00005		−12.8623	42.9158	3.623175

0.000000005		-0.3940	91.6526	3.559501

步长和迭代次数是一对相对应的参数。在迭代次数一定的情况下，步长过大会跳过正确区间，步长过小会无法达到收敛。由此可以发现，为配合步长，在其变小时，迭代次数需增加，相应的，代码的运行时间也会大大增加。

7、n=100，beta=0.1，C=0.3，lambda=0.1，eta=0.00005，改动 m

m	图三（测试）	线性增广拉格朗日函数	合页损失函数	耗时(s)
10000		-16.5655	58.5875	0.342349

100000		−10.6522	35.5399	3.382214
1000000		−9.2968	30.9892	44.876205

在迭代次数不断增长的过程中，可以发现两个函数的绝对值在缩小，这表明结果在逐渐变好。而在迭代次数不够时，模型不能很好的呈现分类结果。相应的，运行时间也与迭代次数正相关。

8、 SVM 模型也适用于多维、多类别的分类，由于无法展示图像或需改动代码过多，这里不做说明。

七、分析

支持向量机模型适用于多类别分类，在应对线性可分、线性不可分的情况下都可以取得较好的结果。另外，通过核函数的方法，其还可以应用于非线性可分的情况。在处理高维的数据时，还具有较大优势。但同时，支持向量机也有很多问题，比如

1、 需求的迭代次数较多，运行时间较长，且随样本量的增加呈现几何倍数的增加。这一点在与前面几种算法的对比中十分明显。

八、附加题

1、见三、。
2、见六、4、6、。
笔记如下（截取了中国大百科全书第三版的相关内容，并加入了自己的理解）：

对等式约束非线性规划问题

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h_j(x) = 0, \ \ j = 1, 2, \cdots, p \end{aligned}$$

其增广拉格朗日函数(Augmented Lagrange function)定义为

$$L_\beta(x, \lambda) = f(x) + \sum_{j=1}^{p} (\lambda_j h_j(x) + \frac{\beta}{2}(h_j(x))^2)$$

式中$\lambda \in R^p$为拉格朗日乘子，$\beta > 0$为罚参数。而迭代方法

$$\begin{cases} x^{k+1} \in \operatorname{argmin}_x L_\beta(x, \lambda^k) \\ \lambda^{k+1} = \lambda^k + \beta h(x^{k+1}) \end{cases}$$

称为增广拉格朗日乘子方法。这一方法由M.R.赫斯泰尼斯(Magnus Rodolph Hestenes)和M.J.鲍威尔(Michael James David Powell)于1969年独立提出，它克服了罚函数方法罚参数需要趋向于无穷大而引起的数值困难，是著名优化软件LANCELOT的基础。R.T.洛克菲勒(Ralph Tyrrell Rockafellar)给出了不等式约束及一般约束情形下的增广拉格朗日乘子法，使该方法成为约束优化问题的一类基本方法。

理解：增广拉格朗日函数法是处理等式约束的非线性规划问题的方法。在运算时，拉格朗日函数由三部分组成，一部分是原函数 f(x)，一部分是约束条件 h_i 与对应朗格朗日乘子 λ_i 的积的和，最后一部分是约束条件 h_i 的平方与惩罚参数 β 的积。在这个式子中，体现了欲求解的函数和提供的约束条件，并引入了非无穷大的惩罚参数。
3、见六、5、。