# 模式识别第二次大作业
# 线性分类器


院(系)名称　高等理工学院

学　　号　16231235

姓　　名　李谨杰

指导教师　郭玉柱


2019 年 6 月

## 一、实验简介

线性感知器是两类分类器最简单的学习算法之一。给出一组 D 维数据点，属于 ω1 和 ω2 两类，该算法试图在这两类样本之间找到一个线性分类超平面。如果样本是一维、二维或三维的，分离超平面将分别是点、线或平面。我们研究的具体算法是一类算法的一个特例，该类算法使用一个精心定义的目标函数上的梯度下降来得到一个解。

## 二、实验原理

假设这两类样本在特征空间中是线性可分的。也就是说，存在一个平面 $G(X)=W^{T}X+w_{n+1}=0$，其中 $W\in R^{n}$ 和 $X\in R^{n}$，使得属于第一类的所有样品都在平面的一侧，而第二类的所有样品都在另一侧。如果存在这样的平面，感知器算法的目标是在给定数据点的情况下学习任何一个这样的平面。一旦学习完成并确定了平面，将来很容易对新点进行分类，因为平面一侧的点会导致 $G(X)=W^{T}X+w_{n+1}$ 的正值，而另一侧的点会给出负值。

根据感知器学习原理，权重向量 $W\in R^{n}$ 可以扩展到 $\widehat{W}=\left(w_1w_2\cdots w_nw_{n+1}\right)^{T}\in R^{n+1}$，特征向量也可以扩展到 $\widehat{X}=\left(x_1x_2\cdots x_n1\right)^{T}\in R^{n+1}$，因此分类平面可以表示为 $G(X)=\widehat{W}^{T}\widehat{X}=0$。更新的学习规则（算法）权重设计为

$$\widehat{\mathbf{W}}(\mathbf{t}+\mathbf{1})=\widehat{\mathbf{W}}(\mathbf{t})+\frac{1}{2}\boldsymbol{\eta}\left\{\widehat{\mathbf{X}}(\mathbf{t})-\widehat{\mathbf{X}}(\mathbf{t})\operatorname{sgn}\left[\widehat{\mathbf{W}}^{T}(\mathbf{t})\widehat{\mathbf{X}}(\mathbf{t})\right]\right\}$$

$$=\begin{cases}\mathbf{W}(\mathbf{t}) & \hat{W}^{T}(t)\widehat{X}(t)>0\\ \mathbf{W}(\mathbf{t})+\boldsymbol{\eta}\widehat{\mathbf{X}}(\mathbf{t}) & \text{otherwise}\end{cases}$$

式中 $\eta$ 为学习速度，可适当调整以提高收敛效率。

## 三、实验目标

（1）了解线性感知器学习算法的工作原理。
（2）了解各种参数对学习速度的影响，以及算法的收敛性。
（3）了解数据分布对算法学习能力的影响。

## 四、实验内容

步骤 1：

（1）根据上述原则和理论，设计和编写面向两类线性学习的感知器程序代码。
（2）创建一个线性可分数据集，两类中的每个数据集均有 50 个以上的样本。
（3）初始化权重向量 $\widehat{W}(0)$ 并为学习率 $\eta$ 选择适当的值 $\eta\in(0,1]$。
（4）运行程序，使用你的数据集，记录结果。关注收敛迭代次数。

步骤 2：

对于不同的数据集，这些数据集的两类模式的数量彼此不同，重复上述（2）~（4）步骤。记录你的观察。

步骤 3：

研究不同可分离性程度下学习速率的变化对学习效果的影响。

步骤 4：

探索上一节中的问题并设计实验来回答这些问题。完成并提交所有实验结果的实验报告，并进行比较分析，总结本实验研究的经验。

## 五、实验结果及分析

为提高效率，我采用固定增量法找到目标向量。固定增量法的算法步骤是：

（1） 任意选择初始的权向量 w（0），置 t=0。

（2） 考查样本 $\widehat{X}(t)$，若 $\hat{W}^{T}(t)\widehat{X}(t) \le b$，则 $\mathbf{W(t+1)=W(t)+\eta\widehat{X}(t)}$，否则继续。b 是一个正实数，为设定的余量。

（3） 考查另一个样本，重复（2），直至处理完全部样本。

（4） 检查是否对所有样本都有 $\hat{W}^{T}(t)\widehat{X}(t) > b$，即损失函数 $J(w)=0$。如果有不满足的情况，重复（2），（3），直到样本全部满足或达到迭代上限。

根据《模式识别》（张学工）上的内容可知，可以证明，对于线性可分的样本集，采用这种梯度下降的迭代算法，经过有限次修正后一定会收敛到一个解向量。记录实验结果如下。

1. 分类结果如图 5.1。

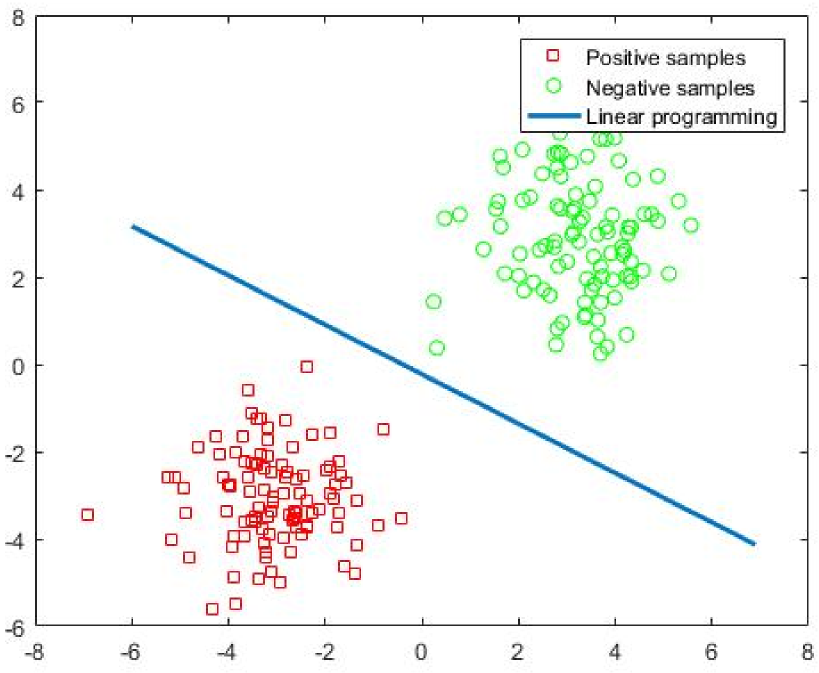


图 5.1 样本分布及某次训练结果

初始化 W（0）=[1,1,1], 更改学习率 $\eta$，记录迭代次数如下：

| 学习率$\eta$ | 0.05 | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 迭代次数 t | 38 | 19 | 9 | 7 | 4 | 3 | 3 | 2 | 2 | 2 |

注：只有 w 值经过改变，才记录迭代次数加一。

作图如图 5.2。

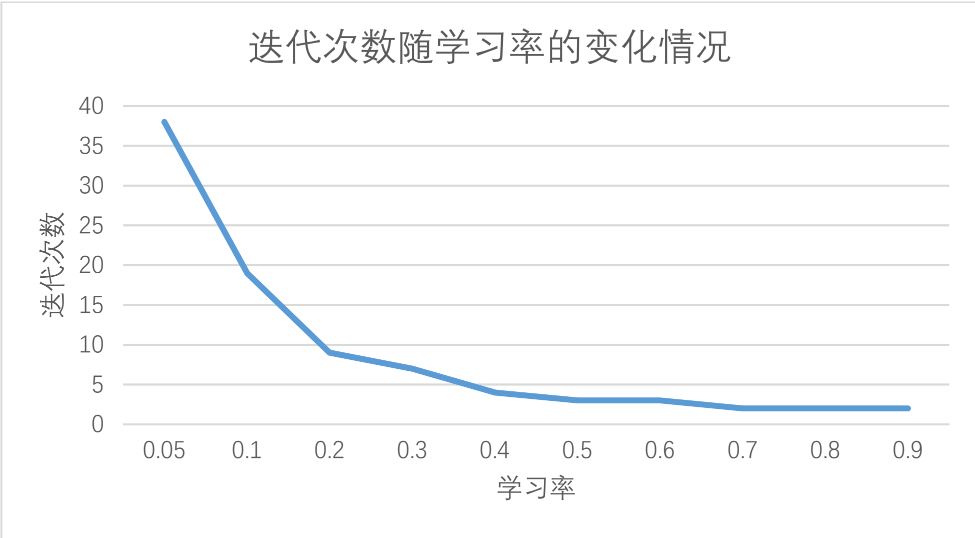


图 5.2 迭代次数随学习率的变化情况

观察结果可知，对于该数据集，学习率越大，迭代次数越少，学习速度越快。

2. 更改两种数据集的数目，随机生成数据集，线性分类结果如下：

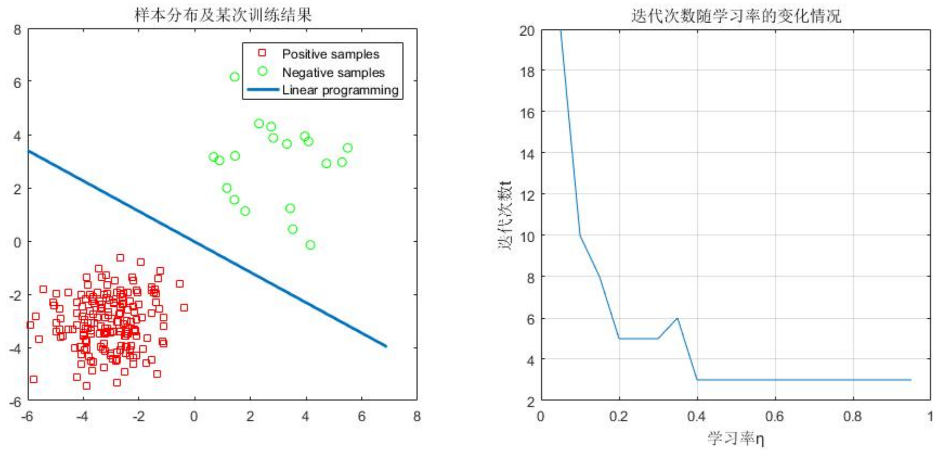


图 5.3 180 个正分类，20 个负分类

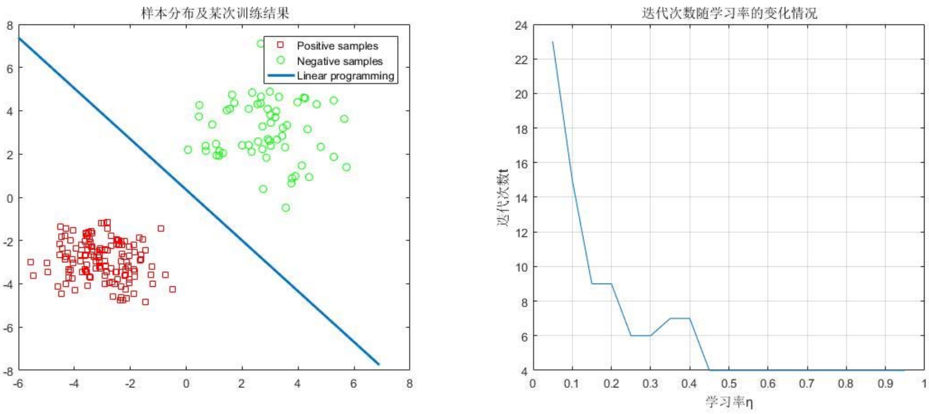


图 5.4 140 个正分类，60 个负分类

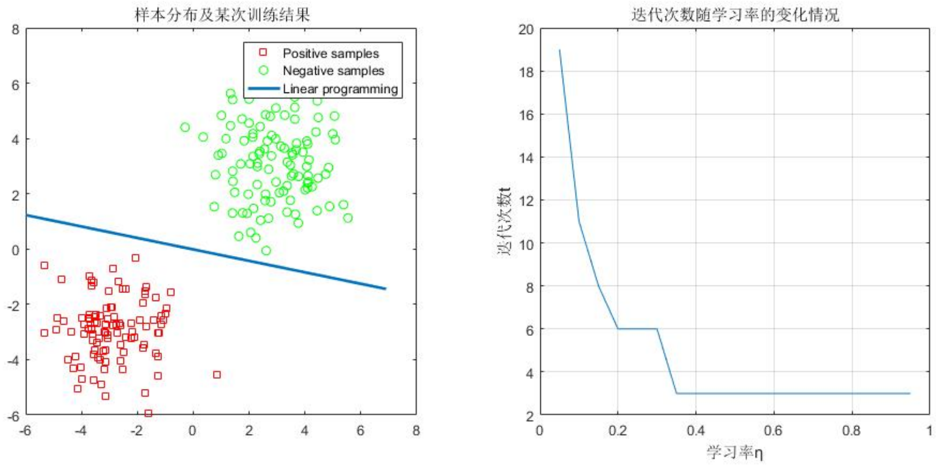


图 5.5 100 个正分类，100 个负分类

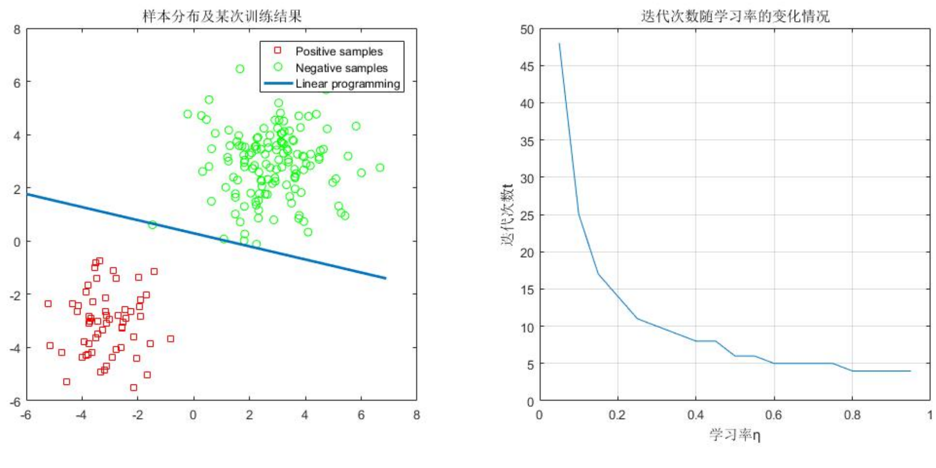


图 5.6 60 个正分类，140 个负分类

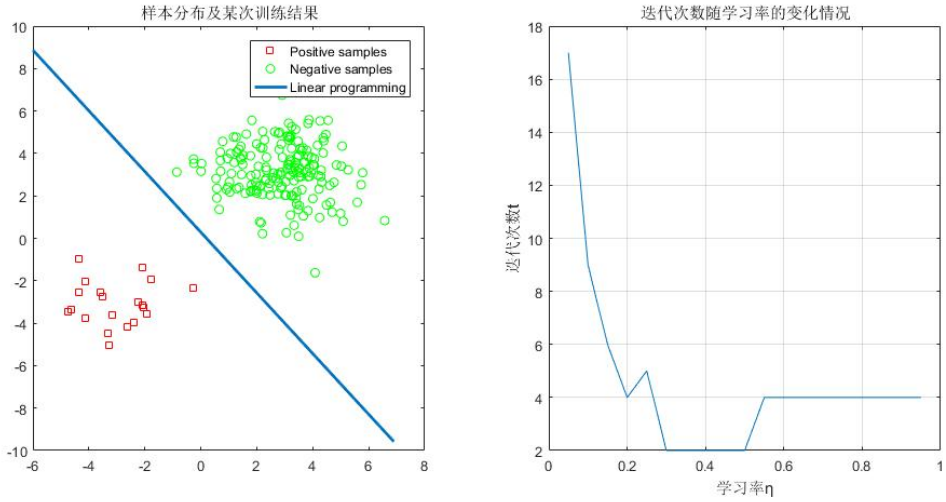


图 5.7 20 个正分类，180 个负分类

观察实验结果有如下特点：

（1） 并非学习率越高收敛效果越快。在一定范围内，学习率越高，收敛速度越快，但当学习率过大后，可能出现震荡现象，导致迭代次数增加。因此，应根据数据集选择合适的学习速率。

（2） 正负类别数量相近的线性可分数据，与数量差异较大的数据比较，其整体迭代次数较多。

3. 研究不同可分离性程度下学习速率的变化对学习效果的影响。

我将数据的分离程度理解为，两个数据是否分得开。在程序中用随机产生数据的半径系数来表示，半径系数越大，表示重叠程度越高。

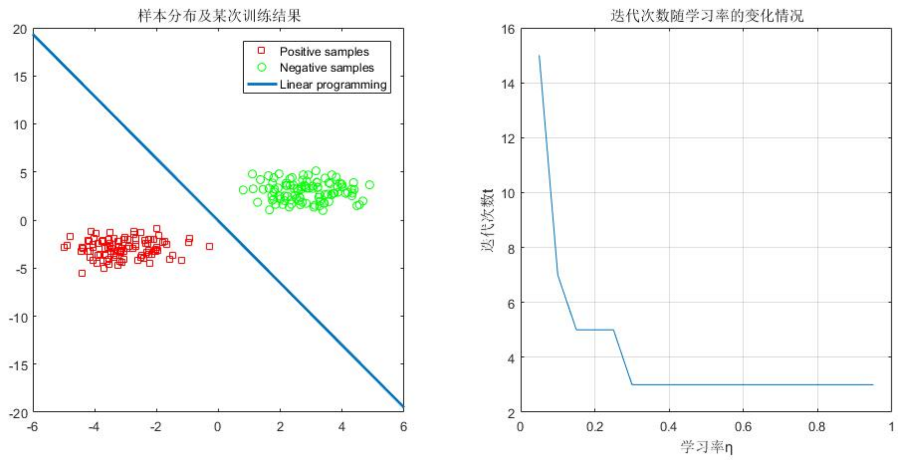


图 5.8 半径系数为 1

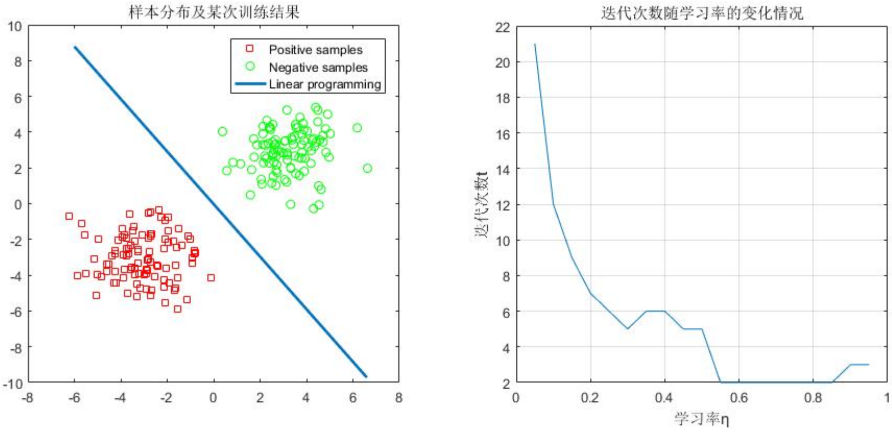


图 5.9 半径系数为 1.2

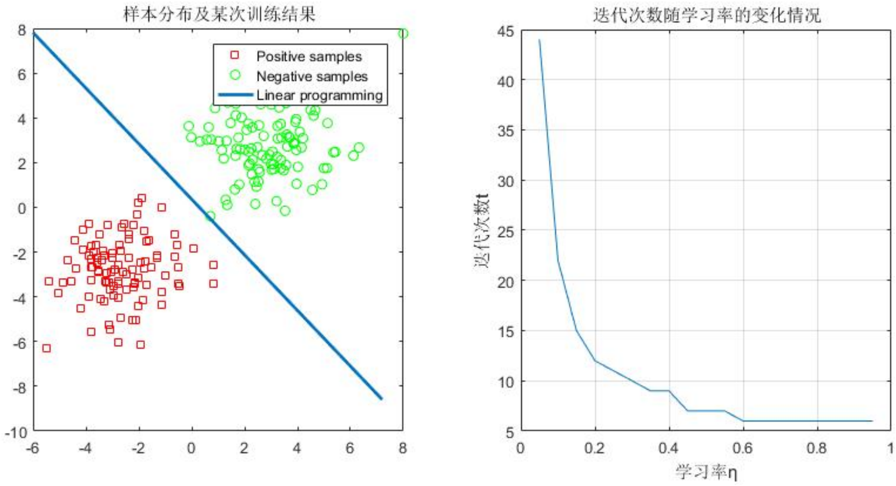


图 5.10 半径系数为 1.4

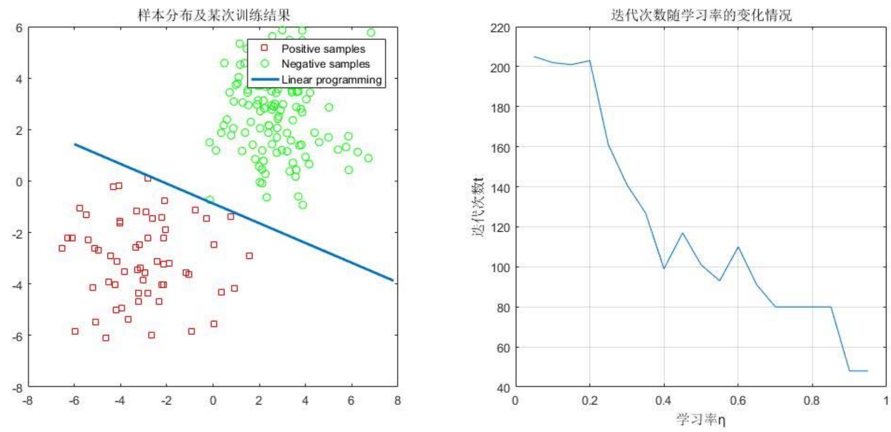


图 5.11 半径系数为 1.6，设定的迭代上限为 200 次

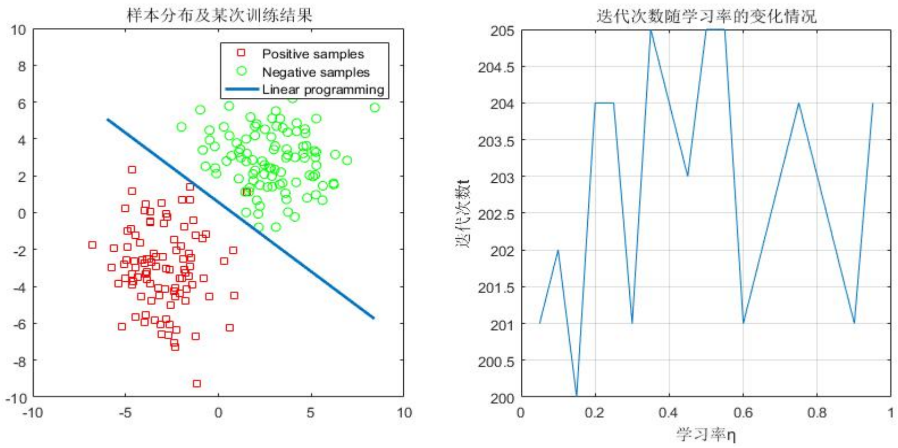


图 5.12 半径系数为 1.8，设定的迭代上限为 200 次

根据实验结果，对于线性可分的数据（图 5.8-图 5.11），学习率相同时，数据的可分离性越低，需要的训练次数就越多，最终稳定的迭代次数也越高。对于半径系数 1.6 这样的数据集，由于边界十分极限，当学习率小于 0.2 时难以在 200 次以内达到收敛，如图 5.11，在这种情况下需要增大学习率。

对于线性不可分的样本，如图 5.12，全部学习率迭代次数均达到上限，按照固定增量法不能确保找到最好的分类边界。

## 六、实验源代码

```
clc;clear;close all;
%% 生成随机数据
shift = 3; %中心的差距
n = 60; %类1样本数
totel=200;
m= totel-n;   %类2样本数
d = 2;  %二维, d=2
sigma = 1.7; %第二类的分散程度
x = randn(d,n)*sigma-shift;
y = randn(d,m)*sigma+shift;
save('x','x');save('y','y');
load('x');load('y');
x(3,:)=1;y(3,:)=1; %规范化增广样本向量
%% 显示生成的数据
figure(1);
subplot(1,2,1);
plot(x(1,:),x(2,:),'rs');   %第一行为行坐标, 第二行为列坐标
hold on;
plot(y(1,:),y(2,:),'go');
legend('Positive samples','Negative samples');title('样本分布及某次训练结果');
y=-y; %规范化增广样本
%% 采用固定增量法
%目标: 对全部样本 w*x(:,i)>b
b=0.4; %余量, 把解区向中间缩小
w(1,:)=[1,1,1];
Eta=0.9;
t_max=200;
tag=1;
for Eta=0.05:0.05:0.95
    t=1;
    J=sum(w(t,:)*y<=b)+sum(w(t,:)*x<=b); %计算损失函数
    while J~=0 && t<=t_max  %如果没有将全部数据分类正确
        for i=1:n
            while w(t,:)*x(:,i)<=b   %处理每个变量
                w(t+1,:)=w(t,:)+Eta*x(:,i)';
                t=t+1;
```

```
        end
    end

    for i=1:m
        while
w(t,:)*y(:,i)<=b  %处理每个变量

w(t+1,:)=w(t,:)+Eta*y(:,i)';
            t=t+1;
        end
    end

J=sum(w(t,:)*y<=b)+sum(w(t,:)*x<
=b);
  end
  result(tag)=t-1;tag=tag+1;
end
%% 作图
    x1 = -shift-
3:0.1:shift+3*sigma;
    y1 = (-w(t,3)-
w(t,1)*x1)/w(t,2);
    plot(x1,y1,'-
','LineWidth',2);
    disp(t-1);
    legend('Positive
samples','Negative
samples','Linear programming');

   subplot(1,2,2);
    plot(0.05:0.05:0.95,result);
    xlabel('学习率η'),ylabel('迭代
次数t'),title('迭代次数随学习率的变化
情况');
    grid on;
```

## 七、收获、体会及建议

通过这次实验，我深刻理解了线性感知器学习算法的工作原理，也体会到不同参数对学习率和系统收敛性的效果，还理解了数据分布对于算法的可学习性的影响。如果老师对报告有任何问题，可以通过 Email：lijinjie@buaa.edu.cn 联系我，谢谢老师！