山东大学 <u>计算机科学与技术</u> 学院

<u>机器学习</u> 课程实验报告

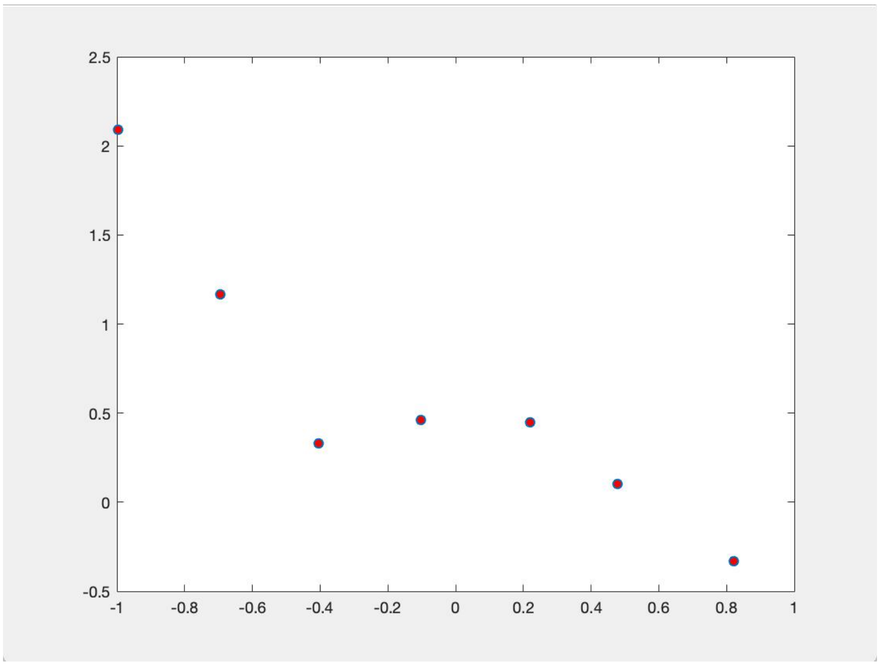
学 号 ：201700301042	姓名：陈佳睿	班级：17 智能
实验题目：正则化		

实验过程中遇到和解决的问题：

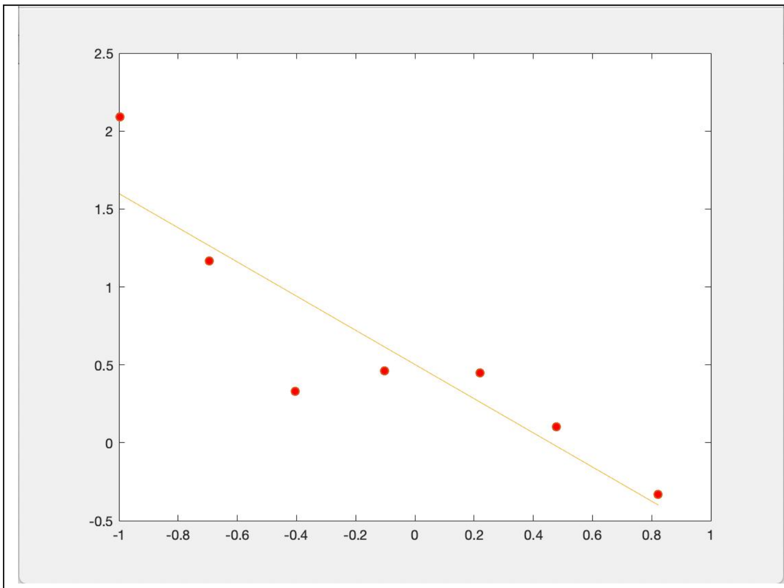
（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

一、正则化的线性回归

本次实验包括两个集合的数据，一个用于线性回归，另外一个用于罗辑回归
实验指导书还提供了一个辅助函数 map_feature.m 用于逻辑回归
正则化线性回归
输入数据 x 是一个独立特征 绘出的 y 可以看作是 x 在二维平面中的函数
使用 matlab 加载数据如下图所示



可以看出只用一条直线去拟合数据太过简单了,效果也很差



所以我们使用一个高阶多项式来更接近数据点
这里我们尝试五次多项式，所以等于我们假设了六个特征
我们要使用数据学习的分别是 x^0，x^1,......x^5 的系数

$$h_\theta(x)=\theta_0+\theta_1 x+\theta_2 x^2+\theta_3 x^3+\theta_4 x^4+\theta_5 x^5=\sum_{i=0}^{5}\theta_i x^i$$

尽管我们在使用多项式拟合数据
但是我们还是会在线性回归中出现问题
因为假设在每一个特征空间中是线性的
使用五次多项式对一个只有 7 个数据点的数据集进行拟合
不可避免的会导致过拟合问题
为了防止这一问题的出现，我们在模型中使用正则化来改善
在正则化问题中，我们的目标通过学习 theta 来最小化残差函数：

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{n}\theta_j^2\right]$$

上式中的 λ 就是正则化参数 正则化参数 λ 是对于我拟合参数的控制变量

如果随着我拟合参数的增大（即对于训练数据的拟合程度增大）

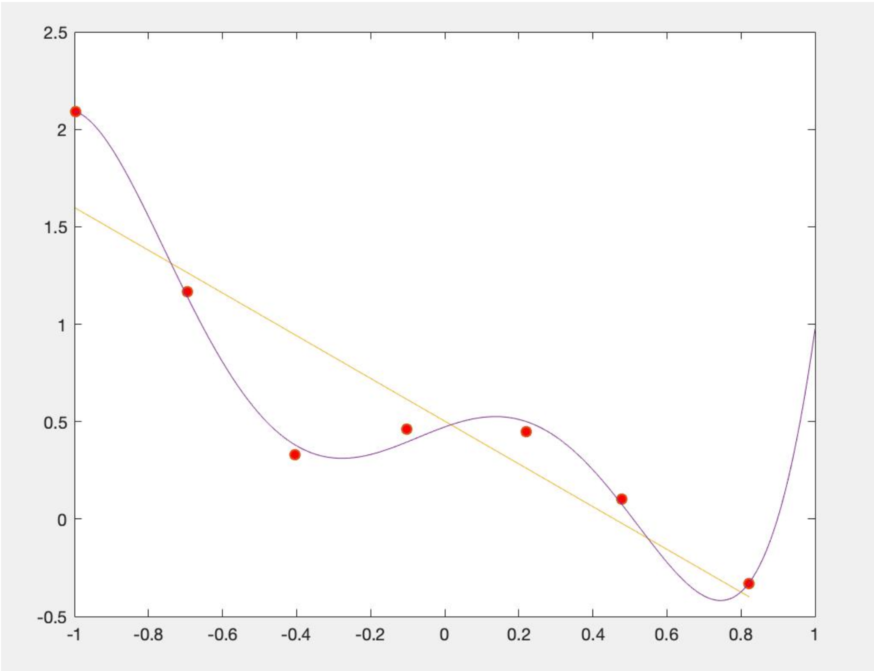
因为损失函数 j（theta）中存在 $\lambda\sum_{j=1}^{n}\theta_j^2$ -这一惩罚项，可以帮助我们减轻过拟合的现象
实验指导书提示我们：

notice that the summation after λ does not include （θ0）^2

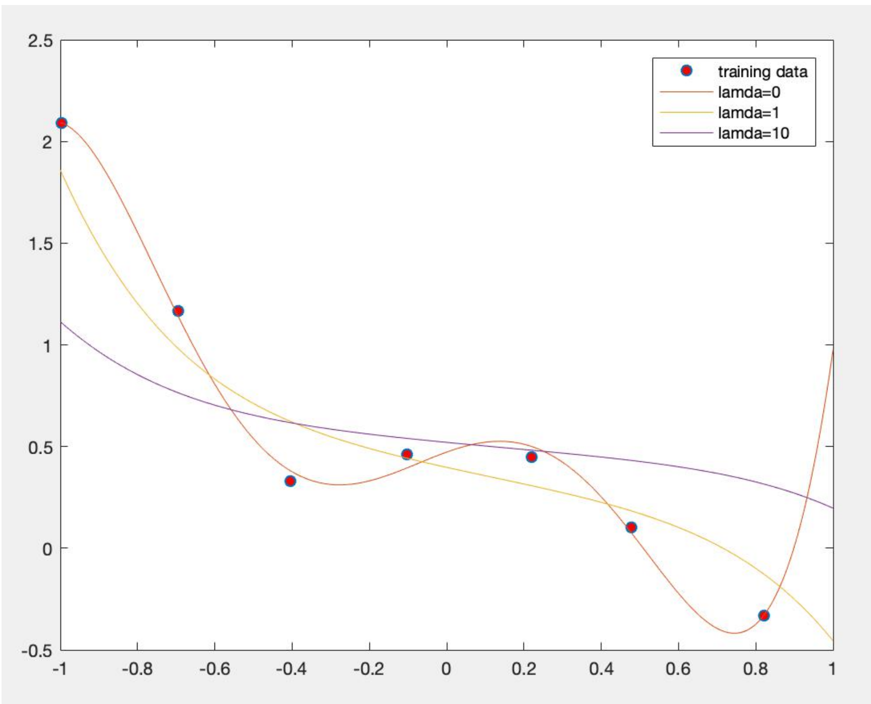
那么这样我们就能找到对于模型拟合程度最好的标准方程

$$\theta = (X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix})^{-1} X^T \vec{y}$$

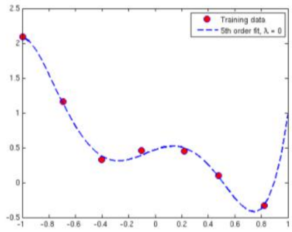
Lamda = 0 时的结果：

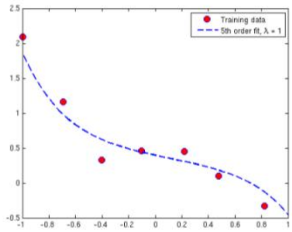


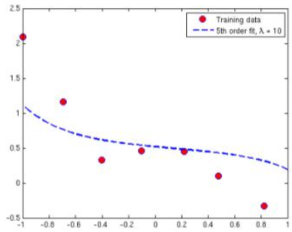
Lamda = 1 和 Lamda = 10 的结果



结论：可以看出，随着 lamba 的值的增大，拟合曲线越来越平滑，但是也并非越平滑越符合训练数据的形式，相比 lamba=1 和 lamda=10，在这 7 个数据的训练集上，lamda=0 的情况要更加符合数据的表达。
当然，这也并不代表在测试集上模型会有很好的表现
所以调参也存在一定的运气成分，当然一个理想的可解释的机器学习模型才是我们调参数的基础
对比实验指导书所给出的结果图，可以认为基本与其预期效果一致

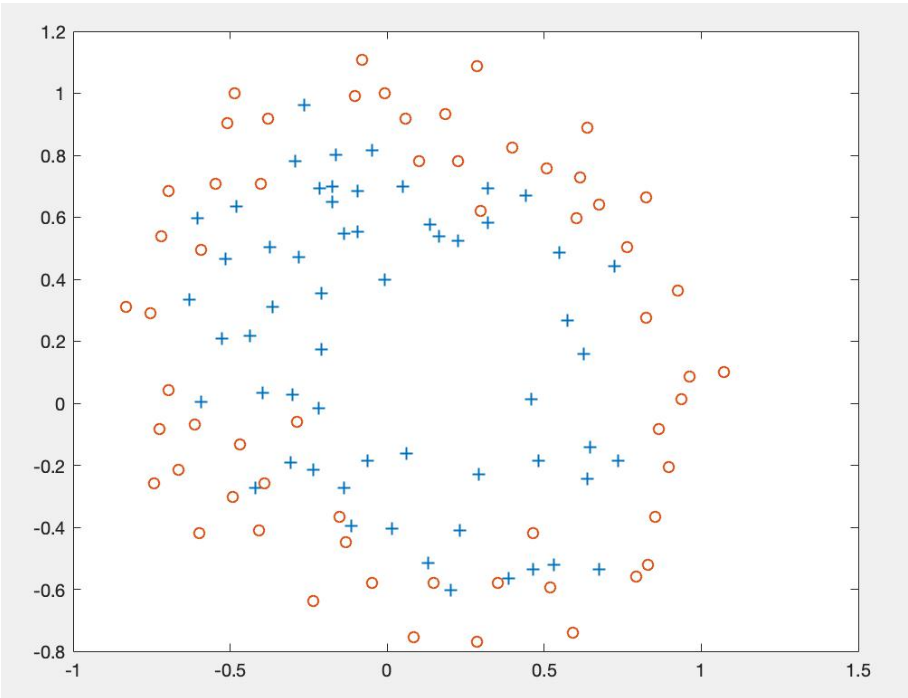






二、正则化的逻辑回归

实验的第二部分是要我们使用牛顿法来实现正则化的逻辑回归
首先，在 matlab 中加载数据
得到的效果如下图所示：



在逻辑回归中，我们使用的假设函数：
将任意输入值限制在[0,1]之间

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}} = P(y=1|x;\theta)$$

加入正则项后，我们的损失函数表示如下：

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

我们将 x 赋值为 u 和 v 的所有单项式(即多项式项)直到六次幂
即将我们的数据特征映射为 27 个维度
再加上一个阈值项一共 28 个维度作为数据的特征

$$x = \begin{bmatrix} 1 \\ u \\ v \\ u^2 \\ uv \\ v^2 \\ u^3 \\ \vdots \\ uv^5 \\ v^6 \end{bmatrix}$$

这里我们使用实验指导书中给出的 map_features 函数
来进行向高维度的特征映射
再在实现的公式中加入正则项
那么新的目标函数形式如下：

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y_i\log\left(h_\theta\left(x_i\right)\right) + \left(1-y_i\right)\log\left(1-h_\theta\left(x_i\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

因为我们无法得到这个问题的解析解
所以我们使用梯度下降的方法来进行求解

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla_\theta J$$

目标函数对 theta 进行求导数得到如下的表达

$$\nabla_\theta J = \begin{bmatrix} \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_0^{(i)} \\ \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_1^{(i)} + \frac{\lambda}{m}\theta_1 \\ \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_2^{(i)} + \frac{\lambda}{m}\theta_2 \\ \vdots \\ \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)x_n^{(i)} + \frac{\lambda}{m}\theta_n \end{bmatrix}$$

Hessian 矩阵形式

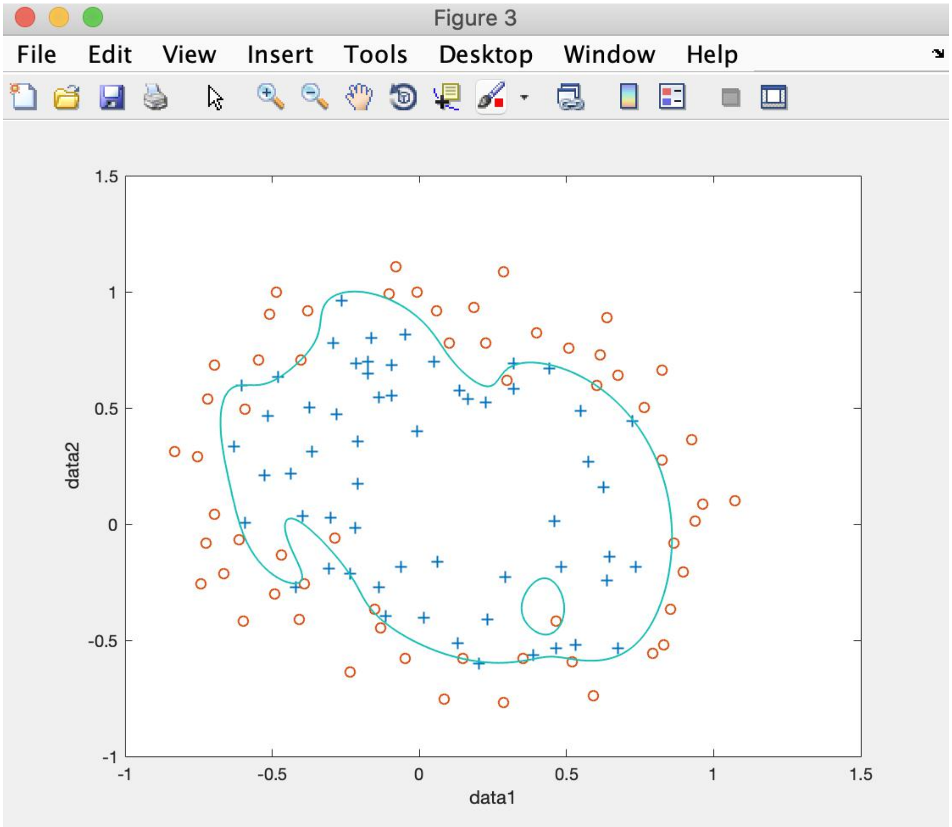
$$H=\frac{1}{m}\left[\sum_{i=1}^{m}h_\theta(x^{(i)})\left(1-h_\theta(x^{(i)})\right)x^{(i)}\left(x^{(i)}\right)^T\right]+\frac{\lambda}{m}\begin{bmatrix}0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1\end{bmatrix}$$

将梯度变形：

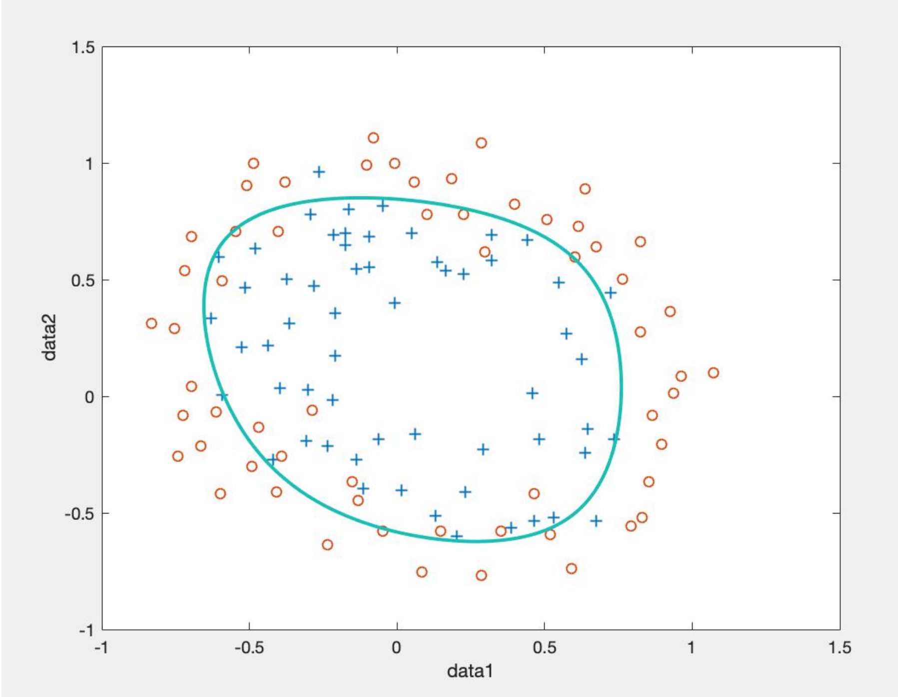
$$\nabla_\theta J=\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x_i\right)-y_i\right)x_i+\begin{bmatrix}0 & & & \\ & 1 & & \\ & & \dots & \\ & & & 1\end{bmatrix}\theta$$

3.实验结果分析

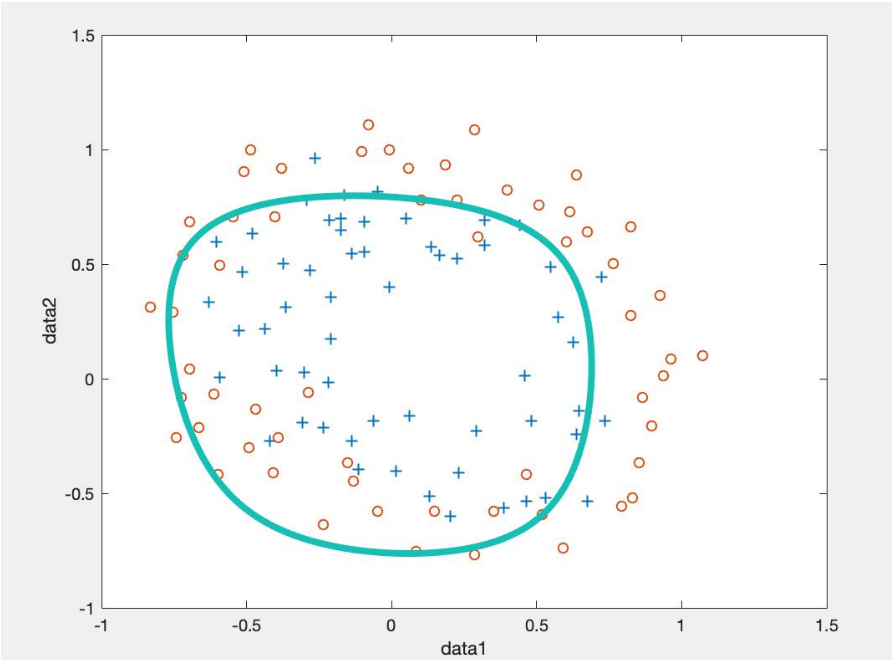
1) lamda = 0



2) lamda = 1


1.5
1
0.5
0
-0.5
-1
data2
-1
-0.5
0
0.5
1
1.5
data1

3) lamda=10


1.5
1
0.5
0
-0.5
-1
data2
-1
-0.5
0
0.5
1
1.5
data1

结果分析与体会：通过这次实验，通过对线性回归和逻辑回归使用正则化，可以看到借助在目标函数加上不同表示的正则化项（惩罚项）可以很好地达到避免过拟合的效果。

代码：

```
x = load ('ex5Logx.dat');

y = load ('ex5Logy.dat');

figure



% Find the indices for the 2 classes



pos = find(y);

neg = find(y == 0);

plot(x(pos, 1), x(pos, 2), '+');

hold on

plot(x(neg, 1), x(neg, 2),'o');

xlabel('data1');

ylabel('data2');





I=eye(28);
```

```
I(1,1)=0;

m=length(y);

%å¾—å ̂°28ç»´çš„Xç”¨äºŽè®¡ç®—boundary



X=map_feature(x(:,1),x(:,2));

%% regulization

% è®¾ç½®æœ€å¤§å¾ªçŽ¯æ¬¡æ•°

num=500;

% %% lamda=0          %% 1 10

% lamda=0;

% theta=zeros(28,1);

% for j=1:num

%      % hæ˜¯117*1ç»´

%      h=1./(1+exp(-X*theta));

%      % X=117*28ç»´ theta=28*1 Xâ€˜=28*117ç»´

%      H=(1/m)*(X'*(h.*(1-h).*X)+lamda*I);

%      invH=inv(H);

%      theta=theta-invH*(1/m)*(X'*(h-y)+lamda*I*theta);

% end

%

%

% % plot
```

```
% u=linspace(-1,1.5,200);

% v=linspace(-1,1.5,200);

% z=zeros(length(u),length(v));

% for j=1:length(u)

%       for k=1:length(v)

%             z(j,k)=map_feature(u(j),v(k))*theta;

%       end

% end

% hold on

% contour(u,v,z',[0,0],'linewidth',1);




% %% lamda=1

% lamda=1;

% theta=zeros(28,1);

% for j=1:num

%       h=1./(1+exp(-X*theta));

%       H=(1/m)*(X'*(h.*(1-h).*X)+lamda*I);

%       invH=inv(H);

%       theta=theta-invH*(1/m)*(X'*(h-y)+lamda*I*theta);

% end

% % plot
```

```
% u=linspace(-1,1.5,200);

% v=linspace(-1,1.5,200);

% z=zeros(length(u),length(v));

% for j=1:length(u)

%       for k=1:length(v)

%            z(j,k)=map_feature(u(j),v(k))*theta;

%       end

% end

% hold on

% contour(u,v,z',[0,0],'linewidth',2);


 %% lamda=10

lamda=10;

theta=zeros(28,1);

for j=1:num

     h=1./(1+exp(-X*theta));

     H=(1/m)*(X'*(h.*(1-h).*X)+lamda*I);

     invH=inv(H);

     theta=theta-invH*(1/m)*(X'*(h-y)+lamda*I*theta);

end

% plot

u=linspace(-1,1.5,200);
```

```
v=linspace(-1,1.5,200);
z=zeros(length(u),length(v));
for j=1:length(u)
    for k=1:length(v)
    z(j,k)=map_feature(u(j),v(k))*theta;
    end
end
hold on
contour(u,v,z',[0,0],'linewidth',4);
% legend('positive','negative','lamda=0','lamda=1','lamda=10');
```