计算机科学与技术 学院

机器学习 课程实验报告

学号：	姓名：	班级：
实验题目：regularized linear regression and regularized logistic regression.		
实验学时： 2	实验日期： 2018/11/2	

实验目的：

熟悉掌握线性回归和逻辑回归的正则化方法。

硬件环境：

操作系统

Windows 10 家庭中文版 64-bit

CPU

Intel Core i5 7200U @ 2.50GHz 41 ° C

Kaby Lake-U/Y 14nm 工艺

RAM

8.00GB 单个的-通道 未知 (15-15-15-35)

主板

HP 81D1 (U3E1)

图像

Generic PnP Monitor (1920x1080@60Hz)

Intel HD Graphics 620 (HP)

存储器

476GB NVMe THNSN5512GPUK TO (未知)

40GB Microsoft 虚拟磁盘 (File-backed Virtual)

光盘驱动器

没有检测到光纤磁盘驱动

音频

Conexant ISST Audio

软件环境：

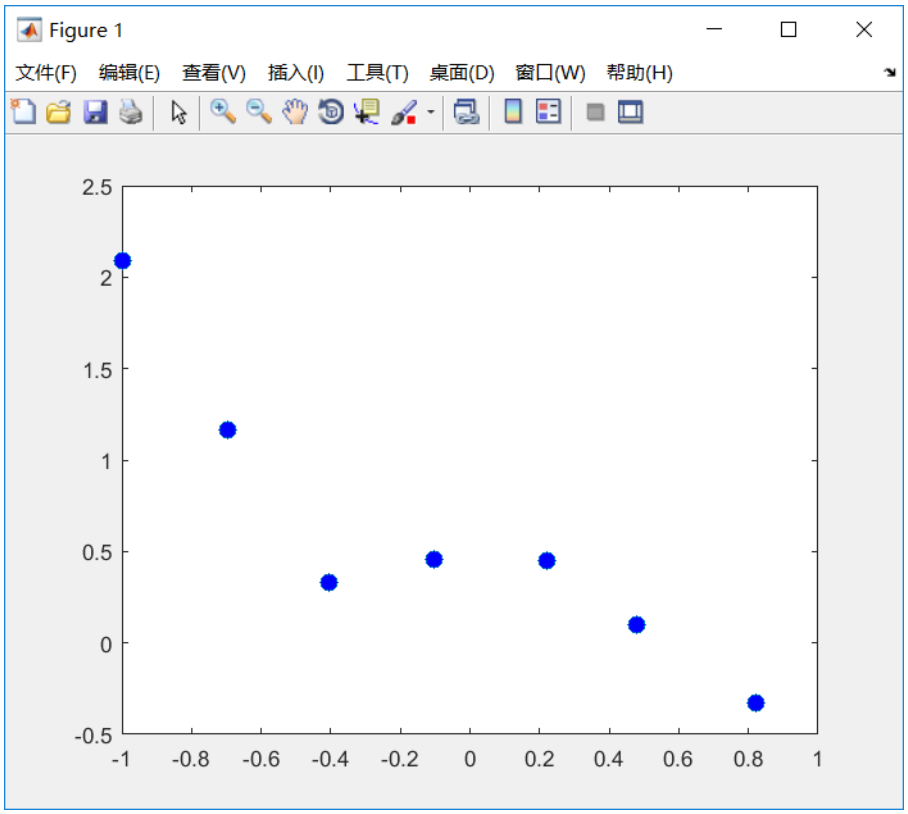
Win10 + matlabR2016a

实验步骤与内容：

一、线性回归正则化

1、首先对数据进行可视化：

如下图所示：



2、构建从 x 的 0 次方到 x 的 5 次方的样本数据矩阵并初始化参数 theta

截距加上 x 的一次项到五次项，将一维的特征拓展为六维的样本数据，如下，同时初始化 6 维的 θ 参数：

```
x = [ones(m, 1), x, x.^2, x.^3, x.^4, x.^5];
theta = zeros(size(x(1,:)))';
```

3、根据如下公式推导得到的参数 theta 的计算公式来定义正则化的超参，并使用 L-2norm 来正则化：

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

```
la = lambda;
L = la.*eye(6);
L(1) = 0;
theta = (x' * x + L)\x' * y
norm_theta = norm(theta)
```

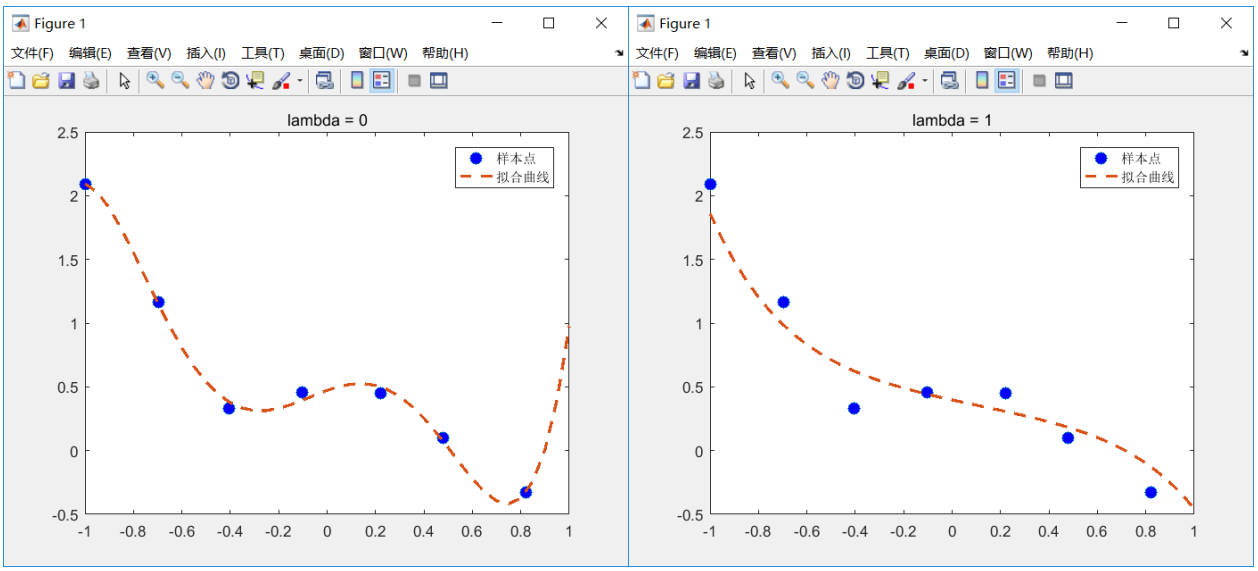
注意正则对角矩阵的第一个对角元素要赋值为 0，并且矩阵为(m+1)*(m+1)。

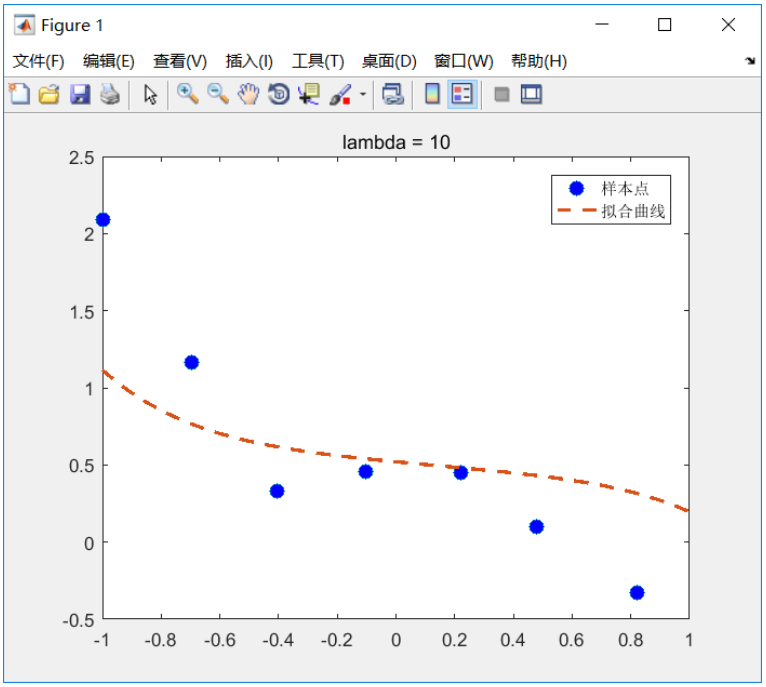
4、最后构建密集的点阵来将拟合的曲线显示出来

更改正则化超参的值得到不同的你和曲线，下面分别是 λ=0（无正则化）、λ=1、和 λ=10 时候的拟合曲线：

```
x_vals = (-1:0.05:1)';
features = [ones(size(x_vals)), x_vals, x_vals.^2, x_vals.^3,...
        x_vals.^4, x_vals.^5];
```

```
    plot(x_vals, features*theta, '--', 'LineWidth', 2)
    title(Title)
    legend('样本点', '拟合曲线')
```



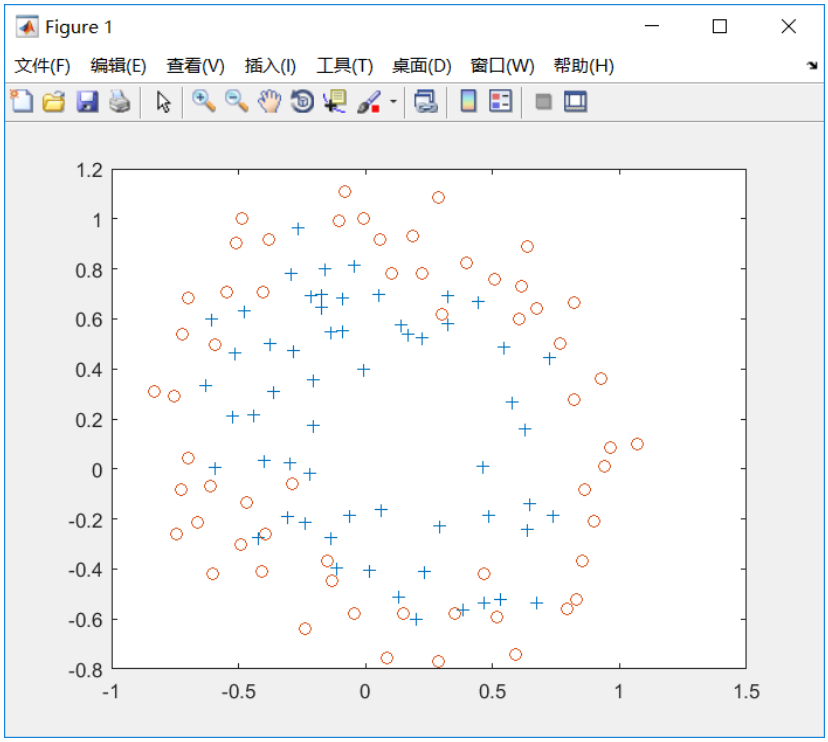


观察后可以发现正则化的超参数值越大，曲线的泛化能力就越强，能够加大对过拟合的抑制程度

一、 逻辑回归的正则化

1、数据集的可视化

对正例数据点和负例分别使用不同的标记显示：



2、调用 map_festure 函数增加 x 的多项式特征，增加维度

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

而后初始化相关变量，迭代次数、匿名的 sigmoid 函数等，利用牛顿法进行参数更新：

```
% 初始化参数 theta
theta = zeros(n, 1);

% 定义匿名 sigmoid 函数
g = @(z)1.0 ./ (1.0 + exp(-z));

% 使用牛顿法先定义最大迭代次数
MAX_ITR = 20;
J = zeros(MAX_ITR, 1);
lam = Lambda;
```

而后进入循环直到达到迭代次数后退出循环：

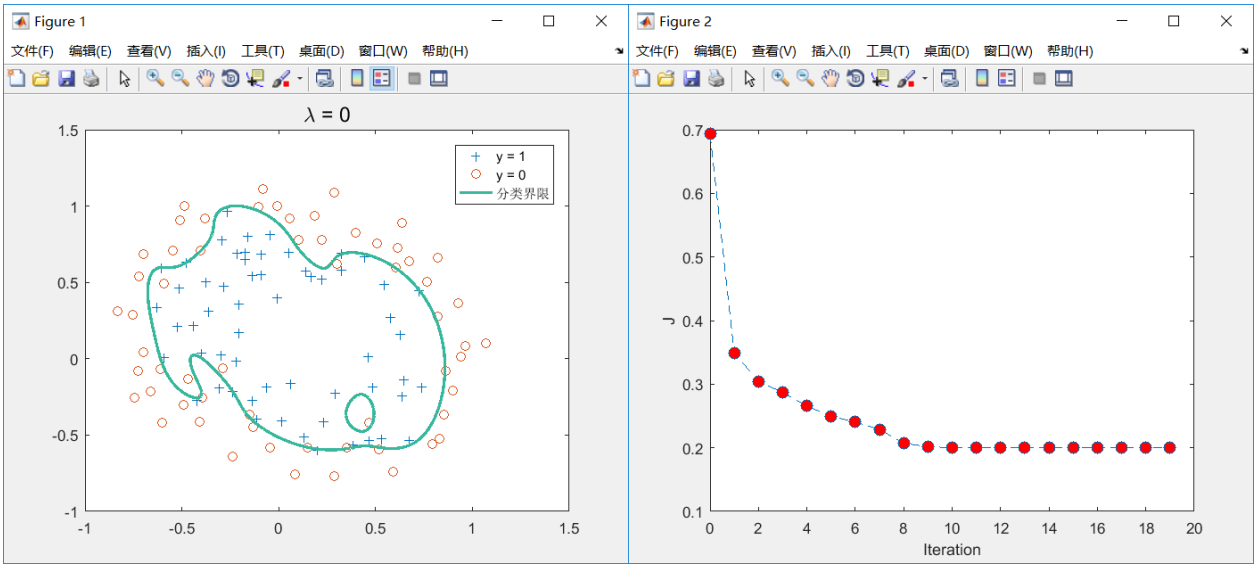
```
  for i = 1:MAX_ITR
      z = x * theta;
      h = g(z);
      % 计算损失函数加上正则项
      J(i) =(1/m)*sum(-y.*log(h) - (1-y).*log(1-h))+ ...
      (lam/(2*m))*norm(theta([2:end]))^2;
      % 计算梯度和海森矩阵
      G = (lam/m).*theta; G(1) = 0; % 额外增加的
      L = (lam/m).*eye(n); L(1) = 0;
      grad = ((1/m).*x' * (h-y)) + G;
      H = ((1/m).*x' * diag(h) * diag(1-h) * x) + L;
```

```
        % 参数的更新
        theta = theta - H\grad;
    end
```

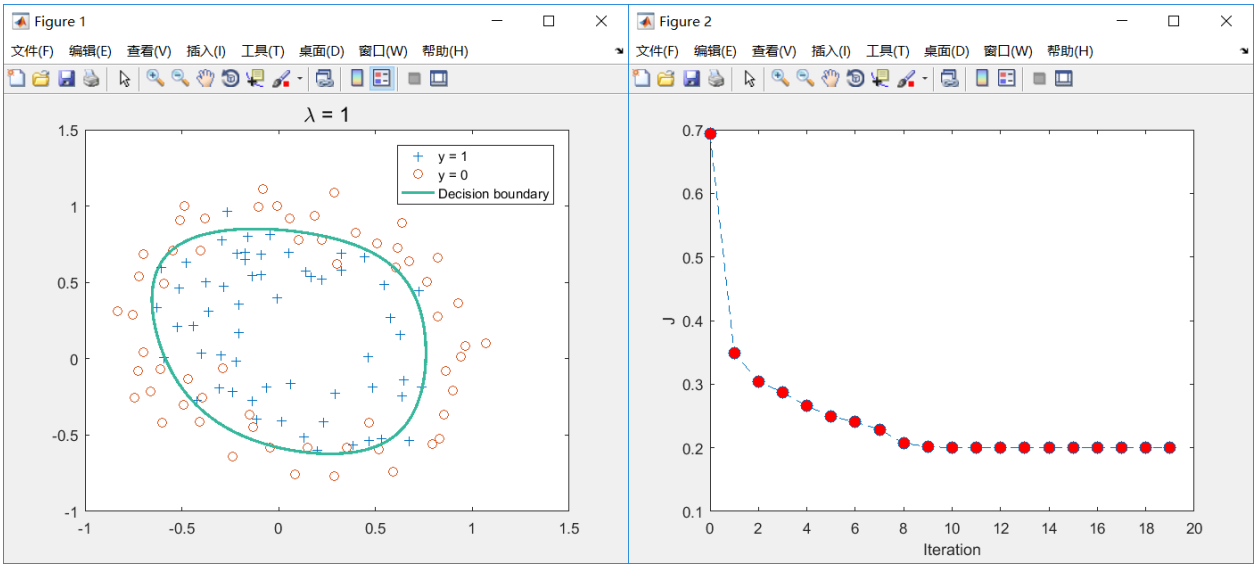
注意类似于线性回归同样要有 `G(1) = 0` 和 `L(1) = 0` 这两项。

3、完成迭代后在数据点集上将分类边界进行可视化，并显示损失函数 J(θ)的下降过程

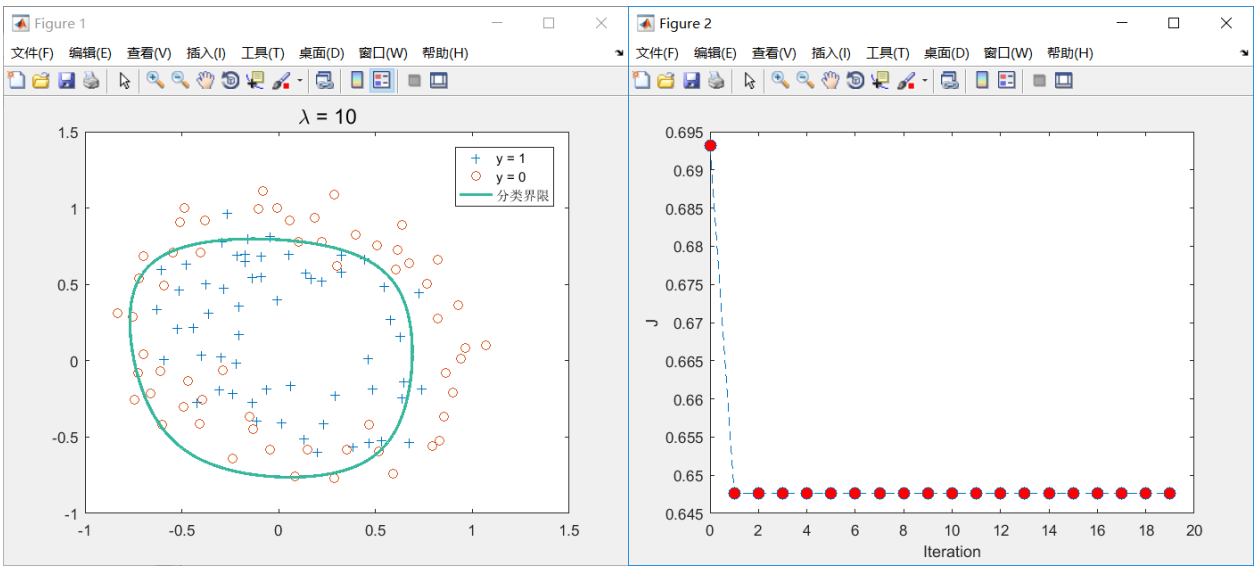
λ = 0：



λ = 1：



λ = 10：



结论分析与体会：

通过实验熟悉掌握线性回归和逻辑回归的正则化方法，对于使用正则化来进行特征选择有了更实际的认识。

附录：程序源代码

（一）正则化线性回归：

```
function ex5Lin(lambda)
% 正则化线性回归
%加载数据
close all; clc
x = load('ex5Linx.dat'); y = load('ex5Liny.dat');
str1 = 'lambda = ';
str2 = num2str(lambda);
Title = [str1,str2];
m = length(y); % 训练样本数量
% 样本展示
figure;
plot(x, y, 'o', 'MarkerFacecolor', 'b', 'MarkerSize', 8);
% 构建从 x 的 0 次方到 x 的 5 次方的样本数据矩阵并初始化参数 theta
x = [ones(m, 1), x, x.^2, x.^3, x.^4, x.^5];
theta = zeros(size(x(1,:)))';
% 定义正则化超参
la = lambda;
L = la.*eye(6);
L(1) = 0;
theta = (x' * x + L)\x' * y
norm_theta = norm(theta)
hold on;
```

```
% 构建密集的矩阵来将拟合的曲线显示出来
x_vals = (-1:0.05:1)';
features = [ones(size(x_vals)), x_vals, x_vals.^2, x_vals.^3,...
        x_vals.^4, x_vals.^5];
plot(x_vals, features*theta, '--', 'LineWidth', 2)
title(Title)
legend('样本点', '拟合曲线')
hold off
```

（二）正则化逻辑回归

```
function ex5Log(Lambda)
% 正则化逻辑回归
close all; clc
x = load('ex5Logx.dat');
y = load('ex5Logy.dat');

% 可视化样本数据，使用不同标记正例和反例
figure % Find theindices for the2 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 ' )

% 使用 map_festure 函数增加 x 的多项式特征
x = map_feature(x(:,1), x(:,2));
[m, n] = size(x);
% 初始化参数 theta
theta = zeros(n, 1);
% 定义匿名 sigmoid 函数
g = @(z)1.0 ./ (1.0 + exp(-z));
% 使用牛顿法先定义最大迭代次数
MAX_ITR = 20;
J = zeros(MAX_ITR, 1);
lam = Lambda;

for i = 1:MAX_ITR
    z = x * theta;
    h = g(z);
    % 计算损失函数加上正则项
    J(i) =(1/m)*sum(-y.*log(h) - (1-y).*log(1-h))+ ...
    (lam/(2*m))*norm(theta([2:end]))^2;
    % 计算梯度和海森矩阵
    G = (lam/m).*theta; G(1) = 0; % 额外增加的
    L = (lam/m).*eye(n); L(1) = 0;
    grad = ((1/m).*x' * (h-y)) + G;
```

```
    H = ((1/m).*x' * diag(h) * diag(1-h) * x) + L;
    % 参数的更新
    theta = theta - H\grad;

end
norm_theta = norm(theta)

u = linspace(-1, 1.5, 200);
v = linspace(-1, 1.5, 200);
z = zeros(length(u), length(v));
for i = 1:length(u)
    for j = 1:length(v)
        z(i,j) = map_feature(u(i), v(j))*theta;
    end
end
z = z'; % 进行转置

contour(u, v, z, [0, 0], 'LineWidth', 2)
legend('y = 1', 'y = 0', '分类界限')
title(sprintf('\\lambda = %g', lam), 'FontSize', 14)
hold off

% 打印 J 并进行可视化
J
figure
plot(0:MAX_ITR-1, J, 'o--', 'MarkerFaceColor', 'r', 'MarkerSize', 8)
xlabel('Iteration'); ylabel('J')
```