山东大学计算机科学与技术学院

机器学习与模式识别课程实验报告

学号：201600181073	姓名：　唐超	班级：　智能 16
实验题目：Regularization		
实验学时：2	实验日期：　　2018.11.9	

实验目的：

实现加入正则项的线性回归和逻辑回归。

硬件环境：

DELL 台式机

软件环境：

MATLAB R2018b

实验步骤与内容：

1. Regularized Linear Regression

用 5 阶多项式进行拟合

$$h_\theta(x)=\theta_0+\theta_1x+\theta_2x^2+\theta_3x^3+\theta_4x^4+\theta_5x^5$$

与非正则化线性回归相比，$J(\theta)$ 多了一个“惩罚项”：

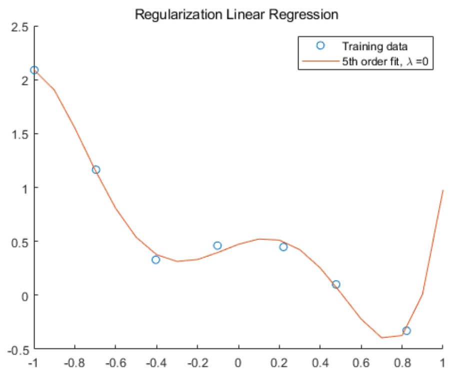
$$J(\theta)=\frac{1}{2m}[\sum_{i=1}^{m}(h_\theta(x^{(i)}-y^{(i)})^2+\lambda\sum_{j=1}^{n}\theta_j^2]$$

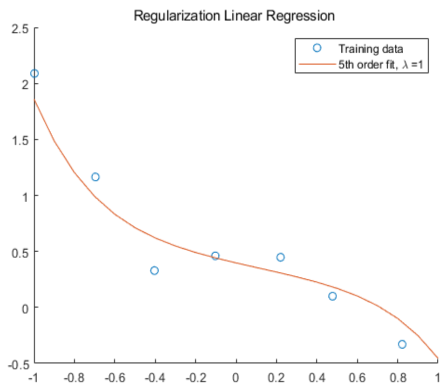
此时，

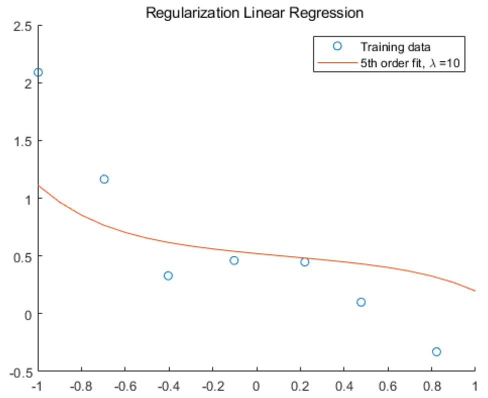
$$\theta=(x^Tx+\lambda A)^{-1}x^Ty$$

其中 A 为（n+1）阶对角方阵，左上角为 0，其他对角元素为 1.

分别取 $\lambda=0,1,10$ ，得到拟合结果如下：



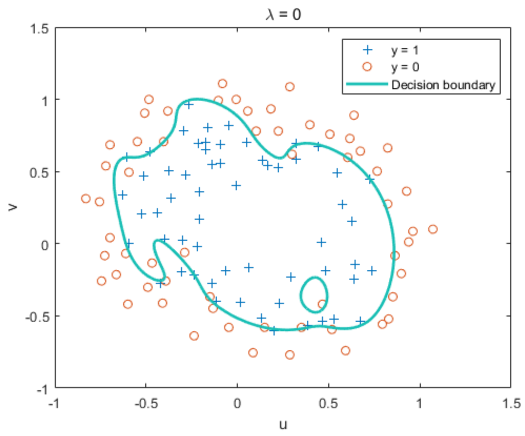


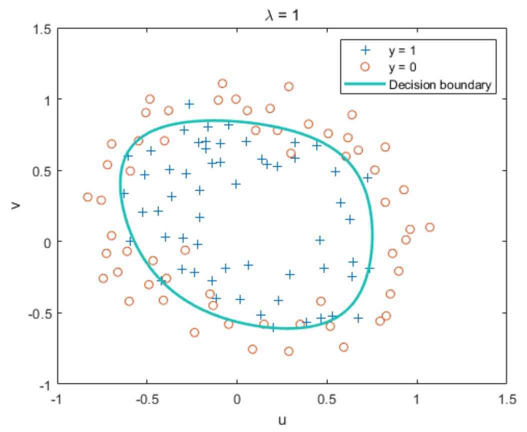


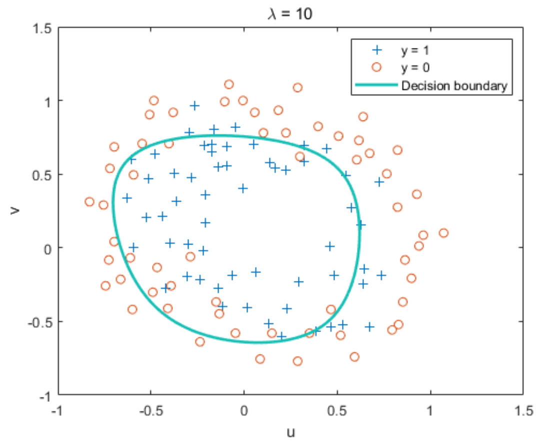
三次 theta 的 L2-norm 为 $norm\_theta=(8.1687,0.8098,0.5931)$。

2. Regularized Logistic Regression

与正则化在线性回归中的应用步骤类似，在逻辑回归的基础上加入正则项迭代求θ，分别取$\lambda=0,1,10$得到分类结果如下：







theta 的 L2-norm 为 norm_theta=(7172.6946, 4.0346, 0.8402).

结论分析与体会：

可见，当我们增加参数来使 $J(\theta)$ 第一项的值减小时，正则化项会逐渐增大，从而平衡模型复杂度和拟合程度来避免过拟合问题。正则化项系数 λ 越大，最小化 $J(\theta)$ 的结果更倾向于满足于正则项最小，从而 theta 的 L2-norm 也越小，并且拟合曲线对训练数据拟合得越不精确，如果 λ 过大，也可能导致欠拟合。

附录：程序源代码

```
%Experiment 5: Regularization

%Regularized Linear Regression
x = load('ex5Linx.dat');
y = load('ex5Liny.dat');
m = length(x);
n = 5; %拟合的多项式阶数

x1 = [ones(m,1), x, x.^2, x.^3, x.^4, x.^5];

d = ones(n+1,1);
D = diag(d);
D(1,1) = 0;
lamda = [0,1,10];
t = (-1:0.1:1)';
T = [ones(length(t),1), t, t.^2, t.^3, t.^4, t.^5];
norm_theta = zeros(length(lamda),1);

for i = 1:length(lamda)
    theta = (x1'*x1 + lamda(i)*D)\x1'*y;
    y1 = T*theta;
    norm_theta(i) = norm(theta);
    figure;
    scatter(x,y);
    hold on
    plot(t, y1)
    legend('Training data', ['5th order fit, \lambda =' num2str(lamda(i))]);
    title('Regularization Linear Regression ');
end


%Regularized Logistic Regression
```

```
x = load('ex5Logx.dat');
y = load('ex5Logy.dat');
pos = find(y==1);
neg = find(y==0);
plot(x(pos,1),x(pos,2),'+')
hold on
plot(x(neg,1),x(neg,2),'o')
xlabel('u');
ylabel('v');



u = x(:,1);
v = x(:,2);
x = map_feature(u,v);
m = 117;
n = 28;

theta = zeros(28,1);
g = inline('1.0./(1.0+exp(-z))');
lambda = 10;
d = ones(n,1);
D = diag(d);
D(1,1) = 0;

for i = 1:10
    grad = 1/m*((g(theta'*x')- y')*x)'+ lambda/m*theta;
    H = 1/m*x'*diag(g(theta'*x').*g(-theta'*x'))*x + lambda/m*D;
    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(j,i) = map_feature(u(i),v(j))*theta;
    end
end
contour(u,v,z, [0,0], 'LineWidth', 2)
legend('y = 1','y = 0','Decision boundary');
title(['\lambda = ' num2str(lambda)]);
```

Published with MATLAB® R2018b