

计算机科学与技术 学院

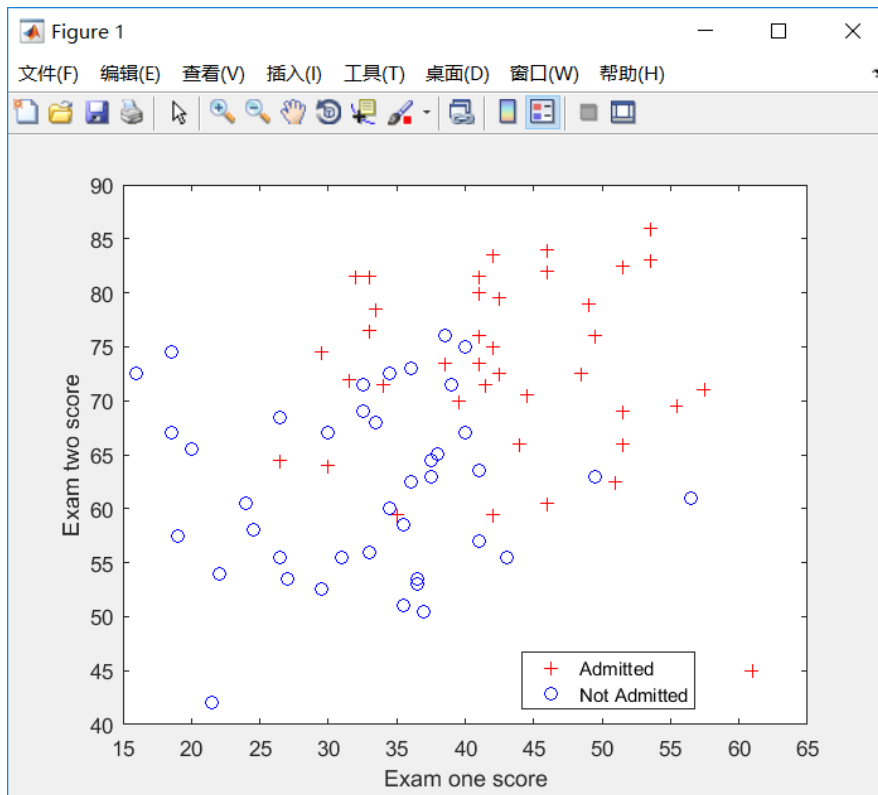
机器学习 课程实验报告

学号：	姓名：	班级：
实验题目：Logistic Regression and Newton 's Method		
实验学时：2.0	实验日期：2018/10/22	
实验目的： 学习使用逻辑回归和牛顿法的梯度下降		
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 T0 (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： 1.1 可视化数据集： <pre>x = load('ex4x.dat'); y = load('ex4y.dat'); [m, n] = size(x); % 给 x 加一列作为 bias x = [ones(m, 1), x]; % 可视化训练数据 figure pos = find(y); neg = find(y == 0); plot(x(pos, 2), x(pos, 3), '+')</pre>		

```

hold on
plot(x(neg, 2), x(neg, 3), 'o')
hold on
xlabel('Exam 1 score')
ylabel('Exam 2 score')

```



1.2 梯度下降:

现计算出第一和第二次的 $J(1)$ 和 $J(2)$ ，而后进入循环迭代，知道迭代次数达到最大次数或收敛，而后打印出参数 θ 如下：其中用 $\theta(t+1) = \theta(t) - H^{-1} \nabla_{\theta} J$ 来近似代替更新参数

```

theta =
-16.3787
  0.1483
  0.1589

```

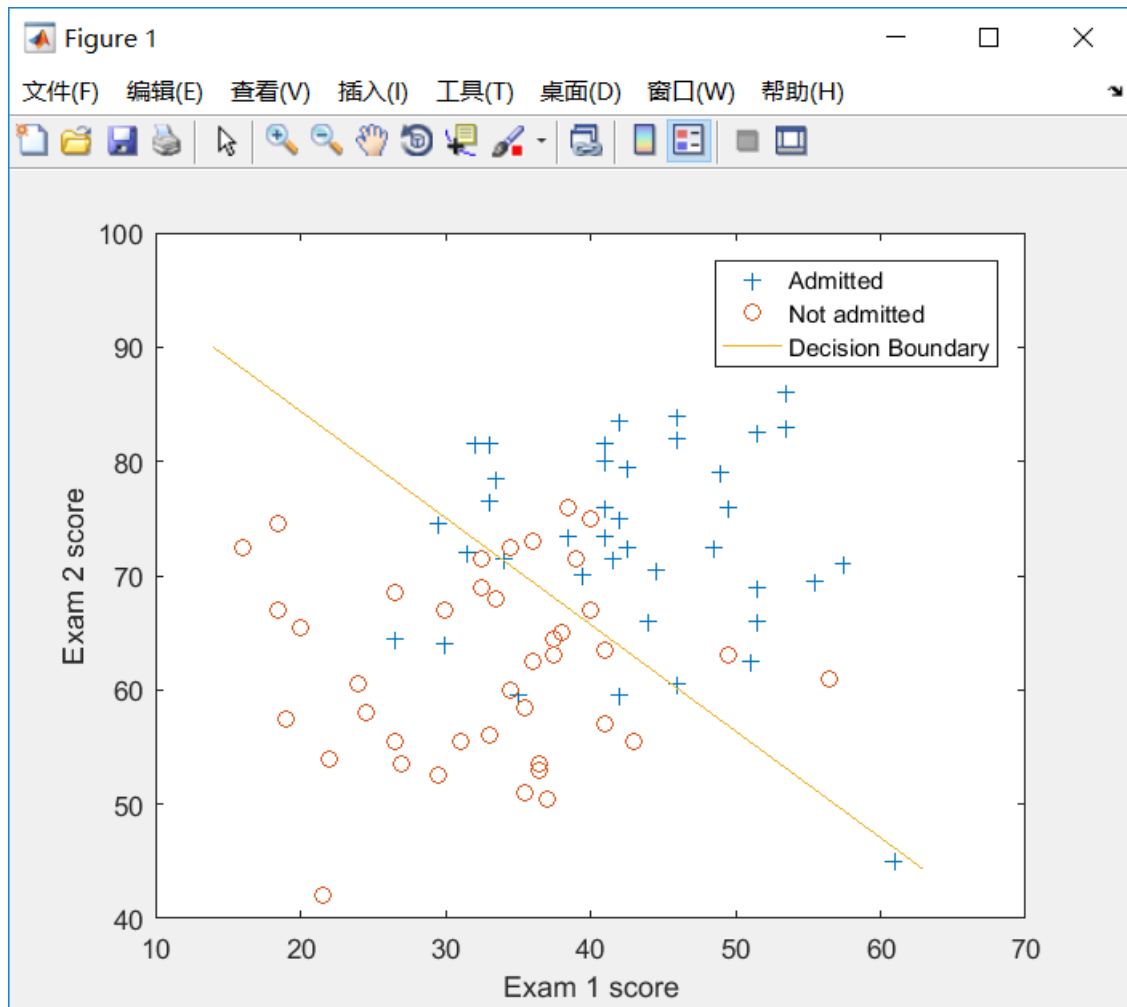
1.3 显示结果和 $J(i)$ 下降过程，并对 [20, 80] 的人进行预估:

```

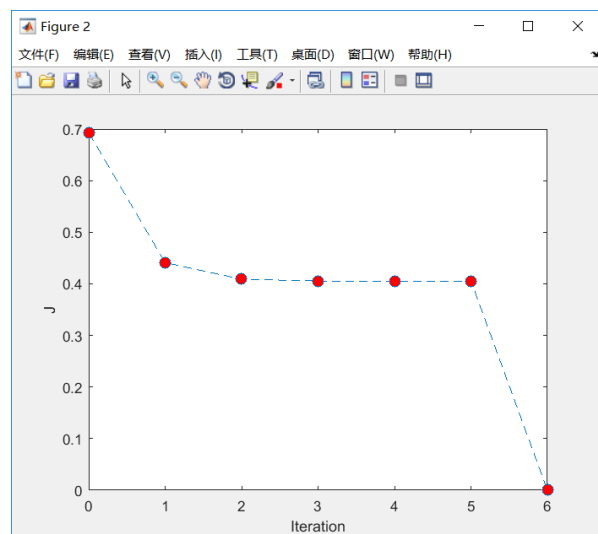
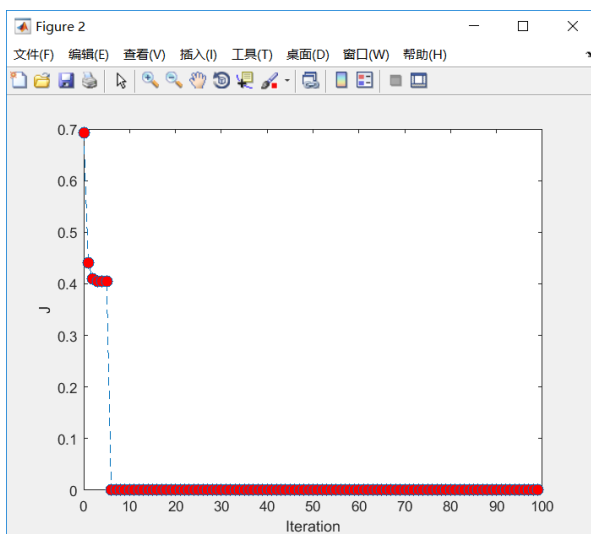
prob =
  0.6680
J =
  0.6931
  0.4409
  0.4089
  0.4055
  0.4054
  0.4054

```

0
0
0
...



1.4 可视化 $J(\theta)$:



结论分析与体会：

比较熟悉掌握了逻辑回归和牛顿法，有所收获。

附录：程序源代码

主要算法函数：ex4.m

```
clear all; close all; clc
x = load('ex4x.dat');
y = load('ex4y.dat');
[m, n] = size(x);
% 给 x 加一列作为 bias
x = [ones(m, 1), x];
% 可视化训练数据
figure
pos = find(y); neg = find(y == 0);
plot(x(pos, 2), x(pos, 3), '+')
hold on
plot(x(neg, 2), x(neg, 3), 'o')
hold on
xlabel('Exam 1 score')
ylabel('Exam 2 score')

% 初始化参数
theta = zeros(n+1, 1);
% 定义匿名 sigmoid 函数
g = @(z) 1.0 ./ (1.0 + exp(-z));
% 定义最大迭代次数和阈值
MAX_ITR = 100;
J = zeros(MAX_ITR, 1);
thred = 0.00001;

% 第一次迭代计算
i = 1;
z = x * theta;
h = g(z);
% 计算梯度和海森矩阵
grad = (1/m) .* x' * (h-y);
H = (1/m) .* x' * diag(h) * diag(1-h) * x;
% 计算最小化的损失函数
J(i) = (1/m) * sum(-y.*log(h) - (1-y).*log(1-h));
theta = theta - H\grad;
% 第二次迭代计算
i = 2;
z = x * theta;
h = g(z);
grad = (1/m) .* x' * (h-y);
H = (1/m) .* x' * diag(h) * diag(1-h) * x;
J(i) = (1/m) * sum(-y.*log(h) - (1-y).*log(1-h));
```

```

theta = theta - H\grad;

for i = 3:MAX_ITR
    z = x * theta;
    h = g(z);

    grad = (1/m).*x' * (h-y);
    H = (1/m).*x' * diag(h) * diag(1-h) * x;

    J(i) = (1/m)*sum(-y.*log(h) - (1-y).*log(1-h));
    %收敛时退出迭代
    if abs(J(i)-J(i-1))<=thred
        break;
    end
    theta = theta - H\grad;
end
% 打印参数
theta
% 计算【20, 80】的同学是否通过
prob = 1 - g([1, 20, 80]*theta)
% 展示分类预测结果
plot_x = [min(x(:,2))-2, max(x(:,2))+2];
plot_y = (-1./theta(3)).*(theta(2).*plot_x +theta(1));
plot(plot_x, plot_y)
legend('Admitted', 'Not admitted', 'Decision Boundary')
hold off
%展示损失函数下降过程
figure
plot(0:MAX_ITR-1, J, 'o--', 'MarkerFaceColor', 'r', 'MarkerSize', 8)
xlabel('Iteration'); ylabel('J')
J

```