

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： 201700301042	姓名： 陈佳睿	班级： 17 人工智能
实验题目： Linear Regression		
实验目的： 通过 matlab/octave/python 进行编码实现一元和二元线性回归， 并进而对于机器学习理论有比较深刻的见解		
硬件环境： Macbook pro		
软件环境： Macos majove		
<p>实验步骤与内容：</p> <p>首先， 因为我使用 octave 来进行实验， 但是对于 ovtave（matlab）的语法还不是很熟悉 所以做实验遇到了比较多的阻碍</p> <p>实验指导书引导了我们去加载所给的关于年龄和身高的两个.dat 文件数据</p> <p>利用数据画出了散点图并使用 ylabel 和 xlabel 函数标注了各自的坐标轴的含义</p> <div data-bbox="236 1207 754 1632"></div>		
<p>上图是加载数据之后所得的结果图</p> <p>接下来， 使用 length 方法将样本的样例个数值赋给变量 m， 创建一个 m 行 2 列的向量 x</p> <p>因为实验用例的特殊性 只有一个解释变量和一个被解释变量 theta0 为常数项的加权</p> <p>Theta1 是唯一解释变量的加权</p> <p>所以 我初始化 theta 为一个两行一列的 0 向量</p>		

和实验指导书中写的一样，将学习率 α 设置为 0.07

delta 矩阵用来记录残差值

我们使用 while 循环结构来进行梯度下降的逼近

将精度设置为 0.00001 以下

循环退出条件设置为 $\text{abs}(\max(\text{delta}(:))) > 0.0001$

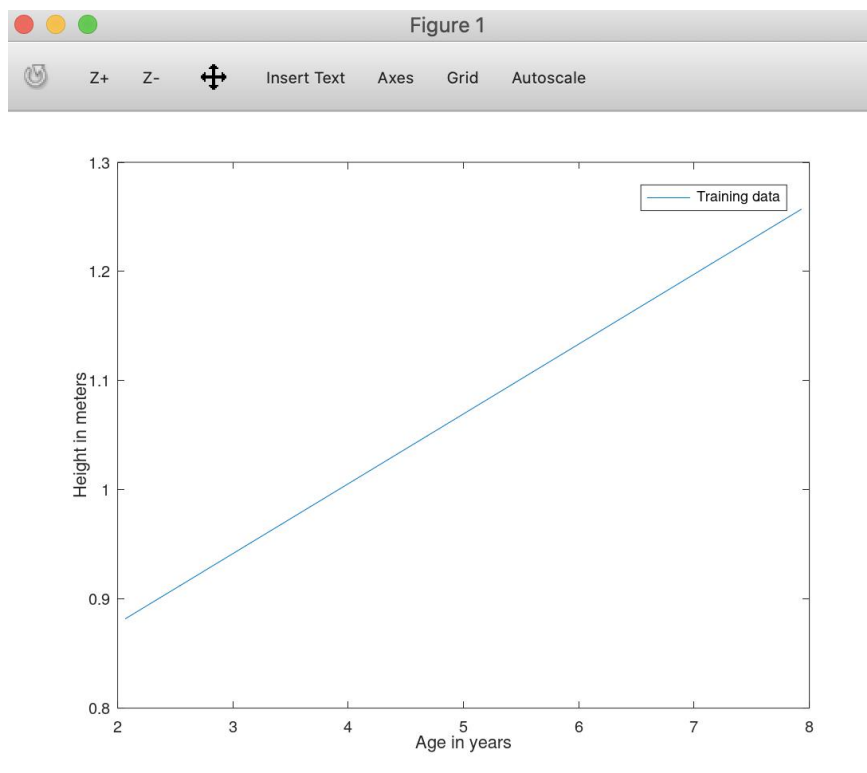
```
>> theta
theta =

    0.749639
    0.063976
```

最终得到的收敛结果如上图

代码:

```
>> while abs(max(delta(:))) > 0.00001
    h = sum(x * theta,2);
    err = h - y;
    delta = x' * err / m;
    theta = theta - alpha * delta;
end
```

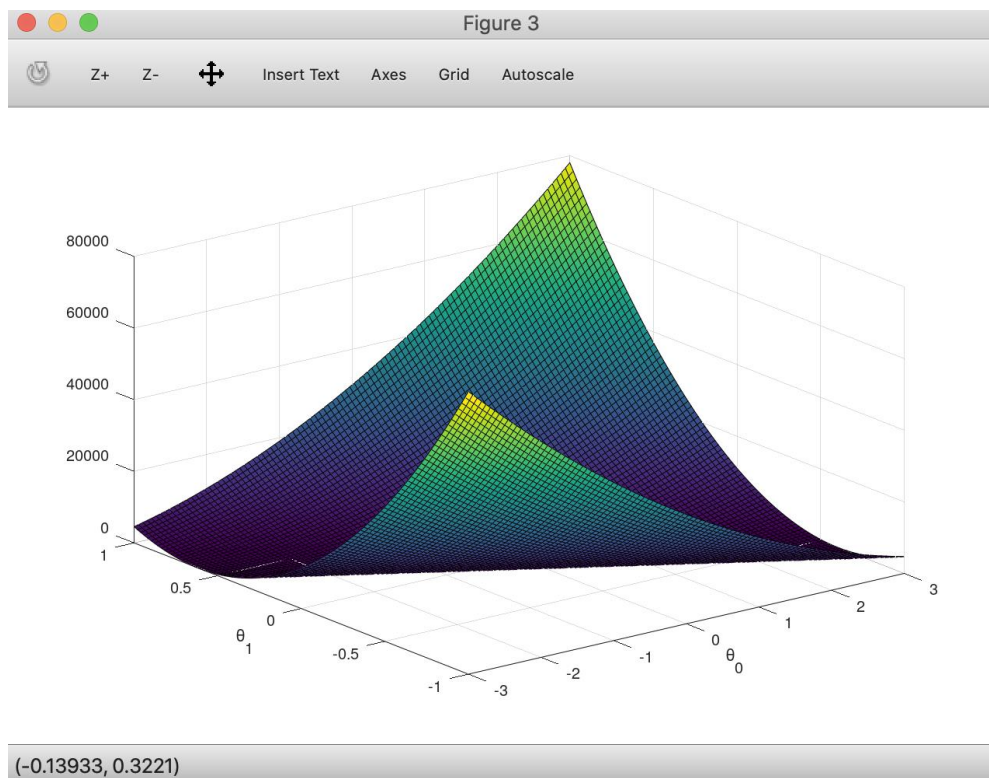


拟合曲线如上图所示

理解残差 $J(\theta)$

使用 surf 函数绘制出了二维平面

```
>> J_vals = zeros(100, 100);
>> theta0_vals = linspace(-3, 3, 100);
>> theta1_vals = linspace(-1, 1, 100);
>> for i=1:length(theta0_vals)
    for j=1:length(theta1_vals)
        t = [theta0_vals(i);theta1_vals(j)];
        h = sum(x*t);
        J_vals(i,j) = sum((h-y).^2)/(2*m);
    end
end
>> J_vals = J_vals';
>> figure;
>> surf(theta0_vals,theta1_vals,J_vals)
>> xlabel('\theta_0');ylabel('\theta_1')
```



结论分析与体会：这次的实验使用 octave (matlab) 进行实现，初步了解了如何通过给定的数据集来绘制点图，并使用 matlab 内置函数进行二维线性回归，通过最小化残差函数来使用线性函数拟合点分布，学会使用 surf 函数来绘制描述 $J(\theta)$ 平面。

代码：

x =

```
load('ex2x.dat');
```

```
y = load('ex2y.dat');
```

```
figure % open a new figure window
```

```
plot(x, y, 'o');
```

```
ylabel('Height in meters')
```

```
xlabel('Age in years')
```

```
m = length(y);
```

```
x = [ones(m, 1), x]; %add a column of ones to x
```

```
theta = zeros(size(x,2),1);
```

```
alpha = 0.07;
```

```
delta = ones(size(theta));
```

```
while abs(max(delta(:))) > 0.00001
```

```
h = sum(x * theta,2);
```

```
err = h - y;
```

```
delta = x' * err / m;
```

```
theta = theta - alpha * delta;
```

end

hold on % Plot new data without clearing old plot

plot(x(:,2), x*theta, '-') % remember that x is now a

matrix with 2

% columns

% and the second column contains the

% time info

legend('Training data', 'Linear regression')

J_vals = zeros(100, 100); % initialize Jvals to 100x100

matrix of

% 0's

theta0_vals = linspace(-3, 3, 100);

theta1_vals = linspace(-1, 1, 100);

for i = 1:length(theta0_vals)

for j = 1:length(theta1_vals)

t = [theta0_vals(i); theta1_vals(j)];

h = sum(x * t);

J_vals(i,j) = sum((h - y).^2) / (2*m);

```
end
```

```
end
```

```
% Plot the surface plot
```

```
% Because of the way meshgrids work in the surf
```

```
command, we need to
```

```
% transpose J_vals before calling surf, or else the axes
```

```
will be
```

```
% flipped
```

```
J_vals = J_vals';
```

```
figure;
```

```
surf(theta0_vals, theta1_vals, J_vals)
```

```
xlabel('\theta_0'); ylabel('\theta_1')
```