

机器学习与模式识别 课程实验报告

学号：201600301304	姓名：贾乘兴	班级：人工智能 16
实验题目：线性回归		
实验学时：2 小时	实验日期：2018/9/28	
实验目的：通过梯度下降的方法不断训练线性回归中的参数，并绘制误差的分布来观察参数下降的方向，记录梯度下降时参数变换，最后利用训练参数对 3.5 与 7 岁身高进行预测		
硬件环境：mac os		
软件环境：matlab 2017b		
<p>实验步骤与内容：</p> <p>一. 导入数据并显示</p> <p>1. 所给数据中，ex1x.dat 为 x 数据，ex1y.dat 为 y 标签，导入并绘制点的分布，代码如下：</p> <pre>%% show the data x=load('ex1x.dat'); y=load('ex1y.dat'); figure plot(x,y,'o'); xlabel('age in years'); ylabel('height in meters');</pre> <p>2. 得到数据的规模，并进行处理，代码如下：</p> <pre>m=length(y); X=[ones(m,1),x];</pre> <p>二. 训练定义</p> <p>1. 目标函数定义：本实验目的为线性回归，故目标函数为$y=wx+b$，对函数进行简化，可表示为$y=X*\theta$，X为$m*2$维矩阵，θ为$2*1$列向量</p> <p>2. 损失函数定义：采用最小二乘的定义，损失函数定义如下</p> $J(\theta) = \frac{1}{2m} \sum_{i=1}^m [h_i(\theta) - y_i]^2$ <p>其中$h(\theta)$为假设，本实验为线性函数</p> $h_i(\theta) = X_i \cdot \theta$ <p>3. 训练目标为最小化损失函数，通过得到最优的θ从而得到最终的目标函数，本实验采用的是梯度下降的方法，$J(\theta)$对θ求导可得</p> $J'(\theta) = \frac{1}{m} \sum_{i=1}^m X_i^T (X_i \theta - y_i)$ <p>θ的更新公式为：</p>		

$$\theta := \theta - \alpha J'(\theta)$$

其中alpha为学习率，本实验采用的学习率为0.07，最终得到theta的更新为

$$\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^m X_i^T (X_i \theta - y_i)$$

本实验设置训练迭代次数为1500，(0, 1)内设置随机的初始值

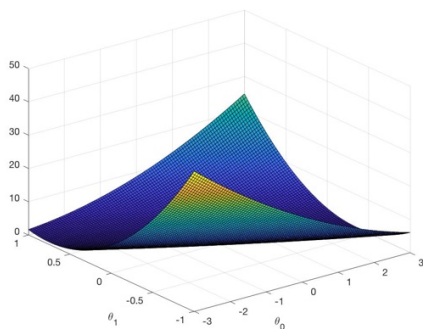
三. 实验过程

1. 绘制损失函数的图像，代码如下：

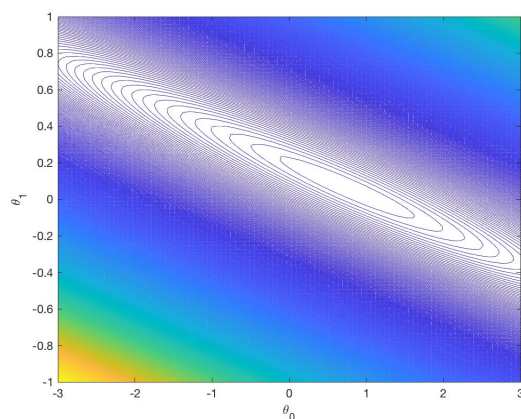
```
%% loss=1/2*sum(x*theta-y)^2
Jtheta=zeros(100,100);
theta0=linspace(-3,3,100);
theta1=linspace(-1,1,100);

for p=1:length(theta0)
    for q=1:length(theta1)
        t=[theta0(p);theta1(q)];
        Jtheta(p,q)=(X*t-y)'*(X*t-y)/(2*m);
    end
end
figure
surf(theta0,theta1,Jtheta')
xlabel('\theta_0');
ylabel('\theta_1');
figure
contour(theta0,theta1,Jtheta',1000)
xlabel('\theta_0');
ylabel('\theta_1');
```

绘制图像结果如下，可大致观察到最低点的大致位置，其中
theta = [theta0;theta1]



绘制等高线图如下：



2. 训练过程更新theta代码如下:

```
%% gradient descent
theta=zeros(2,1);
num=1500;
alpha=0.07;
for k=1:num
    theta=theta-alpha*sum((X*theta-y).*X)'/m);
end
hold on;
plot(X(:,2),X*theta,'-o');
```

3. 本实验为线性回归, 由损失函数定义与目标函数可知, 我们的目的为找到最终的theta, 使得

$$\theta = \arg \min (X \cdot \theta - y)^2$$

故由最小二乘可知, 我们可以直接得到最优的theta, 即:

$$\theta = X^+ y$$

其中 X^+ 为 X 的伪逆

$$X^+ = (X^T X)^{-1} X^T$$

所得到theta为最优的

将实验原始数据, 梯度下降训练结果, 最小二乘的结果绘制成图像进行对比, 代码入下:

```
%% show the data
.....(代码如上)

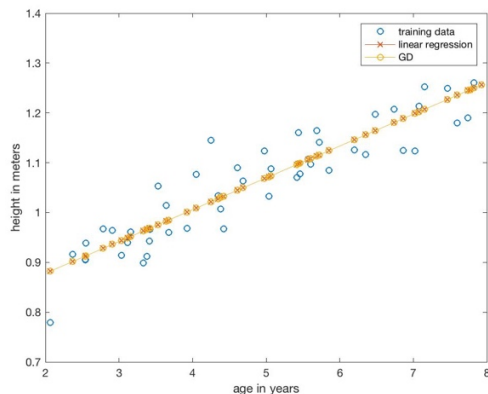
%% show the train(y=Xw,w=inv(X'X)*X')
theta=pinv(X)*y;
hold on;
plot(X(:,2),X*theta,'-x');

%% gradient descent
.....(代码如上)

%% plot legend
```

```
legend('training data','linear regression','GD')
```

得到的图像为



通过对比可知梯度下降的收敛效果很好，与精确的最优解基本一致

四. 实验结果

1. 实验结果保留四位小数，所得梯度下降和线性回归结果相等， $\theta_0=0.7502$
 $\theta_1=0.0639$ ，最小二乘误差为 $9.8707e-04$
2. 实验结果保留四位小数在初始值为 $[0, 0]$ ，学习率 0.07 下， θ 第一次迭代的数值为 $[0.0745; 0.3800]$
3. 实验结果保留四位小数，对年龄为 3.5 与 7 岁进行预测，得到身高分别为 $0.9737m$ 与 $1.1973m$

结论分析与体会：通过本次实验，对梯度下降方法有了初步的了解，该方法在函数不复杂的情况下可以较为有效的收敛于全局最优，但当函数复杂的情况下，结果对参数依赖较大，同时我们可以使用一些数值优化方法对梯度下降进行调整，比如加入物理中动量的模型

附录：程序源代码

ex1.m

```
clear,clc;
%% show the data
x=load('ex1x.dat');
y=load('ex1y.dat');
figure
plot(x,y,'o');
xlabel('age in years');
ylabel('height in meters');

m=length(y);
```

```

X=[ones(m,1),x];

%% show the train(y=Xw,w=inv(X'X)*X')
theta=pinv(X)*y
(X*theta-y)'*(X*theta-y)/(2*m)
hold on;
plot(X(:,2),X*theta,'-x');

%% gradient descent
theta=zeros(2,1);
alpha=0.07;
for k=1:1500
    theta=theta-alpha*sum((X*theta-y).*X)'/(m);
    if k==1
        theta
    end
end
hold on;
plot(X(:,2),X*theta,'-o');
legend('training data','linear regression','GD')

%% loss=1/2*sum(x*theta-y)^2
Jtheta=zeros(100,100);
theta0=linspace(-3,3,100);
theta1=linspace(-1,1,100);

for p=1:length(theta0)
    for q=1:length(theta1)
        t=[theta0(p);theta1(q)];
        Jtheta(p,q)=(X*t-y)'*(X*t-y)/(2*m);
    end
end
figure
surf(theta0,theta1,Jtheta')
xlabel('\theta_0');
ylabel('\theta_1');

figure
contour(theta0,theta1,Jtheta',1000)
xlabel('\theta_0');
ylabel('\theta_1');

%% pre
y1=[1,3.5]*theta
y2=[1,7.0]*theta

```