

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：多元线性回归		
实验学时：2 小时	实验日期：2018/9/30	
实验目的：通过在多元线性回归中应用梯度下降的方法，对不同学习率的收敛速度进行判断，找到收敛效果较好的学习率，同时使用伪逆较快的计算得到结果，与梯度下降的结果进行对比分析，最后，对给定数据进行预测，用梯度下降与伪逆计算得到结果对比		
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</p> <p>1. 所给数据中，ex2x.dat 为 x 数据，ex2y.dat 为 y 标签，导入数据代码如下：</p> <pre>%% load data x=load('ex2x.dat'); y=load('ex2y.dat');</pre> <p>2. 得到数据的规模，并进行数据预处理，将数据标准化，标准化为：</p> $x' = \frac{x - E(x)}{\sqrt{\text{var}(x)}}$ <p>代码如下：</p> <pre>m=length(y); X=[ones(m,1),x]; %% process sig=std(X); mu=mean(X); X(:,2)=(X(:,2)-mu(2))/sig(2); X(:,3)=(X(:,3)-mu(3))/sig(3);</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$m*3$维矩阵，θ为$3*1$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>		

$$h_i(\theta) = X_i \cdot \theta$$

3. 训练目标为最小化损失函数，通过得到最优的theta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theta求导可得

$$J'(\theta) = \frac{1}{m} \sum_{i=1}^m X_i^T (X_i \theta - y_i)$$

theta的更新公式为：

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

最终得到theta的更新为

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

本次实验目的为通过绘制不同学习率下多次迭代的误差变化的曲线，从而选择得到较好的学习率，学习率选择范围为（0.001，10）

三. 实验过程

1. 绘制不同学习率下损失函数的图像，选取学习率为1.0, 0.5, 0.2, 0.1，代码如下：

```
%% learning rate
```

```
theta0=rands(3,1);
```

```
num=50;
```

设置第一组学习率为1.0

```
% group 1
```

```
alpha=1.0;
```

```
Jtheta=zeros(num,1);
```

```
theta=theta0;
```

```
for k=1:num
```

```
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
```

```
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
```

```
end
```

```
figure
```

```
plot(0:num-1,Jtheta,'b-','LineWidth',2);
```

```
y_pre(1)=x_pre*theta;
```

```
Jtheta(num)
```

设置第二组学习率为0.5

```
% group 2
```

```
alpha=0.5;
```

```
Jtheta=zeros(num,1);
```

```
theta=theta0;
```

```
for k=1:num
```

```
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
```

```
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
```

```
end
```

```
hold on
```

```
plot(0:num-1,Jtheta,'r-','LineWidth',2);
y_pre(2)=x_pre*theta;
Jtheta(num)
```

设置第三组学习率为0.2

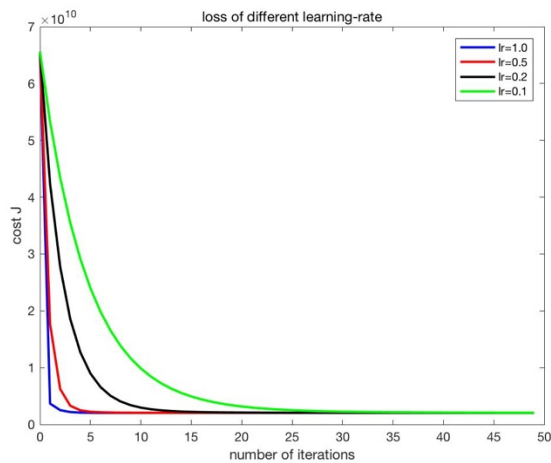
```
% group 3
alpha=0.2;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
end
hold on
plot(0:num-1,Jtheta,'k-','LineWidth',2);
y_pre(3)=x_pre*theta;
Jtheta(num)
```

设置第四组学习率为0.1

```
% group 4
alpha=0.1;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
end
hold on
plot(0:num-1,Jtheta,'g-','LineWidth',2);
y_pre(4)=x_pre*theta;
Jtheta(num)
```

```
% plot
xlabel('number of iterations');
ylabel('cost J');
legend('lr=1.0','lr=0.5','lr=0.2','lr=0.1')
title('loss of different learning-rate')
```

绘制多组学习率下的图像结果如下,



通过观察可知，学习率在1.0附近时收敛速度最快

2. 获得theta最终训练结果，并预测（1，1650，3）时y的数据

采取学习率为1.0时50次迭代便收敛了，结果为

theta = 1.0e+05 * [3.4041; 1.1063; -0.0665]

预测数据代码如下：

```
%% pre-data
x_pre=[1 1650 3];
x_pre(2)=x_pre(2)-mu(2)/sig(2);
x_pre(3)=x_pre(3)-mu(3)/sig(3);
y_pre=[0 0 0 0];

% group 1
alpha=1.0;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
end
figure
plot(0:num-1,Jtheta,'b-','LineWidth',2);
y_pre(1)=x_pre*theta;
```

预测结果为2.0433e+09

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theta为最优的

最终计算得到的结果保留四位小数，该方法结果与学习率为 1.0 的结果相等

结论分析与体会：在调整学习率过程中使用了 for 循环，时间代价大，后通过二分法不断细化较快的确定了最优，通过本次实验，对梯度下降方法有了进一步的了解，在合适的学习率下该线性回归可以极快的收敛

附录：程序源代码

ex2.m

```
clear,clc;
%% load data
x=load('ex2x.dat');
y=load('ex2y.dat');

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

%% process
sig=std(X);
mu=mean(X);
X(:,2)=(X(:,2)-mu(2))/sig(2);
X(:,3)=(X(:,3)-mu(3))/sig(3);

%% pre-data
x_pre=[1 1650 3];
x_pre(2)=x_pre(2)-mu(2)/sig(2);
x_pre(3)=x_pre(3)-mu(3)/sig(3);
y_pre=[0 0 0 0];

%% learning rate
theta0=rands(3,1);
num=50;

% group 1
```

```

alpha=1.0;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
end
figure
plot(0:num-1,Jtheta,'b-','LineWidth',2);
y_pre(1)=x_pre*theta;
Jtheta(num)
theta

% group 2
alpha=0.5;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
end
hold on
plot(0:num-1,Jtheta,'r-','LineWidth',2);
y_pre(2)=x_pre*theta;
Jtheta(num)

% group 3
alpha=0.2;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;
end
hold on
plot(0:num-1,Jtheta,'k-','LineWidth',2);
y_pre(3)=x_pre*theta;
Jtheta(num)

% group 4
alpha=0.1;
Jtheta=zeros(num,1);
theta=theta0;
for k=1:num
    Jtheta(k)=(X*theta-y)'*(X*theta-y)/(2*m);
    theta=theta-alpha*sum((X*theta-y).*(X))'/m;

```

```

end
hold on
plot(0:num-1,Jtheta,'g-','LineWidth',2);
y_pre(4)=x_pre*theta;
Jtheta(num)

% plot
xlabel('number of iterations');
ylabel('cost J');
legend('lr=1.0','lr=0.5','lr=0.2','lr=0.1')
title('loss of different learning-rate')

%% train(y=Xw,w=inv(X'X)*X')
theta0=pinv(X)*y
real_y_pre=x_pre*theta0;

%% show the result
y_pre
real_y_pre

```