《数据挖掘导论》

（Multivariate Linear Regression）

学 院 名 称 ： 数据科学与计算机学院

专业（班级）： 16 软件工程电子政务

学 生 姓 名 ： 孙肖冉

学 号 ： 16340198

时 间 ： 2019 年 4 月 1 日

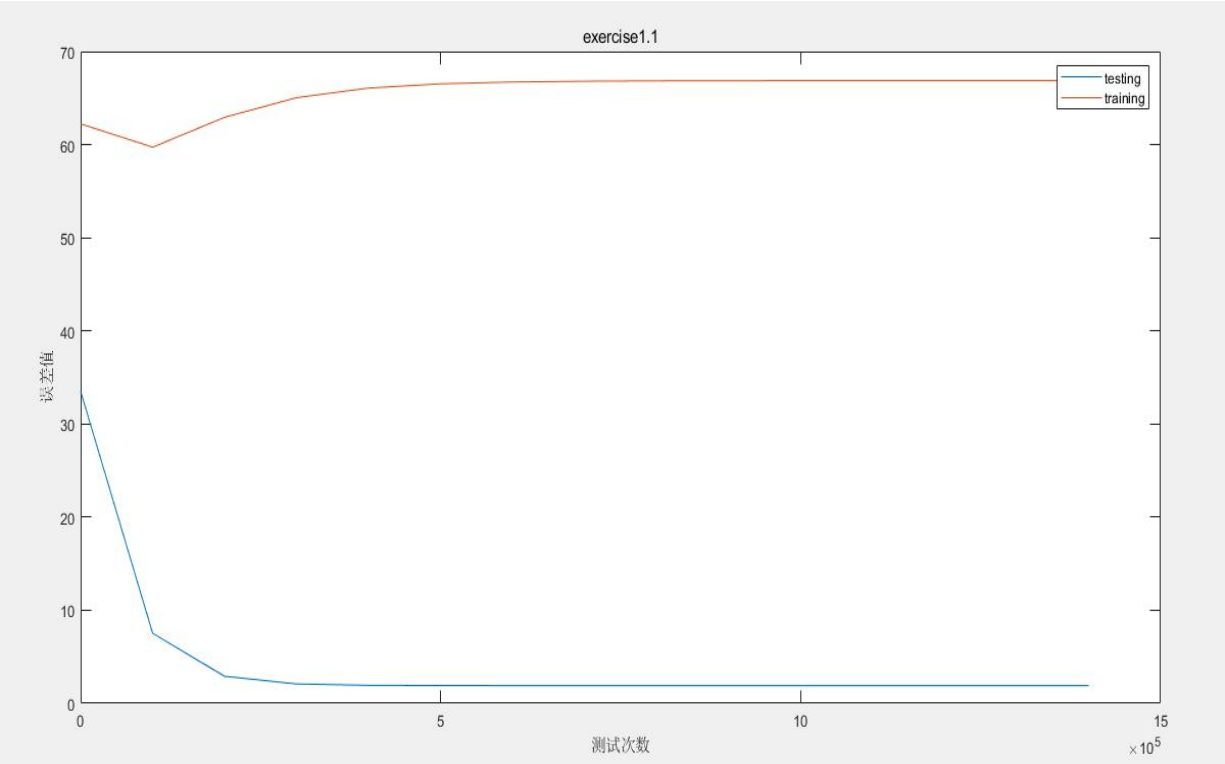
Exercise 1: How many parameters do you use to tune this linear regression model? Please use Gradient Descent to obtain the optimal parameters. Before you train the model, please set the number of iterations to be 1500000, the learning rate to 0.00015, the initial values of all the parameters to 0.0. During training, at every 100000 iterations, i.e., 100000 , 200000,…, 1500000, report the current training error and the testing error in a figure (you can draw it by hands or by any software). What can you find in the plots? Please analyze the plots.
Exercise 1: 你需要用多少个参数来训练该线性回归模型？请使用梯度下降方法训练。训练时，请把迭代次数设成1500000，学习率设成0.00015，参数都设成0.0。在训练的过程中，每迭代100000步，计算训练样本对应的误差，和使用当前的参数得到的测试样本对应的误差。请画图显示迭代到达100000步、200000步、… … 1500000时对应的训练样本的误差和测试样本对应的误差（图可以手画，或者用工具画图）。从画出的图中，你发现什么？请简单分析。

批量梯度下降法每一次迭代都需要对所有的样本进行处理计算。

关键代码：

```
%读取文件数据
data = load('dataForTraining.txt');
data2 = load('dataForTesting.txt');

%读取自变量数据
x = data(:,1:2);
x2 = data2(:,1:2);
%读取因变量数据
y = data(:,3);
y2 = data2(:,3);
m = length(y);
m2 = length(y2);

%增添一列为1的数据，计算对应的b
x = [ones(m,1),x];
x2 = [ones(m2,1),x2];

theta = zeros(3,1);%系数；第一个对应bas，第二个对应面积系数，第三个距离
alpha = 0.0002;   %自定义的学习率
max_iter = 1500000;  %自定义的迭代次数
%记录误差值
error = zeros(1500000,1);
error2 = zeros(1500000,1);

%进行迭代
for i = 1:max_iter
    %改变参数
     theta = theta - alpha/m * x' * ( x * theta - y) ;
     %计算误差值
     error(i) = sum((x * theta - y).^2) / (2*m);
     error2(i) = sum((x2*theta - y2).^2) /(2*m2);
end
```

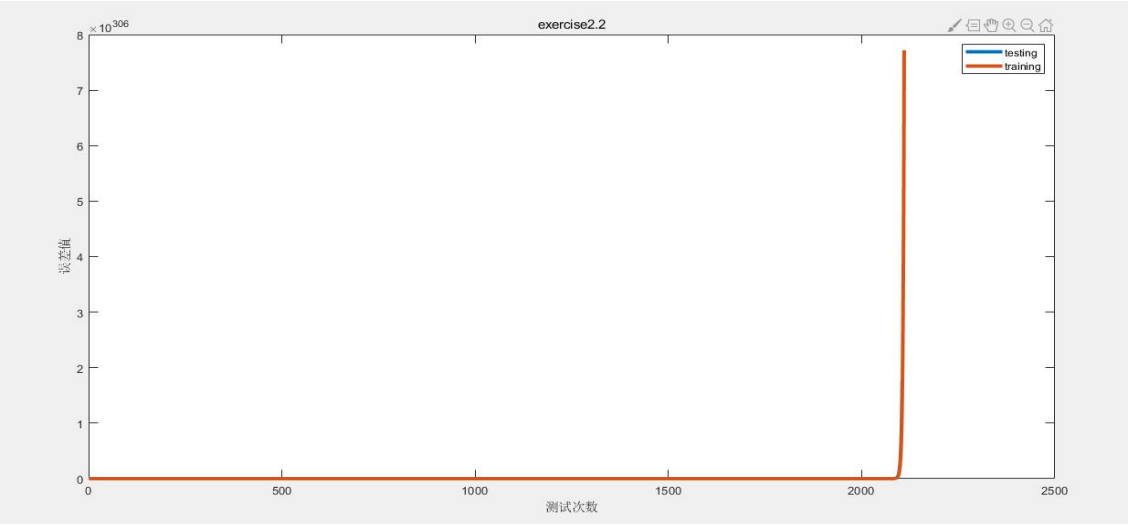
以100000次迭代为间隔



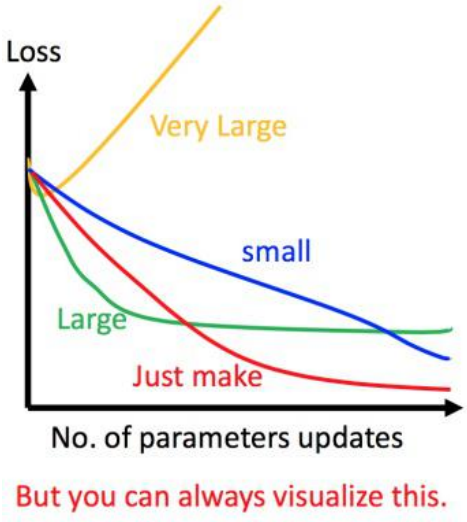
Exercise 2: Now, you change the learning rate to a number of different values, for instance, to 0.0002 (you may also change the number of iterations as well) and then train the model again. What can you find? Please conclude your findings.
Exercise 2: 现在，你改变学习率，比如把学习率改成0.0002（此时，你可以保持相同的迭代次数也可以改变迭代次数），然后训练该回归模型。你有什么发现？请简单分析。

将exercise1中所用代码的学习率改为0.0002进行计算处理，得到的误差值中会出现NAN的情况，意味着此时得到的误差值过大以致溢出。

当以1个迭代为间隔的时候就可以得到下图：



这个结果说明了，利用本次数据进行梯度下降法多变量的线性拟合时，使用0.002的学习率改变参数无法得到最优解。下图粗略地描述了学习率与误差值的关系，如下图所示当学习率过大时，会出现误差值飞涨的情况。



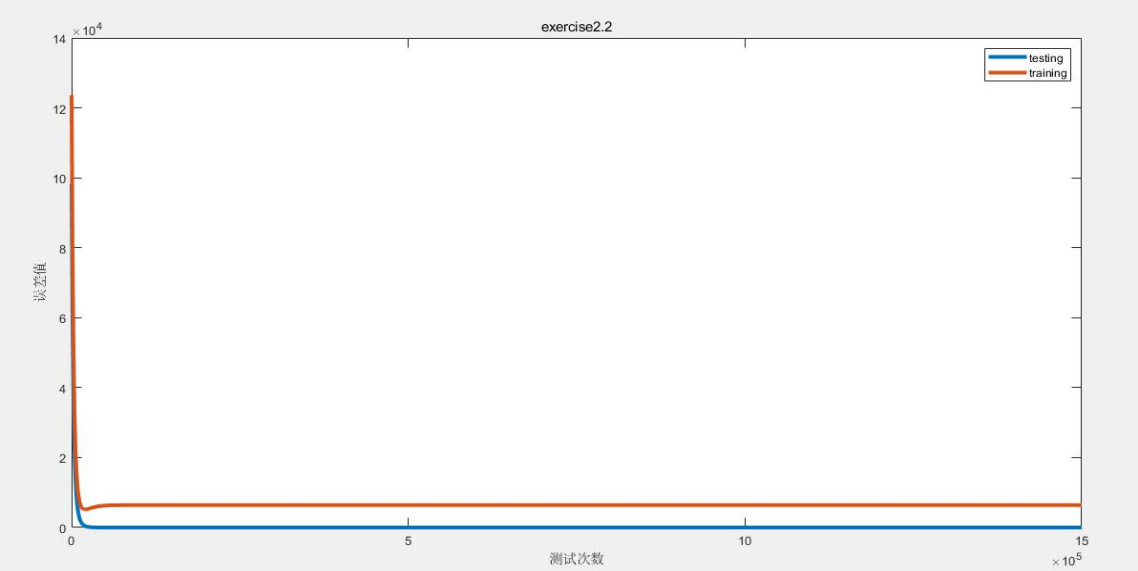
学习率过大或者过小对于模型的收敛都会有一定的影响，如果学习率设置的过大可能会导致模型无法收敛，如果当迭代的次数足够大的时候，还可能会导致因为代价函数的损失值太大造成内存溢出。如果，学习率设置的过小会导致，模型的收敛速度太慢，从而我们需要提高模型的迭代次数，才能使模型收敛，这样会造成对硬件资源和时间的浪费。

如果我们预先对数据进行标准化处理，则可以在学习率为0.0002的情况下测试函数仍然是收敛的。

预处理的代码：

```
function [ X_norm,avg,sigma ] = normalize(X)
avg = mean(X,1);
sigma = std(X,1);
X_norm  = (X - repmat(avg,size(X,1),1)) ./  repmat(sigma,size(X,1),1);
end
```

以一次迭代为间隔的结果图：



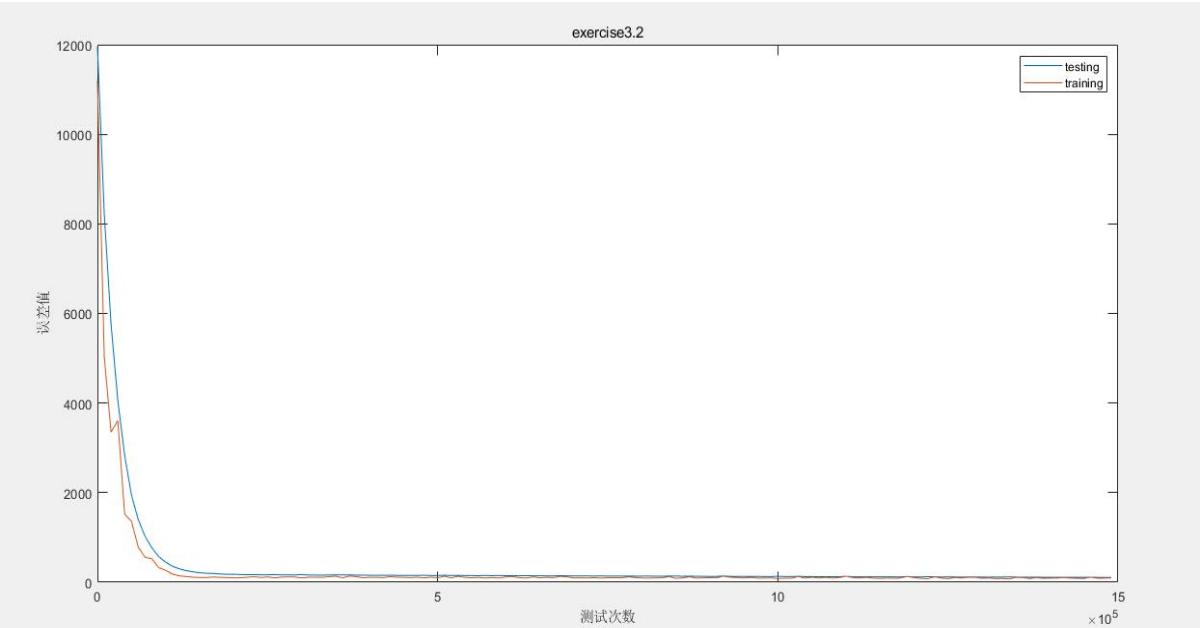
Exercise 3: Now, we turn to use other optimization methods to get the optimal parameters. Can you use Stochastic Gradient Descent to get the optimal parameters? Plots the training error and the testing error at each K-step iterations (the size of K is set by yourself). Can you analyze the plots and make comparisons to those findings in Exercise 1?
Exercise 3: 现在，我们使用其他方法来获得最优的参数。你是否可以用随机梯度下降法获得最优的参数？请使用随机梯度下降法画出迭代次数（每K次，这里的K你自己设定）与训练样本和测试样本对应的误差的图。比较Exercise 1中的实验图，请总结你的发现。

随机梯度下降法与批量下降法不同之处在于，和批量梯度有所不同的地方在于，每次迭代只选取一个样本的数据，一旦到达最大的迭代次数或是满足预期的精度，就停止。在本次测试中，我进行了1500000次迭代。

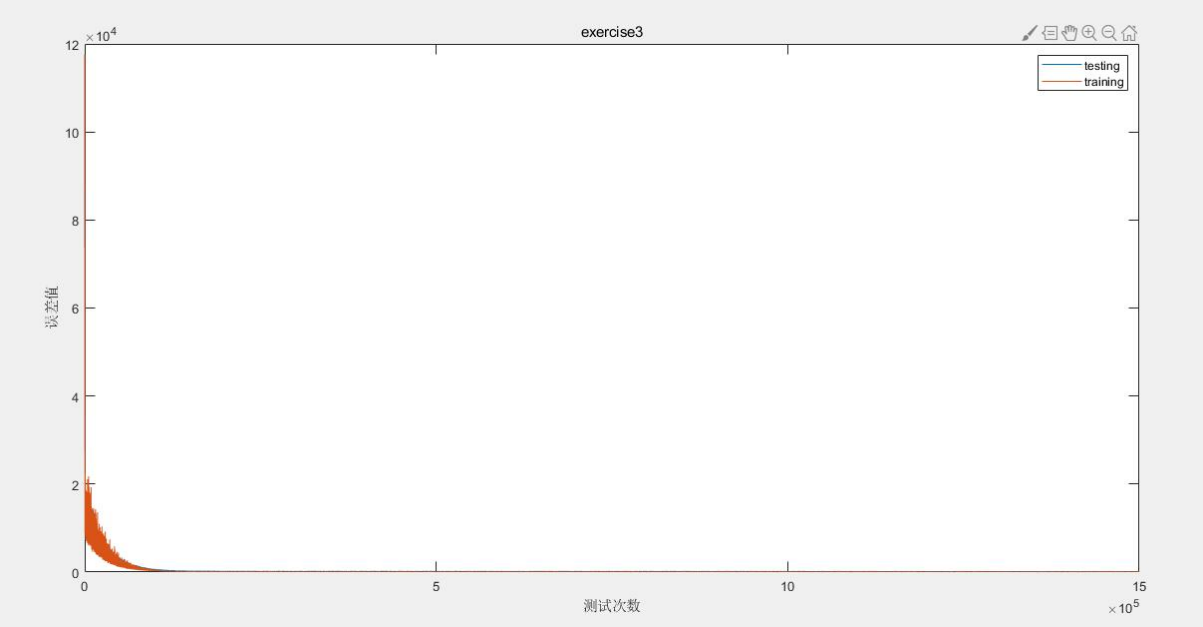
关键代码：

```
while true
        i = randi(m);%取1-m的随机值
        x_temp = x(i,:);
        y_temp = y(i,:);
        %改变参数
        theta = theta - alpha/m * x_temp' * ( x_temp * theta - y_temp) ;
        %计算误差值
        error(iter) = sum((x * theta - y).^2) / (2*m);
        error2(iter) = sum((x2*theta - y2).^2) /(2*m2);
        iter = iter + 1;
        if iter > max_iter
            break;
        end
end
```

因为取值存在一定的随机性，为了得到更明确的结果，我进行了多次测试，并给出其中两个测试对应的结果图。
测试一：
①以10000迭代为间隔：

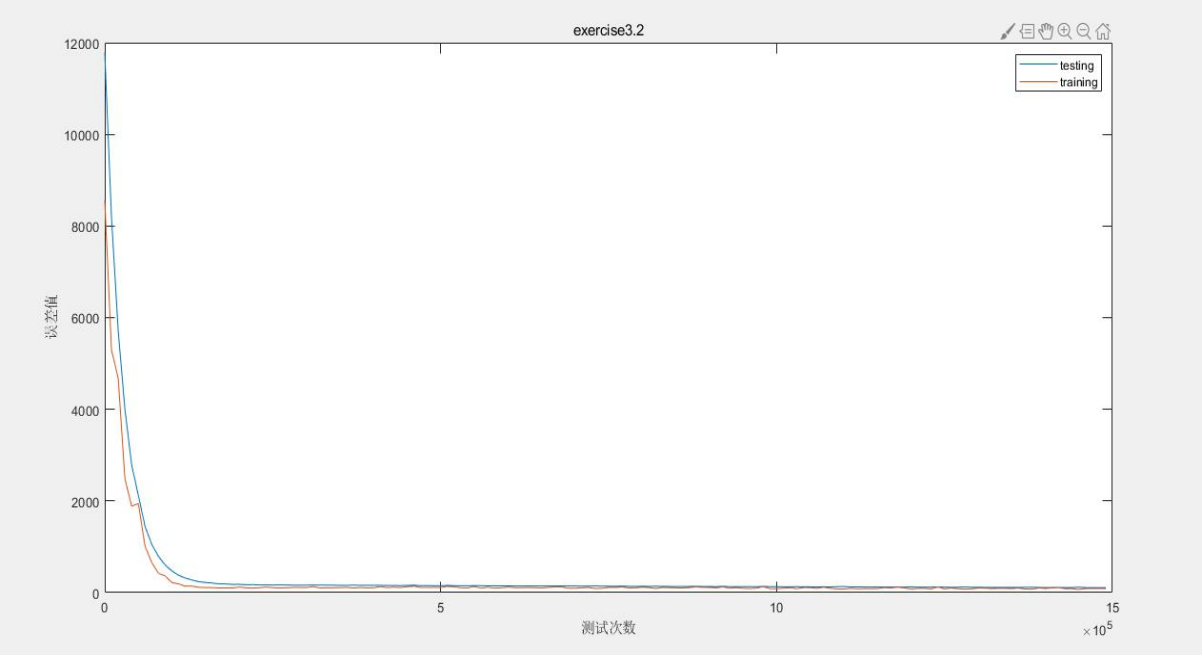


②以1次迭代为间隔

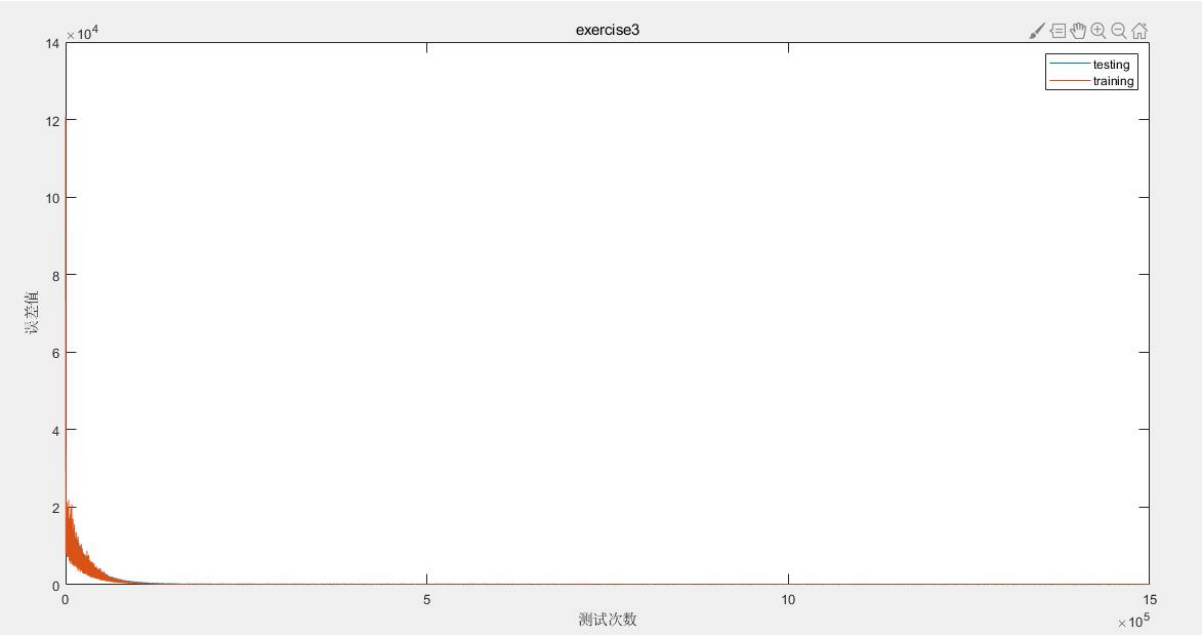


测试二：

①：以10000个迭代为间隔



②以1次迭代为间隔



可以看出在这两次测试中，得到的误差值连线存在波动情况且不同测试之间的波动差异很大，但是整体趋势与exercise1中的结果基本吻合：在迭代次数较少时会误差值很大，且下降很快，当迭代次数达到一定时，误差值的变化趋于平稳。

在数据量过多时，使用批量梯度下降法会使得计算耗费大量资源，此时可以选用随机梯度下降法进行代替，但同时要注意观察训练过程中误差的波动情况，尽可能避免大程度的波动造成的结果错误。