華東理工大學

模式识别大作业

题　　目　房屋价格预测

学　　院　信息科学与工程

专　　业　控制科学与工程

组　　员　艾奇辉

指导教师　赵海涛

完成日期：　2019 年 12 月 6 日

模式识别作业报告——房屋价格预测

组员：艾奇辉

经过这段时间模式识别课程的学习，对于该方向的一些经典模型与算法有了一定的初步认识。借由本次作业，正好可以锻炼我对于算法的实际运用的能力。本次作业主要是利用线性回归来进行房屋价格的预测，以下为详细过程。

一、房屋价格预测

房屋的价格一直都与房屋大小存在着密切的联系，房屋的价格处于一种波动的状态。当人们想要购买房屋时，如果可以根据想要购买的房屋的平方数，提前预估出房屋的价格，那么就可以更加准确地去做一些前期的准备。

本文就将对一组给定的房屋大小与价格的数据进行处理，然后作出线性回归模型，从而可以方便地知晓不同平方数的预测房屋价格。

二、整体解决方案

本次的房屋价格预测给定的数据包含了房屋的大小即平方数，还有就是房屋的价格。本次实验主要目的就是对于这两列数据进行分析，从而得到合适的回归模型，然后利用该回归模型对于想要的房屋平方数的价格进行预测。

2.1 数据分析

本次的数据的原始类型如表 1 所示，提取了前五行数据进行展示。

表 1 原始数据

	House square	House price
1	210	39990
2	160	32990
3	240	36900
4	141	23200
5	300	53990

表中的数据第一列是数据的编号，这对于接下来要求的线性回归模型并未影响，所以删除该列。

第二列数据是房屋的大小的数据，第三列数据是房屋的价格。这两列数据是本次求解回归模型的关键数据。

2.2 数据处理

因为本次求解的数据有一些数字较大，为了更加方便的计算，故对数据进行规范化处理，也就是进行均值和标准差的求解，进而利用下列公式对数据进行处理。

$$x=\frac{x-a}{b},a是数据x的平均值，b是数据的标准差$$

规划范的代码如下：

```
function [X_norm, jun, sigma] = featureNormalize(X)
X_norm = X;
jun = zeros(1, size(X, 2));     %size（x，2)返回矩阵列数
sigma = zeros(1, size(X, 2));
jun = sum(X) / length(X);
sigma = std(X);

    for i=1:length(X)
        X_norm(i,:) = (X(i,:) - jun) ./ sigma;
    end
end
```

在数据规范化后，绘制出房屋大小与房屋价格的离散点图。

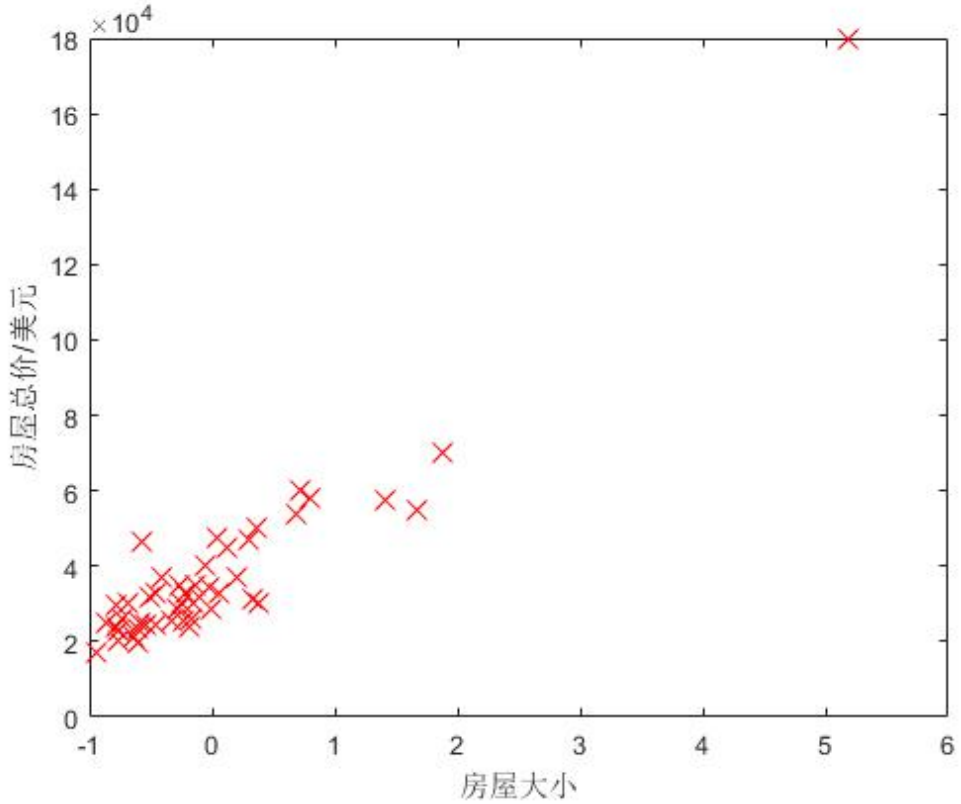

图 1 数据规范化的离散点图

2.3 线性回归模型的建立

在获得了以上的数据之后，我们采用线性回归方程来对数据进行模型建立。以下为线性回归梯度下降算法的过程。

假设我们有一个数据集包含 n 个数据对$(x_i,y_i),i=1,2,...,n$，其中x_i是自变量，$x_i \in R^{1\times d}$，

y_i是因变量，$y_i \in R$。x_i可以认为是我们在生产过程中获得的一些传感器和仪表的数据构成的向量，比如x_i的各个维度分别对应压力，温度，浓度等等，i 表示采样的时刻，y_i是对应x_i的某个指标，比如产品的质量指标等，它是连续的一个数值。我们希望有一个模型 $f(x_i,\beta)$ 可以预测未来的y_i，即$y_i = f(x_i,\beta)$（其中$x_i \in R^{1\times d}$，$\beta \in R^{d\times 1}$）。简单地说，这个模型应该“非常好”地适合于已有的数据，我们令误差$\varepsilon_i = y_i - f(x_i,\beta)$。线性回归的目标就是要求取$\beta$来极小化$\varepsilon_i$的平方和

$$J(\beta) = \frac{1}{n}\sum_{i=1}^{n} {\varepsilon_i}^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} [y_i - f(x_i,\beta)]^2$$

上式即为线性回归中的目标函数.

通常，将 $f(x_i,\beta)$进行简化，最简单的，$f(x_i,\beta)$可以看作是一个超平面，即 $f(x_i,\beta) = x_i\beta + \beta_0$，其中$\beta = (\beta_1,\ldots,\beta_d)^T$。则有方程

$$\begin{Bmatrix} y_1 = \beta_0 + x_1\beta + \varepsilon_1 \\ y_2 = \beta_0 + x_2\beta + \varepsilon_2 \\ \vdots \\ y_n = \beta_0 + x_n\beta + \varepsilon_n \end{Bmatrix}$$

令 $Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$，$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1d} \\ 1 & x_{21} & x_{22} & \ddots & x_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \ldots & x_{nd} \end{bmatrix}$，$E = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$，$B = \begin{bmatrix} \beta_0 \\ \beta \end{bmatrix}$,则有

$$Y = XB + E$$

$$J(B) = \frac{1}{n}\sum_{i=1}^{n} \left[y_i - \left(\sum x_{ij}\beta_j + \beta_0\right)\right]^2 = \frac{1}{n}(Y - XB)^T(Y - XB)$$

β_0的作用主要是产生一个偏置，如$\sum x_i = 0$，而$\sum y_i \neq 0$ 时，那么在 y 轴上就会产生一个截距，这是线性回归问题优化时要考虑的。

求解 $J(B)$的最小值有几种方法，下面用求导数的方式来求解，即用梯度下降法来求解。

$$J(B) = \frac{1}{n}(Y - XB)^T(Y - XB)$$

$$= \frac{1}{n}(Y^TY - Y^TXB - B^TX^TY + B^TX^TXB)$$

$$= \frac{1}{n}(Y^TY - 2B^TX^TY + B^TX^TXB)$$

$$\nabla J(B) = \frac{2}{n}(-X^TY + X^TXB)$$

因此线性回归问题可用梯度下降法来解。下面我们给出梯度下降法进行求解线性回归问题的一般步骤：

①给定初值$B_1, i=1$，学习率α，给定阈值ε；；

②求$B_{i+1} = B_i - \alpha \nabla J(B_i)$ ；

③若$\| J(B_{i+1}) - J(B_i) \|_2^2 \leq \varepsilon$，转向④，否则，$i = i+1$，转向②

④输出B_{i+1}.

算法代码如下：

```
function [theta, J] = gradientDescent(X, y, theta, alpha, iterations)
m = length(y); %  数据集的长度
J = zeros(iterations, 1);

for iter = 1:iterations
    theta = theta - (alpha/m)*(((X*theta) - y)'*X)';
    J(iter) = computeloss(X, y, theta);%存储每一次迭代的 J

end
End
```

之后得到的线性回归直线如下图所示：

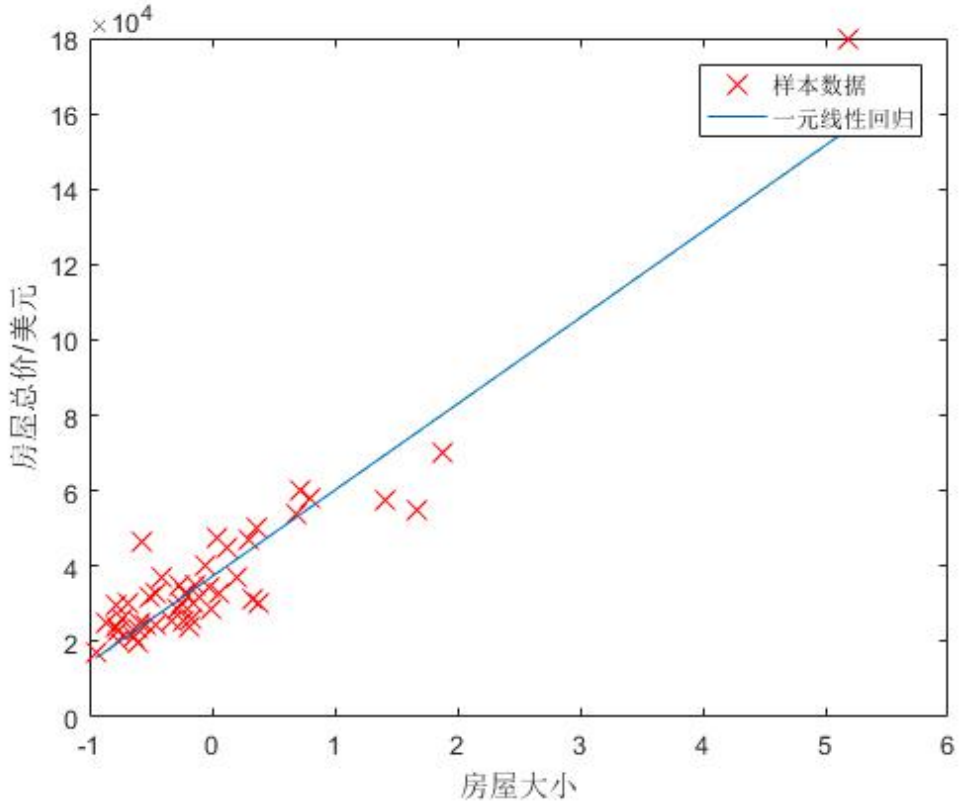

图 2 线性回归方程

此外，样本预测模型与实际值之间难免有误差，因此定义损失函数，公式如下

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2$$

代码如下：

```
function J = computeloss(X, y, theta)
m = length(y);
J = 0;
  predictions = X*theta;
    sqrErrors = (predictions-y).^2;
    J = 1/(2*m) * sum(sqrErrors);

end
```

损失函数在每一次迭代过程中会产生变化，下图为损失函数的变化图：

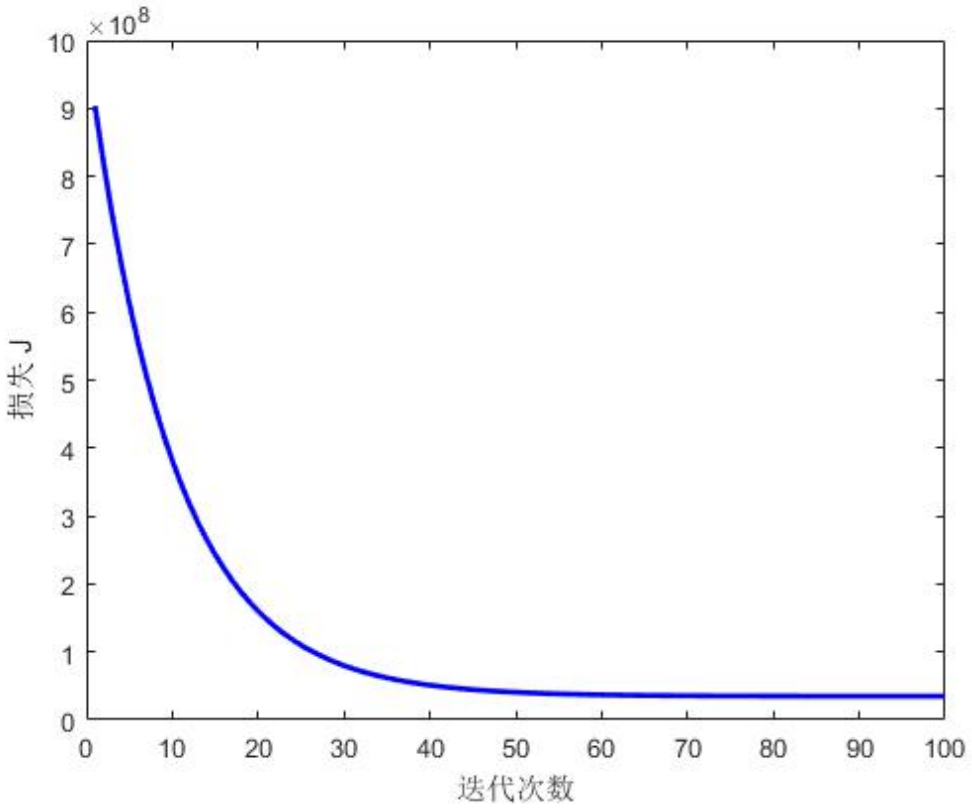

图 3 损失函数

可以看到，随着迭代次数的增加，损失越来越小，也就意味着数据的预测值与真实值的差距越来越小，也就越来越准确。

2.4 主体程序

做完了前期的准备工作，接下来就是利用该线性回归模型对于房屋的价格进行预测，代码如下：

```
clear ; close all; clc
fprintf('Loading data ...\n');

%加载数据
data = load('ex1data3.txt');
X = data(:,1);
y = data(:,2);
m = length(y);

% 显示一些样本情况
fprintf('数据中的最先五个样本: \n');
fprintf(' x = %.0f , y = %.0f \n', [X(1:5,:) y(1:5,:)]');%不显示小数点后的数
fprintf('程序暂停. 按动继续.\n');
pause;
% 缩放特征并求零均值
fprintf('规范化特征 ...\n');
[X, jun, sigma] = featureNormalize(X);
plotData(X,y);
% 添加项
X = [ones(m, 1) X];
% ================ Part 2: 梯度下降计算 ================

fprintf('运行梯度下降...\n');
% 选取一些参数值
alpha = 0.05;
iterations = 100;

%初始化 Theta
theta = zeros(2, 1);

[theta, J] = gradientDescent(X, y, theta, alpha, iterations);

%绘制线性回归方程
hold on
plot(X(:,2),X*theta)
legend('样本数据','一元线性回归');

%绘制收敛图
figure;

plot(1:numel(J), J, '-b', 'LineWidth', 2);
```

```
xlabel('迭代次数');
ylabel('损失  J');

% 梯度下降结果
fprintf('由梯度下降计算的 theta: \n');
fprintf(' %f \n', theta);
fprintf('\n');

% 估算 160 平方英尺房屋的价格
d = 160;
d = (d - jun)./sigma;%规范化
d = [ones(1, 1) d];
price = d * theta;

fprintf(['160 平方英尺的住宅的预测价格 ' ...
         '(梯度下降法):\n $%f\n'], price);
```

最后选取了几组预测的数据如下表

表 2 预测数据

House square	House price
313	55374.003506
149	24748.560552
200	34272.326349

之后将预测的数据与原数据进行一个比较

表 3 预测数据与原数据对比图

House square	House price（原始）	House price（预测）	差值
313	57990	55374.003506	2,615.996494
149	24250	24748.560552	-498.560552
200	34700	34272.326349	427.673651

可以看出，本次的线性回归模型预测的数据的准确度还是有着一定保障，误差在可以接受的范围之内。

三、小组分工

程序设计及编写：艾奇辉

程序调试：艾奇辉

实验报告：艾奇辉

四、作业总结

由于是第一次接触这种类型的大作业，在刚开始进行的时候会有一些无从下手。之后从网上查阅了一些资料，然后总结了一些别人的经验，最终编写出了合适的程序。在编写的过程中，有一些参数的选取刚开始会有一些问题，像是梯度下降算法中的 alpha 和迭代次数，这些参数在一次一次地调整过程中得到了一定的改善，最终呈现出较为完善的结果。

通过这次作业我也发现自身在编程方面存在着很多的问题，感谢赵老师给予的这次机会，我也会在以后继续加强自己的编程练习。