Programming Exercise 1: Linear Regression
编程练习 1:线性回归

Machine Learning

机器学习

Introduction
介绍

In this exercise, you will implement linear regression and get to see it work on data. Before starting on this programming exercise, we strongly recommend watching the video lectures and completing the review questions for the associated topics.

在本练习中，您将实现线性回归，并了解它对数据的作用。在开始编程练习之前，我们强烈建议观看视频讲座，并完成相关主题的复习题。

To get started with the exercise, you will need to download the starter code and unzip its contents to the directory where you wish to complete the exercise. If needed, use the cd command in Octave/MATLAB to change to this directory before starting this exercise.

要开始练习，您需要下载启动代码并将其内容解压缩到您希望完成练习的目录中。如果需要，在开始这个练习之前，使用 Octave/MATLAB 中的 cd 命令切换到这个目录。

You can also find instructions for installing Octave/MATLAB in the "Environment Setup Instructions" of the course website.

您也可以在课程网站的“环境设置说明”中找到安装 Octave/MATLAB 的说明。

Files included in this exercise
本练习中包含的文件

ex1.m - Octave/MATLAB script that steps you through the exercise
ex1_multi.m - Octave/MATLAB script for the later parts of the exercise
ex1data1.txt - Dataset for linear regression with one variable
ex1data2.txt - Dataset for linear regression with multiple variables
submit.m - Submission script that sends your solutions to our servers
[Λ] warmUpExercise.m - Simple example function in Octave/MATLAB
[Λ] plotData.m - Function to display the dataset

ex1.m - Octave/MATLAB 脚本，指导您完成练习 ex1 multi.m - Octave/MATLAB 脚本，用于练习后面的部分 ex1data1.txt -用于单变量线性回归的数据集 ex1data2.txt -用于多变量线性回归的数据集 submit.m -将您的解决方案发送到我们的服务器 [Λ]warmupexercise . m-octave/MATLAB 中的简单示例函数[Λ] plotData.m -用于显示数据集的函数
[Λ] computeCost.m - Function to compute the cost of linear regression
[Λ]计算线性回归成本的 computeCost.m 函数
[Λ] gradientDescent.m - Function to run gradient descent
[Λ] gradientDescent.m -运行梯度下降的函数
[†] computeCostMulti.m - Cost function for multiple variables
[computecostmulti . m ——多变量的成本函数
[†] gradientDescentMulti.m - Gradient descent for multiple variables
[+gradientdescentmulti . m—多个变量的梯度下降
[†] featureNormalize.m - Function to normalize features
[+feature normalize . m—归一化特征的函数
[†] normalEqn.m - Function to compute the normal equations
[+normal eqn . m—用于计算法线方程的函数

Λ indicates files you will need to complete

Λ表示您需要完成的文件
† indicates optional exercises
表示可选练习

Throughout the exercise, you will be using the scripts ex1.m and ex1 multi.m.

在整个练习中，您将使用脚本 ex1.m 和 ex1 multi.m。

These scripts set up the dataset for the problems and make calls to functions that you will write. You do not need to modify either of them. You are only required to modify functions in other files, by following the instructions in this assignment.

这些脚本为问题建立数据集，并调用您将要编写的函数。您不需要修改它们中的任何一个。你只需要按照本作业中的说明修改其他文件中的函数。

For this programming exercise, you are only required to complete the first part of the exercise to implement linear regression with one variable. The second part of the exercise, which is optional, covers linear regression with multiple variables.

对于本编程练习，您只需完成练习的第一部分，即可实现一个变量的线性回归。练习的第二部分是可选的，包括多个变量的线性回归。

Where to get help
从哪里获得帮助

The exercises in this course use Octave[1] or MATLAB, a high-level programming language well-suited for numerical computations. If you do not have Octave or MATLAB installed, please refer to the installation instructions in the "Environment Setup Instructions" of the course website.

本课程中的练习使用八度音阶[1] 或者 MATLAB，一种非常适合数值计算的高级编程语言。如果您没有安装 Octave 或 MATLAB，请参考课程网站"环境设置说明"中的安装说明。

At the Octave/MATLAB command line, typing help followed by a func- tion name displays documentation for a built-in function. For example, help plot will bring up help information for plotting. Further documentation for Octave functions can be found at the Octave documentation pages. MAT- LAB documentation can be found at the MATLAB documentation pages.

在 Octave/MATLAB 命令行中，键入 help 后跟一个函数名会显示内置函数的文档。例如，"帮助绘图"将显示绘图的帮助信息。关于 Octave 函数的更多文档可以在 Octave documentation pages。MAT- LAB 文档可以在 MATLAB documentation pages。

We also strongly encourage using the online **Discussions** to discuss exercises with other students. However, do not look at any source code written

by others or share your source code with others.

我们也强烈鼓励使用在线讨论与其他学生讨论练习。但是，不要看别人写的任何源代码，也不要和别人分享你的源代码。

---

1 Simple Octave/MATLAB function

2 简单的倍频程/MATLAB 函数

The first part of ex1.m gives you practice with Octave/MATLAB syntax and the homework submission process. In the file warmUpExercise.m, you will find the outline of an Octave/MATLAB function. Modify it to return a 5 x 5 identity matrix by filling in the following code:

ex1.m 的第一部分让你练习 Octave/MATLAB 语法和作业提交过程。在 warmUpExercise.m 文件中，您将找到 Octave/MATLAB 函数的概要。通过填写以下代码，将其修改为返回一个 5 x 5 的单位矩阵:

```
A = eye(5);
```

---

[1] Octave is a free alternative to MATLAB. For the programming exercises, you are free to use either Octave or MATLAB.

[1] Octave 是 MATLAB 的免费替代品。对于编程练习，您可以自由使用 Octave 或 MATLAB。

When you are finished, run ex1.m (assuming you are in the correct directory, type "ex1" at the Octave/MATLAB prompt) and you should see output similar to the following:

完成后，运行 ex1.m(假设您在正确的目录中，在 Octave/MATLAB 提示符下键入“ex1 ”),您应该会看到类似下面的输出:

```
ans =
ans =


Diagonal Matrix
对角矩阵


   1   0   0   0   0
   0   1   0   0   0
   0   0   1   0   0
   0   0   0   1   0
   0   0   0   0   1
   一   0   0   0   0
   0   一   0   0   0
   0   0   一   0   0
   0   0   0   一   0
   0   0   0   0   一
```

Now ex1.m will pause until you press any key, and then will run the code for the next part of the assignment. If you wish to quit, typing **ctrl-c** will stop the program in the middle of its run.

现在 ex1.m 将暂停，直到你按下任何键，然后将运行赋值的下一部分的代码。如果你想退出，键入 ctrl-c 将会在程序运行过程中停止程序。

2.1 Submitting Solutions
2.2 提交解决方案

After completing a part of the exercise, you can submit your solutions for grading by typing submit at the Octave/MATLAB command line. The submission script will prompt you for your login e-mail and submission token and ask you which files you want to submit. You can obtain a submission token from the web page for the assignment.

完成部分练习后，您可以通过在 Octave/MATLAB 命令行中键入 submit 来提交您的解决方案以进行评分。子任务脚本将提示您输入登录电子邮件和提交令牌，并询问您要提交哪些文件。您可以从作业的网页获取提交令牌。

You should now submit your solutions.
您现在应该提交您的解决方案。

You are allowed to submit your solutions multiple times, and we will take only the highest score into consideration.
您可以多次提交您的解决方案，我们将只考虑最高分。

---

3 Linear regression with one variable
4 一元线性回归

In this part of this exercise, you will implement linear regression with one variable to predict profits for a food truck. Suppose you are the CEO of a restaurant franchise and are considering different cities for opening a new outlet. The chain already has trucks in various cities and you have data for profits and populations from the cities.

在本练习的这一部分中，您将使用一个变量来实现线性回归，以预测食品车的利润。假设你是一家特许经营餐厅的首席执行官，正在考虑在不同的城市开设新的分店。该连锁店已经在各个城市有了卡车，而且你有了这些城市的利润和人口数据。

You would like to use this data to help you select which city to expand to next.

您希望使用这些数据来帮助您选择下一个要扩展到的城市。

The file ex1data1.txt contains the dataset for our linear regression problem. The first column is the population of a city and the second column is the profit of a food truck in that city. A negative value for profit indicates a loss.

文件 ex1data1.txt 包含用于线性回归问题的数据集。第一列是一个城市的人口，第二列是该城市一辆快餐车的利润。利润为负值表示亏损。

The ex1.m script has already been set up to load this data for you.

ex1.m 脚本已经设置好为您加载这些数据。

4.1 Plotting the Data

4.2 绘制数据

Before starting on any task, it is often useful to understand the data by visualizing it. For this dataset, you can use a scatter plot to visualize the data, since it has only two properties to plot (profit and population). (Many other problems that you will encounter in real life are multi-dimensional and can't be plotted on a 2-d plot.)

在开始任何任务之前，通过可视化来理解数据通常是有用的。对于此数据集，您可以使用散点图来可视化数据，因为它只有两个属性可以绘制(利润和人口)。(你在现实生活中会遇到的许多其他问题是多维的，无法在二维图上绘制出来。)

In ex1.m, the dataset is loaded from the data file into the variables X and y:

在 ex1.m 中，数据集从数据文件加载到变量 X 中和 y:

```
data = load('ex1data1.txt');        % read comma separated data
X = data(:, 1); y = data(:, 2); m = length(y);
```


```
data = load('ex1data1.txt');        % read comma separated data
X = data(:, 1); y = data(:, 2); m = length(y);
                                    % number of training examples
```


Next, the script calls the plotData function to create a scatter plot of

the data. Your job is to complete plotData.m to draw the plot; modify the file and fill in the following code:

接下来，脚本调用 plotData 函数来创建数据的散点图。你的工作是完成 plotData.m 来绘制绘图；修改文件并填入以下代码：

```
plot(x, y, 'rx', 'MarkerSize', 10); ylabel('Profit in $10,000s');
plot(x, y, 'rx', 'MarkerSize', 10); ylabel('Profit in $10,000s');
% Set the y-axis label
xlabel('Population of City in 10,000s');  % Set the x-axis label
xlabel('Population of City in 10,000s');  % Set the x-axis label
```

Now, when you continue to run ex1.m, our end result should look like Figure 1, with the same red "x" markers and axis labels.

现在，当您继续运行 ex1.m 时，我们的最终结果应该如图所示 1，具有相同的红色“x”标记和轴标签。

To learn more about the plot command, you can type help plot at the Octave/MATLAB command prompt or to search online for plotting documentation. (To change the markers to red "x", we used the option 'rx' together with the plot command, i.e., **plot(..,[your options here],.., 'rx');**)

要了解有关 plot 命令的更多信息，可以在 Octave/MATLAB 命令提示符下键入 help plot，或者在线搜索绘图文档。(为了将标记更改为红色“x ”，我们将选项“rx”与 plot 命令一起使用，即 plot(..，[您可以在此选择]，..， 'rx')；)

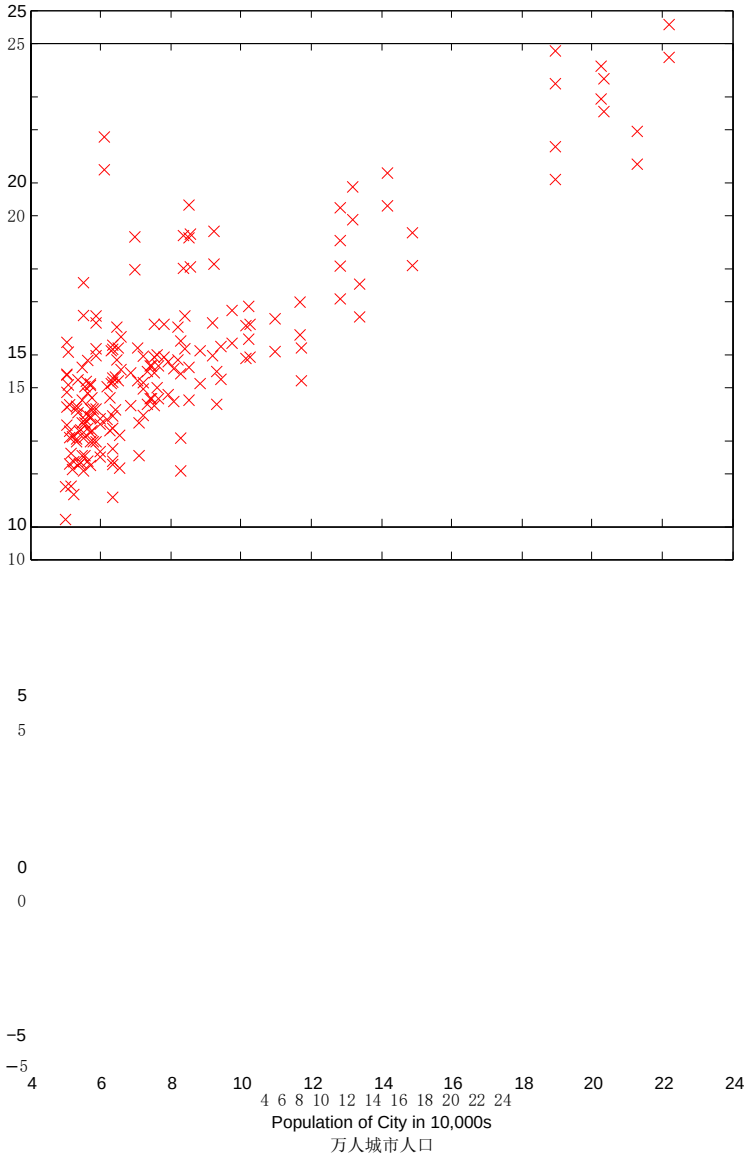


Figure 1: Scatter plot of training data
图 1: 训练数据的散点图

4.3 Gradient Descent

4.4 梯度下降

In this part, you will fit the linear regression parameters θ to our dataset using gradient descent.

在这一部分，您将使用梯度下降将线性回归参数 θ 拟合到我们的数据集。

4.4.1 Update Equations
4.4.2 更新方程式

The objective of linear regression is to minimize the cost function

线性回归的目标是最小化成本函数

$$J(\theta) = \frac{1}{} \Sigma \; h\ (x^{(i)}) - y^{(i)\ 2}$$

$$j(\theta) = 1\ \sigma\ h(x(I)) — y(I)2$$

where the hypothesis $h_\theta(x)$ is given by the linear model

其中假设 h θ (x) 由线性模型给出

$$h_\theta(x) = \theta^T x = \theta_0 + \theta_1 x_1$$

$$h\ \theta\ (x) = \theta\ T\ x = \theta\ 0 + \theta\ 1x1$$

Recall that the parameters of your model are the θ_j values. These are the values you will adjust to minimize cost J(θ). One way to do this is to use the batch gradient descent algorithm. In batch gradient descent, each iteration performs the update

回想一下，模型的参数是 θ j 值。这些是您将调整以最小化成本 J(θ) 的值。一种方法是使用批量梯度下降算法。在批量梯度下降中，每次迭代执行更新

$$\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} (h(x^{(i)}) - y^{(i)}) x^{(i)}$$ (simultaneously update θ for all j).

θ := θ − α (h(x) − y)x (同时更新所有 j 的 θ)。

With each step of gradient descent, your parameters θ_j come closer to the optimal values that will achieve the lowest cost J(θ).

随着梯度下降的每一步，参数 θ j 越来越接近实现最低成本 J(θ)的最佳值。

4.4.3 Implementation

Implementation Note: We store each example as a row in the the X matrix in Octave/MATLAB. To take into account the intercept term (θ_0), we add an additional first column to X and set it to all ones. This allows us to treat θ_0 as simply another 'feature'.

Implementation Note: We store each example as a row in the the X matrix in Octave/MATLAB. To take into account the intercept term (θ0), we add an additional first column to X and set it to all ones. This allows us to

4.4.4 履行

In ex1.m, we have already set up the data for linear regression. In the following lines, we add another dimension to our data to accommodate the θ_0 intercept term. We also initialize the initial parameters to 0 and the learning rate alpha to 0.01.

在 ex1.m 中，我们已经为线性回归设置了数据。在接下来的几行中，我们为数据增加了另一个维度，以适应 θ0 截距项。我们还将初始参数初始化为 0，并将学习率 α 初始化为 0.01。

```
X = [ones(m, 1), data(:,1)]; % Add a column of ones to x
theta = zeros(2, 1); % initialize fitting parameters
```

X = [ones(m, 1), data(:,1)]; % Add a column of ones to x theta = zeros(2, 1); % initialize fitting parameters

```
iterations = 1500;
alpha = 0.01;
```

4.4.5 Computing the cost J(θ)

4.4.6 计算成本 J(θ)

As you perform gradient descent to learn minimize the cost function J(θ), it is helpful to monitor the convergence by computing the cost. In this section, you will implement a function to calculate J(θ) so you can check the convergence of your gradient descent implementation.

当您执行梯度下降来学习最小化成本函数 J(θ)时，通过计算成本来监控收敛是有帮助的。在本节中，您将实现一个计算 J(θ)的函数，以便

检查梯度下降实现的收敛性。

Your next task is to complete the code in the file computeCost.m, which is a function that computes J(θ). As you are doing this, remember that the variables X and y are not scalar values, but matrices whose rows represent the examples from the training set.

您的下一个任务是完成文件 computeCost.m 中的代码，这是一个计算 J(θ)的函数。当您这样做时，请记住变量 X 和 y 不是标量值，而是矩阵，其行表示来自定型集的示例。

Once you have completed the function, the next step in ex1.m will run computeCost once using θ initialized to zeros, and you will see the cost printed to the screen.

一旦完成了函数，ex1.m 中的下一步将使用初始化为零的 θ 运行 computeCost 一次，您将看到成本打印到屏幕上。

You should expect to see a cost of 32.07.

你应该会看到 32.07 英镑的成本。

You should now submit your solutions.

您现在应该提交您的解决方案。

4.4.7 Gradient descent

4.4.8 梯度下降

Next, you will implement gradient descent in the file gradientDescent.m. The loop structure has been written for you, and you only need to supply the updates to θ within each iteration.

接下来，您将在文件 gradientDescent.m 中实现梯度下降。已经为您编写了循环结构，您只需要在每次迭代中提供对 θ 的更新。

As you program, make sure you understand what you are trying to optimize and what is being updated. Keep in mind that the cost J(θ) is parameterized by the vector θ, not X and y. That is, we minimize the value of J(θ) by changing the values of the vector θ, not by changing X or y. Refer to the equations in this handout and to the video lectures if you are uncertain.

当你编程的时候，确保你明白你要优化的是什么，更新的是什么。请记住，成本 J(θ)是由向量 θ 确定的，而不是 X 和 y。也就是说，我们通过改变向量 θ 的值来最小化 J(θ)的值，而不是改变 X 或 y。如果不确定，请参考讲义中的等式和视频讲座。

A good way to verify that gradient descent is working correctly is to look at the value of J(θ) and check that it is decreasing with each step. The starter code for gradientDescent.m calls computeCost on every iteration and prints the cost. Assuming you have implemented gradient descent and computeCost correctly, your value of J(θ) should never increase, and should converge to a steady value by the end of the algorithm.

验证梯度下降是否正常工作的一个好方法是查看 J(θ)的值，并检查它是否随每步下降。gradientDescent.m 的起始代码在每次迭代时调用 computeCost 并打印成本。假设您已经正确实现了梯度下降和 computeCost，您的 J(θ)值应该永远不会增加，并且应该在算法结束时收敛到一个稳定的值。

After you are finished, ex1.m will use your final parameters to plot the linear fit. The result should look something like Figure 2:

完成后，ex1.m 将使用您的最终参数绘制线性拟合。结果应该类似于图 2:

Your final values for θ will also be used to make predictions on profits in areas of 35,000 and 70,000 people. Note the way that the following lines in ex1.m uses matrix multiplication, rather than explicit summation or looping, to calculate the predictions. This is an example of code vectorization in Octave/MATLAB.

θ 的最终值也将用于预测 35，000 人和 70，000 人地区的利润。请

注意 ex1.m 中的下面几行使用矩阵乘法，而不是显式求和或循环来计算预测。这是 Octave/MATLAB 中代码矢量化的一个例子。

You should now submit your solutions.
您现在应该提交您的解决方案。

```
predict1 = [1, 3.5] * theta;
predict2 = [1, 7] * theta;
```

4.5 Debugging

4.6 排除故障

Here are some things to keep in mind as you implement gradient descent:

实施梯度下降时，请记住以下几点：

- Octave/MATLAB array indices start from one, not zero. If you're stor- ing θ_0 and θ_1 in a vector called theta, the values will be theta(1) and theta(2).
- Octave/MATLAB 数组索引从 1 开始，而不是零。如果你把 θ 0 和 θ 1 存储在一个叫做 θ 的向量中，那么值将是 θ (1)和 θ (2)。
- If you are seeing many errors at runtime, inspect your matrix operations to make sure that you're adding and multiplying matrices of compat- ible dimensions. Printing the dimensions of variables with the size command will help you debug.
- 如果您在运行时看到许多错误，请检查您的矩阵运算，以确保您正在添加和乘以兼容维度的矩阵。用 size 命令打印变量的尺寸将有助于调试。

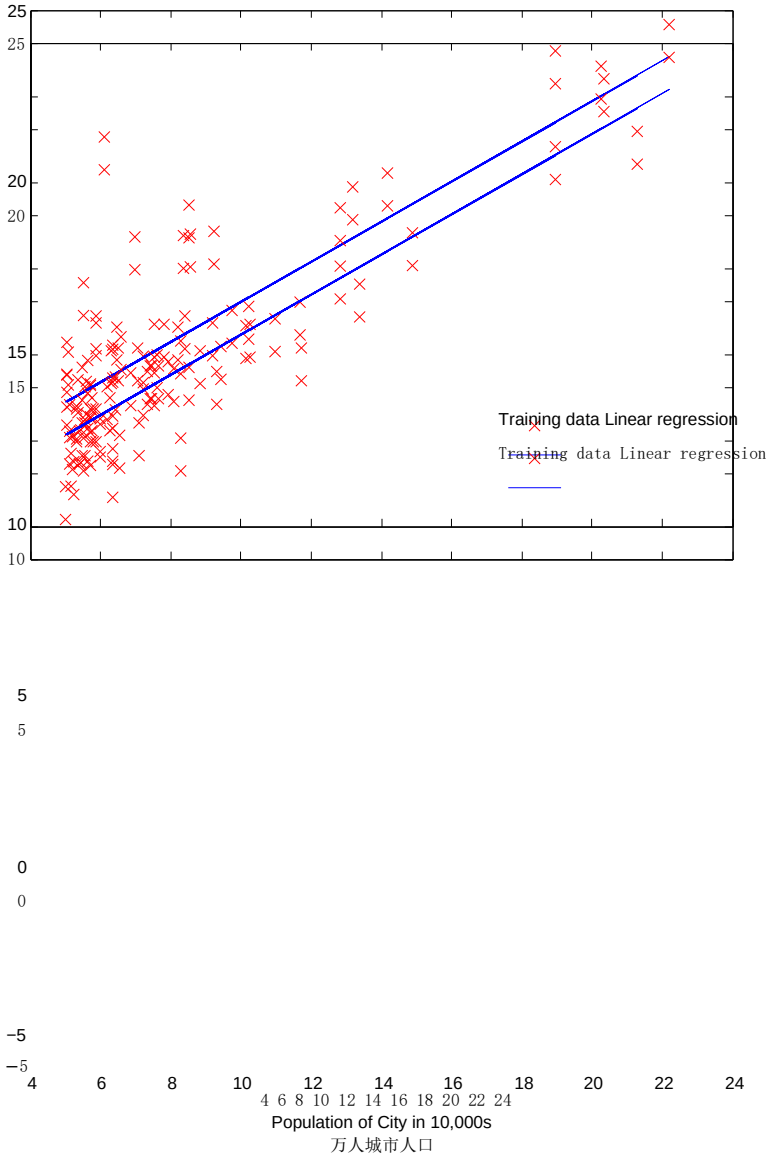


Figure 2: Training data with linear regression fit
图 2:线性回归拟合的训练数据

- By default, Octave/MATLAB interprets math operators to be matrix

operators. This is a common source of size incompatibility errors. If you don't want matrix multiplication, you need to add the "dot" notation to specify this to Octave/MATLAB. For example, A*B does a matrix multiply, while A.*B does an element-wise multiplication.

- 默认情况下，Octave/MATLAB 将数学运算符解释为矩阵运算符。这是尺寸不兼容错误的一个常见来源。如果你不想要矩阵乘法，你需要添加“点”符号来指定 Octave/MATLAB。例如，A*B 执行矩阵乘法，而 A.*B 执行元素级乘法。

4.7 Visualizing J(θ)

4.8 可视化 J(θ)

To understand the cost function J(θ) better, you will now plot the cost over a 2-dimensional grid of θ_0 and θ_1 values. You will not need to code anything new for this part, but you should understand how the code you have written already is creating these images.

为了更好地理解成本函数 J(θ),现在将在 θ0 和 θ1 值的二维网格上绘制成本。您不需要为这个部分编写任何新的代码，但是您应该理解您已经编写的代码是如何创建这些图像的。

In the next step of ex1.m, there is code set up to calculate J(θ) over a grid of values using the computeCost function that you wrote.

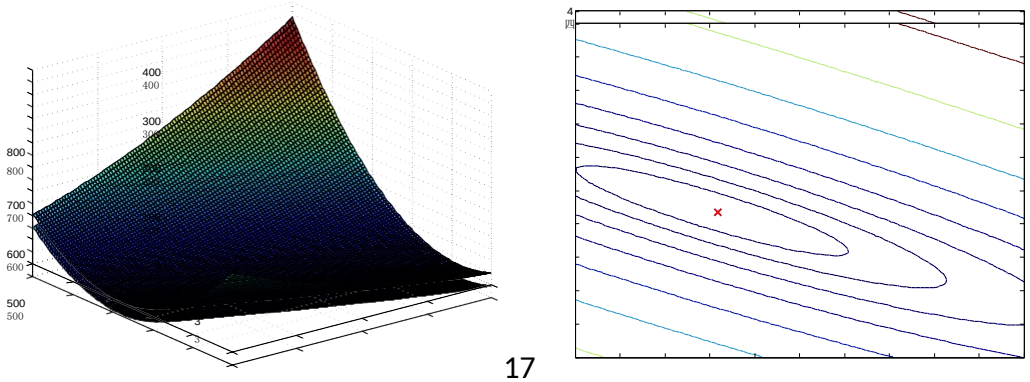
在 ex1.m 的下一步中，使用您编写的 computeCost 函数设置代码来计算数值网格上的 J(θ)。

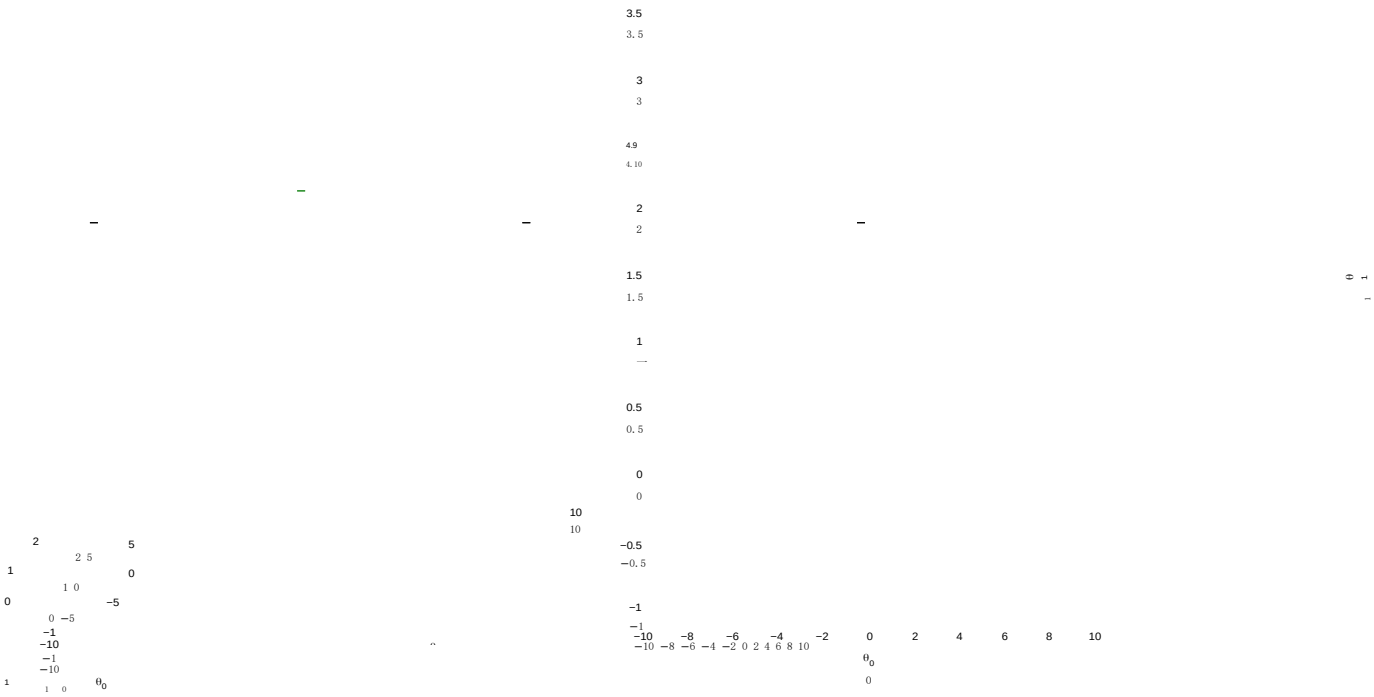
```
% initialize J vals to a matrix of 0's
J vals = zeros(length(theta0 vals), length(theta1 vals));

% Fill out J vals
for i = 1:length(theta0 vals)
    for j = 1:length(theta1
    vals)
        t = [theta0 vals(i); theta1
        vals(j)]; J vals(i,j) =
        computeCost(x, y, t);
```

```
% initialize J vals to a matrix of 0's
J vals = zeros(length(theta0 vals), length(theta1 vals));

% Fill out J vals
for i = 1:length(theta0 vals) for j
    = 1:length(theta1 vals)
        t = [theta0 vals(i); theta1 vals(j)]; J
        vals(i,j) = computeCost(x, y, t);
    end
end
```

After these lines are executed, you will have a 2-D array of J(θ) values. The script ex1.m will then use these values to produce surface and contour plots of J(θ) using the surf and contour commands. The plots should look something like Figure 3:

执行完这些行之后，您将得到一个 J(θ)值的二维数组。然后，脚本 ex1.m 将使用 surf 和 contour 命令使用这些值生成 J(θ)的表面和等值线图。这些情节应该看起来像图 3:





(a) Surface (b) Contour, showing minimum

(a)表面(b)轮廓，显示最小值

Figure 3: Cost function J(θ)

图 3:成本函数 J(θ)

The purpose of these graphs is to show you that how J(θ) varies with changes in θ_0 and θ_1. The cost function J(θ) is bowl-shaped and has a global mininum. (This is easier to see in the contour plot than in the 3D surface plot). This minimum is the optimal point for θ_0 and θ_1, and each step of gradient descent moves closer to this point.

这些图表的目的是向您展示 J(θ)如何随着 θ0 和 θ1 的变化而变化。成本函数 J(θ)是碗形的，并且具有全局最小值。(这在等高线图中比在 3D 表面图中更容易看到)。这个最小值就是 θ0 和 θ1 的最优点，梯度下降的每一步都向这个点靠拢。

Optional Exercises

可选练习

If you have successfully completed the material above, congratulations! You now understand linear regression and should able to start using it on your own datasets.

如果你已经成功完成了上面的材料，恭喜你！您现在已经理解了线性回归，应该能够开始在自己的数据集上使用它了。

For the rest of this programming exercise, we have included the following optional exercises. These exercises will help you gain a deeper understanding of the material, and if you are able to do so, we encourage you to complete them as well.

对于本编程练习的其余部分，我们包括了以下可选练习。这些练习将帮助你对材料有更深的理解，如果你能做到，我们鼓励你也完成它们。

---

5 Linear regression with multiple variables

6 多元线性回归

In this part, you will implement linear regression with multiple variables to predict the prices of houses. Suppose you are selling your house and you want to know what a good market price would be. One way to do this is to first collect information on recent houses sold and make a model of housing prices.

在这一部分中，您将实现多个变量的线性回归来预测房价。假设你正在出售你的房子，你想知道一个好的市场价格是多少。一种方法是首先收集最近出售的房屋信息，并制作一个房价模型。

The file ex1data2.txt contains a training set of housing prices in Portland, Oregon. The first column is the size of the house (in square feet), the second column is the number of bedrooms, and the third column is the price of the house.

文件 ex1data2. txt 包含俄勒冈州 Port- land 房价的训练集。第一栏是

房子的大小(平方英尺)，第二栏是卧室的数量，第三栏是房子的价格。

The ex1_multi.m script has been set up to help you step through this exercise.

ex1_multi.m脚本可帮助您逐步完成本练习。

6.1 Feature Normalization
6.2 特征标准化

The ex1_multi.m script will start by loading and displaying some values from this dataset. By looking at the values, note that house sizes are about 1000 times the number of bedrooms. When features differ by orders of magnitude, first performing feature scaling can make gradient descent converge much more quickly.

ex1_multi.m脚本将从加载和显示该数据集中的一些值开始。通过查看这些值，注意到房子的大小大约是卧室数量的1000倍。当特征按大小顺序不同时，首先执行特征缩放可以使梯度下降收敛得更快。

Your task here is to complete the code in featureNormalize.m to
您在这里的任务是完成featureNormalize.m中的代码

- Subtract the mean value of each feature from the dataset.
- 从数据集中减去每个要素的平均值。
- After subtracting the mean, additionally scale (divide) the feature values by their respective “standard deviations.”
- 减去平均值后，将特征值除以各自的“标准偏差”

The standard deviation is a way of measuring how much variation there is in the range of values of a particular feature (most data points will lie within 2 standard deviations of the mean); this is an alternative to taking the range of values (max-min). In Octave/MATLAB, you can use the "std" function to compute the standard deviation. For example, inside featureNormalize.m, the quantity `X(:,1)` contains all the values of x_1 (house sizes) in the training set, so `std(X(:,1))` computes the standard deviation of the house sizes. At the time that featureNormalize.m is called, the extra column of 1's corresponding to $x_0 = 1$ has not yet been added to X (see ex1 multi.m for details).

标准差是一种测量特定特性的值的范围内有多少变化的方法(大多数数据点将位于平均值的 2 个标准差内)；这是取值范围(最大-最小)的替代方法。在 Octave/MATLAB 中，可以使用"std"函数来计算标准差。例如，在 featureNormalize.m 中，数量 X(:, 1)包含训练集中 x1(房屋大小)的所有值，因此 std(X(:, 1))计算房屋大小的标准偏差。在 featureNormalize.m 被调用时，对应于 x0 = 1 的 1 的额外列还没有被添加到 X 中(参见 ex1 multi.m 的详情)。

You will do this for all the features and your code should work with datasets of all sizes (any number of features / examples). Note that each column of the matrix X corresponds to one feature.

您将对所有要素执行此操作，并且您的代码应适用于所有大小的数据集(任意数量的要素/示例)。注意，矩阵 X 的每一列对应于一个特征。

You should now submit your solutions.

您现在应该提交您的解决方案。

Implementation Note: When normalizing the features, it is important to store the values used for normalization - the *mean value* and the *standard deviation* used for the computations. After learning the parameters from the model, we often want to predict the prices of houses we have not seen before. Given a new $\mathbf{x}$ value (living room area and number of bed rooms), we must first normalize $\mathbf{x}$ using the mean and standard deviation that we had previously computed from the training set.

Implementation Note: When normalizing the features, it is important to store the values used for normalization - the *mean value* and the *standard deviation* used for the computations. After learning the parameters from the model, we often want to predict the prices of houses we have not seen before. Given a new $\mathbf{x}$ value (living room area and number of bed rooms), we must first normalize $\mathbf{x}$ using

6.3 Gradient Descent

6.4 梯度下降

Previously, you implemented gradient descent on a univariate regression problem. The only difference now is that there is one more feature in the matrix X. The hypothesis function and the batch gradient descent update

rule remain unchanged.

以前，您在单变量回归问题上实现了梯度下降。现在唯一不同的是矩阵 x 多了一个特征，假设函数和批量梯度下降更新规则不变。

You should complete the code in computeCostMulti.m and gradientDescentMulti.m to implement the cost function and gradient descent for linear regression with

您应该完成 computeCostMulti.m 和 gradientDescentMulti.m 中的代码，以实现线性回归的成本函数和梯度下降

multiple variables. If your code in the previous part (single variable) already supports multiple variables, you can use it here too.

多个变量。如果前一部分中的代码(单个变量)已经支持多个变量，您也可以在这里使用它。

Make sure your code supports any number of features and is well-vectorized. You can use 'size(X, 2)' to find out how many features are present in the dataset.

确保您的代码支持任意数量的特性，并且是良好矢量化的。您可以使用“size(X, 2)”来确定数据集中有多少个要素。

You should now submit your solutions.

您现在应该提交您的解决方案。

Implementation Note: In the multivariate case, the cost function can also be written in the following vectorized form:

$$J(\theta) = \frac{1}{2m}(X\theta - \vec{y})^T(X\theta - \vec{y})$$

where

$$X = \begin{bmatrix} -(x(1))^T- \\ -(x(2))^T- \\ \vdots \\ -(x(m))^T- \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \end{bmatrix}$$

The vectorized version is efficient when you're working with numerical computing tools like Octave/MATLAB. If you are an expert with matrix operations, you can prove

Implementation Note: In the multivariate case, the cost function can also be written in the following vectorized form:

$$J(\theta) = \frac{1}{2m}(X\theta - \vec{y})^T(X\theta - \vec{y})$$

where

$$X = \begin{bmatrix} -(x(1))^T- \\ -(x(2))^T- \\ \vdots \\ -(x(m))^T- \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(m) \end{bmatrix}$$

The vectorized version is efficient when you're working with numerical computing tools like Octave/MATLAB. If you are an expert with matrix operations

6.4.1 Optional (ungraded) exercise: Selecting learning rates

6.4.2 可选(未评分)练习:选择学习率

In this part of the exercise, you will get to try out different learning rates for the dataset and find a learning rate that converges quickly. You can change the learning rate by modifying ex1_multi.m and changing the part of the code that sets the learning rate.

在练习的这一部分，您将尝试数据集的不同学习率，并找到一个快速收敛的学习率。您可以通过修改 ex1 multi.m 并更改设置学习率的代码部分来更改学习率。

The next phase in ex1_ multi.m will call your gradientDescent.m function and run gradient descent for about 50 iterations at the chosen learning rate. The function should also return the history of J(θ) values in a vector

ex1 multi.m 的下一阶段将调用 gradientDescent.m 函数，并以选定的学习速率运行大约 50 次梯度下降迭代。该函数还应该返回向量中 J(θ)值的历史记录

J. After the last iteration, the ex1 multi.m script plots the J values against the number of the iterations.

J. 在最后一次迭代之后，ex1 multi.m 脚本根据迭代次数绘制 J 值。

If you picked a learning rate within a good range, your plot look similar Figure 4. If your graph looks very different, especially if your value of J(θ) increases or even blows up, adjust your learning rate and try again. We recommend trying values of the learning rate α on a log-scale, at multiplicative steps of about 3 times the previous value (i.e., 0.3, 0.1, 0.03, 0.01 and so on). You may also want to adjust the number of iterations you are running if that will help you see the overall trend in the curve.

如果你在一个很好的范围内选择了一个学习率，你的图看起来会很相似 4. 如果你的图表看起来非常不同，特别是如果你的 J(θ)值增加甚至爆炸，调整你的学习速度，再试一次。我们建议在对数标度上尝试学习率 α 的值，乘法步长约为先前值的 3 倍(即 0.3、0.1、0.03、0.01 等)。如果有助于看到曲线的整体趋势，您可能还想调整正在运行的迭代次数。

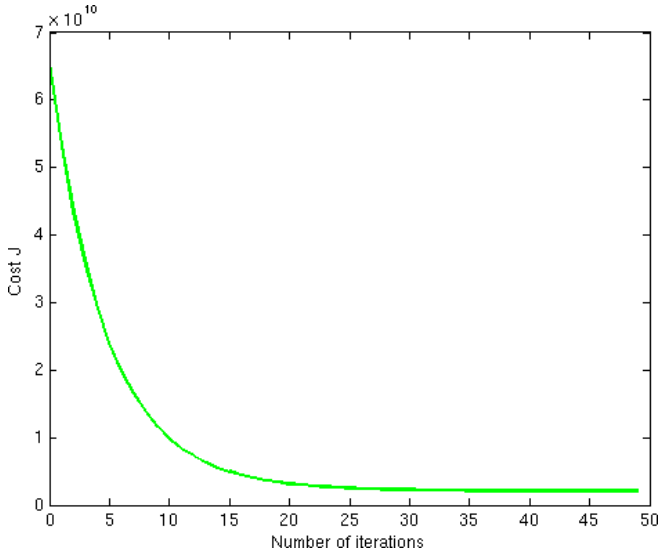


Figure 4: Convergence of gradient descent with an appropriate learning rate

图 4:具有适当学习速率的梯度下降收敛

Implementation Note: If your learning rate is too large, J(θ) can di- verge and 'blow up', resulting in valu

Octave/MATLAB Tip: To compare how different learning learning rates affect convergence, it's helpful to plot J for several learning rates on the same figure. In Octave

```
plot(1:50, J1(1:50), 'b');
hold on;
plot(1:50, J2(1:50), 'r');
plot(1:50, J3(1:50), 'k');
```


The final arguments 'b', 'r', and 'k' specify different colors for the plots.

Implementation Note: If your learning rate is too large, $J(\theta)$ can di- verge and 'bl

Octave/MATLAB Tip: To compare how different learning learning rates affect convergence, it's helpful to plot J for several learning rates on th

```
plot(1:50, J1(1:50), 'b');
hold on;
plot(1:50, J2(1:50), 'r');
plot(1:50, J3(1:50), 'k');
```

The final arguments 'b', 'r', and 'k' specify different colors for the plots.

Notice the changes in the convergence curves as the learning rate changes. With a small learning rate, you should find that gradient descent takes a very long time to converge to the optimal value. Conversely, with a large learning rate, gradient descent might not converge or might even diverge!

注意当学习率改变时收敛曲线的变化。在学习率很小的情况下，你应该会发现梯度下降需要非常长的时间才能收敛到最优值。相反，在学习率很大的情况下，梯度下降可能不收敛，甚至可能发散！

Using the best learning rate that you found, run the ex1 multi.m script to run gradient descent until convergence to find the final values of θ. Next, use this value of θ to predict the price of a house with 1650 square feet and 3 bedrooms. You will use value later to check your implementation of the normal equations. Don't forget to normalize your features when you make this prediction!

使用您找到的最佳学习率，运行 ex1 multi.m 脚本运行梯度下降，直到收敛，以找到 θ 的最终值。接下来，用 θ 的这个值来预测一栋 1650 平方英尺、3 间卧室的房子的价格。稍后您将使用 value 来检查法线方程的实现。做这个预测的时候别忘了把你的特征正常化！

You do not need to submit any solutions for these optional (ungraded) exercises.

对于这些可选的(未评分的)练习，您不需要提交任何答案。

6.5 Normal Equations
6.6 正规方程

In the lecture videos, you learned that the closed-form solution to linear regression is

在讲座视频中，您了解到线性回归的封闭解是

$$\theta = \mathbf{X}^T\mathbf{X}^{-1}\mathbf{X}^T\rightarrow y.$$

θ = XT X 1 XT→y。

Using this formula does not require any feature scaling, and you will get an exact solution in one calculation: there is no "loop until convergence" like in gradient descent.

使用这个公式不需要任何特征缩放，一次计算就能得到精确解：不存在像梯度下降中那样的"循环直到收敛"。

Complete the code in normalEqn.m to use the formula above to calculate θ. Remember that while you don't need to scale your features, we still need to add a column of 1's to the X matrix to have an intercept term (θ_0). The code in ex1.m will add the column of 1's to X for you.

完成 normalEqn.m 中的代码，使用上面的公式计算 θ 。请记住，虽然您不需要缩放要素，但我们仍然需要向 X 矩阵添加一列 1，以获得截距项（θ 0）。ex1.m 中的代码会将 1 的列添加到 X 中。

You should now submit your solutions.

您现在应该提交您的解决方案。

Optional (ungraded) exercise: Now, once you have found θ using this method, use it to make a price prediction for a 1650-square-foot house with 3 bedrooms. You should find that gives the same predicted price as the value you obtained using the model fit with gradient descent (in Section 3.2.1).

*可选(未评分)练习:现在，一旦你用这种方法找到 θ，就用它来预测一栋 1650 平方英尺、有 3 间卧室的房子的价格。您应该会发现，给出的预测价格与使用梯度下降模型拟合获得的价格相同(在第节中)*3.2.1).

Submission and Grading

提交和分级

After completing various parts of the assignment, be sure to use the submit function system to submit your solutions to our servers. The following is a breakdown of how each part of this exercise is scored.

完成作业的各个部分后，请务必使用提交功能系统将您的解决方案提交到我们的服务器。下面是这个练习的每一部分是如何评分的。

Part	Submitted File	Points
Warm up exercise	warmUpExercise.m	10 points
Compute cost for one variable	computeCost.m	40 points
Gradient descent for one variable	gradientDescent.m	50 points
Total Points		100 points
部分	提交的文件	点
热身运动	warmUpExercise.m	10 分
计算一个变量的成本	computeCost.m	40 分
单变量梯度下降	gradientDescent.m	50 分
总分		100 分

Optional Exercises

可选练习

Part	Submitted File	Points
Feature normalization	featureNormalize.m	0 points
Compute cost for multiple variables	computeCostMulti.m	0 points
Gradient descent for multiple variables	gradientDescentMulti.m	0 points
Normal Equations	normalEqn.m	0 points
部分	提交的文件	点

特征标准化 计算多个变量的成本 多变量梯度下降	featureNormalize.m 计算机科学多媒体	0 分 0 分 0 分
正规方程	正常设备	0 分

You are allowed to submit your solutions multiple times, and we will take only the highest score into consideration.

您可以多次提交您的解决方案，我们将只考虑最高分。