编程练习 1：线性回归

机器学习

# 介绍

在本练习中，您将实现线性回归，并了解它在数据上的工作原理。在开始此编程练习之前，我们强烈建议您观看视频 讲座并完成相关主题的复习题目。

要开始练习，您需要下载入门代码并将其内容解压缩到要完成练习的目录中。如果需要，在开始本练习之前，请使用 cd comm和 Octave/MATLAB 更改为此目录。

您还可以在课程网站的“环境设置说明”中找到有关安装Octave / MATLAB的说明。

## 本练习中包含的文件

ex1.m - Octave/MATLAB 脚本，用于引导您完成练习 ex1 multi.m - 练习后面部分的 Octave/MATLAB 脚本 ex1data1.txt - 具有一个变量 ex1data2 的线性回归数据集.txt - 用于使用多个变量的线性回归的数据集 submit.m - 将您的解决方案发送到我们服务器的提交脚本 [*？*] warmUpExercise.m - Octave/MATLAB 中的 Simple example function [*？*] plotData.m - 显示数据集的函数

[*?*] computeCost.m - 计算线性回归成本的函数

[*?*] 梯度下降.m - 运行梯度下降的函数

[†] computeCostMulti.m - 多个变量的成本函数

[†] gradientDescentMulti.m - 多个变量的梯度下降

[†] featureNormalize.m - 用于规范化特征的函数

[†] normalEqn.m - 计算正态方程的函数

*?* 表示您需要完成的文件

† 表示 可选练习

在整个练习过程中，您将使用 脚本 ex1.m 和 ex1 multi.m。这些脚本为概率 ms 设置数据集，并调用要编写的函数。您无需修改其中任何一个。您只需按照此赋值中的说明修改其他文件中的函数。

对于此编程练习，您只需要完成练习的第一部分即可使用一个变量实现线性回归。练习的第二部分是可选的，涵盖具有多个变量的线性回归。

## 从何处获取帮助

本课程中的练习使用Octave[[1]](#footnote-1) 或MATLAB，这是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，请参阅课程网站“环境设置说明”中的安装说明。

在 Octave/MATLAB 命令行中，键入 help 后跟函数名称将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。有关倍频程函数的更多文档，请参阅[倍频程文档页面。](http://www.gnu.org/software/octave/doc/interpreter/) MATLAB 文档可以在 [MATLAB 文档说明页面上](http://www.mathworks.com/help/matlab/?refresh=true)找到[。](http://www.mathworks.com/help/matlab/?refresh=true)

我们还强烈建议使用在线 **讨论** 与其他学生讨论练习。但是，不要查看他人编写的任何源代码或与他人共享您的源代码。

# 简单的倍频程/MATLAB函数

ex1.m 的第一部分为您提供了 Octave/MATLAB 语法和家庭作业提交过程的练习。在文件warmUpExercise.m中，您将找到Octave / MATLAB函数的大纲。通过填写以下代码对其进行修改以返回 5 x 5 单位矩阵：

A = 眼（5）;

完成后，运行 ex1.m（假设您位于正确的目录中，在 Octave/MATLAB 提示符下键入“ex1”），您应该会看到类似于以下内容的输出：

年 =

诊断矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |

现在 ex1.m 将暂停，直到您按任意键，然后运行分配的下一部分的代码。如果要退出，键入 ctrl-c 将在程序运行过程中停止程序。

## 提交解决方案

完成部分练习后，您可以通过在 Octave/MATLAB 命令行中键入 submit 来订阅解决方案以进行评分。提交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 Web p 期限获取提交令牌。

*您现在应该提交您的解决方案。*

您可以多次提交解决方案，我们将仅考虑最高分。

# 具有一个变量的线性回归

在本练习的这一部分中，您将 使用一个变量实现线性回归，以预测食品运输车的利润。假设您是一家餐厅特许经营店的首席执行官，并且正在考虑在不同的城市开设新 店。该连锁店已经在各个城市拥有卡车，并且您拥有 来自城市的利润和人口的数据。

您希望使用此数据来帮助您选择要扩展到的下一个城市。

文件 ex1data1.txt 包含线性回归问题的数据集。第一列是城市的人口 ，第二列是该市食品卡车的利润。利润的负值表示亏损。

ex1.m 脚本已设置为为您加载此数据。

## 绘制数据

在开始执行任何任务之前，通过可视化数据来取消数据状态通常很有用。对于此数据集，您可以使用散点图来可视化数据，因为它只有两个要绘制的属性（利润和总体）。（你在现实生活中会遇到的很多其他问题都是多维的和c的不要在2D图上绘制。

在 ex1.m 中，数据集从数据文件加载到变量 *X* 和 *y 中*：

|  |  |
| --- | --- |
| data = load（'ex1data1.txt'）; | % 读取 逗号分隔的数据 |
| X = data（：， 1）; y = data（：， 2）; m = length（y）; | 训练示例的百分比 |

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

|  |  |
| --- | --- |
| plot（x， y， 'rx'， 'MarkerSize'， 10）; | % 绘制 数据 |
| ylabel（“10，000美元的利润”）; | % 设置 y−轴标签 |
| xlabel（“10，000s的城市人口”）; | % 设置 x−轴标签 |

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

要了解有关绘图命令的更多信息，可以在 Octave/MATLAB 命令提示符下键入帮助绘图，或在线搜索绘图文档。（要将标记更改为红色“x”，我们使用了选项'rx' 与 plot 命令一起，即 plot（..，[您的选择在这里],..,

'rx'）; )

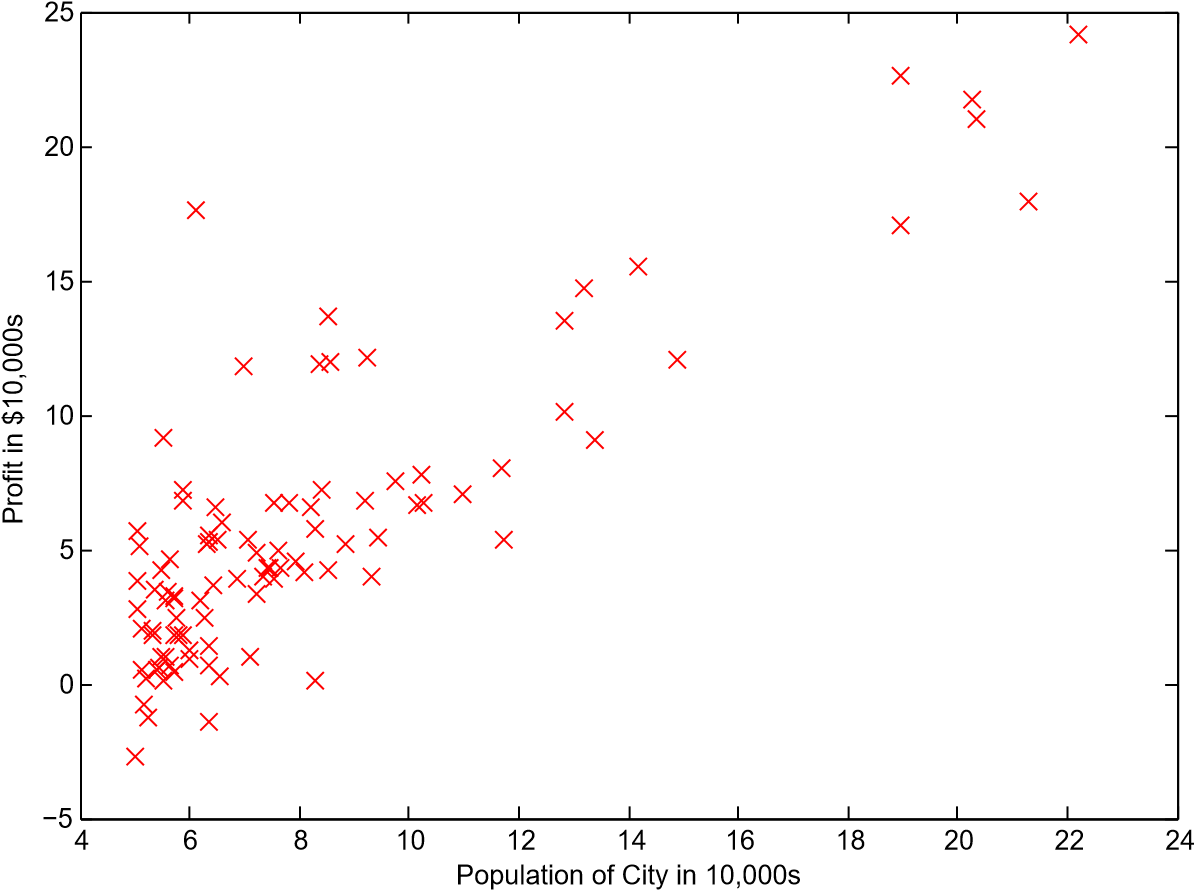


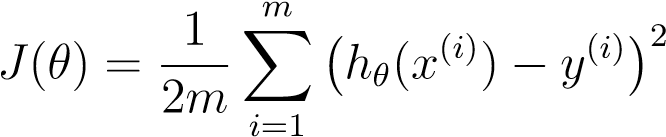
图 1：训练数据的散点图

## 梯度下降

在这一部分中，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中。

### 更新方程式

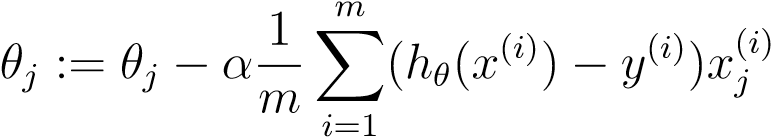
线性回归的目标是 最小化成本函数



其中假设 *hθ*（*x*） 由线性模型给出

*hθ*（*x*） = *θTx* = *θ*0 + *θ*1*x*1

回想一下，模型的参数是 *θj* 值。这些是您将调整的值，以最大限度地降低成本 *J*（*θ*）。执行此操作的一种方法是使用批处理梯度下降算法。在批处理梯度下降中，每次迭代都会执行更新

 （同时更新所有 *j* 的 *θ* *j*）*.*

随着梯度下降的每一步，您的参数 *θj* 更接近最优值，从而实现最低成本的 *J*（*θ*）。

**实现说明：**我们将每个示例存储为 Octave/MATLAB中X矩阵中的一行。为了考虑截距 term （*θ*0），我们向 X 添加额外的第一列 ，并将其设置为所有列。这允许我们将*θ*0简单地视为另一个“特征”。

### 实现

在 ex1.m 中，我们已经为线性回归设置了数据。在下面的行中，我们向数据添加另一个维度以适应 *θ*0 截距项。我们还将初始参数初始化为 0，将学习速率 alpha 初始化为 0.01。

|  |
| --- |
| X = [ones（m， 1）， data（：，1）]; % 将一列 1 添加到 x theta = 零（2， 1）; % 初始化拟合参数  迭代次数 = 1500;阿尔法 = 0.01; |

### 计算成本 *J*（*θ*）

当您执行梯度下降以学习最小化成本函数 *J*（*θ*）时，通过计算成本来监控收敛性是有帮助的。在本节中，您将实现一个函数来计算 *J*（*θ*），以便您可以检查梯度下降实现的收敛性。

您的下一个任务是完成文件 computeCost.m 中的代码，该文件是计算 *J*（*θ*） 的函数。执行此操作时，请记住，变量 *X* 和 *y* 不是标量值，而是矩阵，其行表示训练集中的示例。

完成该函数后，ex1.m 中的下一个s tep 将使用*初始化*为零的 θ 运行 computeCost 一次，您将看到打印到屏幕上的成本。

您应该期望看到 32.07 的成本。

*您现在应该提交您的解决方案。*

### 梯度下降

接下来，您将在文件 gradientDescent.m 中实现渐变下降。循环结构已经为您编写，您只需要在每次迭代中提供对*θ*的更新。

在编程时，请确保您了解要优化的内容以及要更新的内容。保持在最小d 中，成本 *J*（*θ*） 由向量 θ 参数化，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，而不是通过改变*X*或*y*。如果您不确定，请参阅本讲义中的方程式和视频讲座。

验证梯度下降是否正确工作的一个好方法是查看*J*（*θ*）的值，并检查它是否随着每一步而减小。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并计算了Cost，则*J*（*θ*）的值不应增加，并且应该在算法结束时收敛到稳定值。

完成后， ex1.m 将使用最终参数来绘制线性拟合。结果应如图 2 所示：

*θ*  的最终值还将用于预测 35，000 人和 70，000 人的区域内的利润。请注意 ex1.m 中的以下行使用矩阵乘法（而不是显式求和或循环）来计算预测的方式。这是Octave/MATLAB中代码矢量化的一个示例。

*您现在应该提交您的解决方案。*

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

## 调试

在实现梯度下降时，请记住以下几点： • Octave/MATLAB 数组索引从 o ne 开始，而不是从零开始。如果要将 *θ*0 和 *θ*1  存储在称为 θ 的向量中，则值将为 θ（1） 和 θ（2）。。

* 如果在运行时看到许多错误，请检查矩阵运算，以确保对兼容的 di mensions 的矩阵进行相加和乘法。使用 size 命令打印变量的尺寸 将帮助您进行调试。

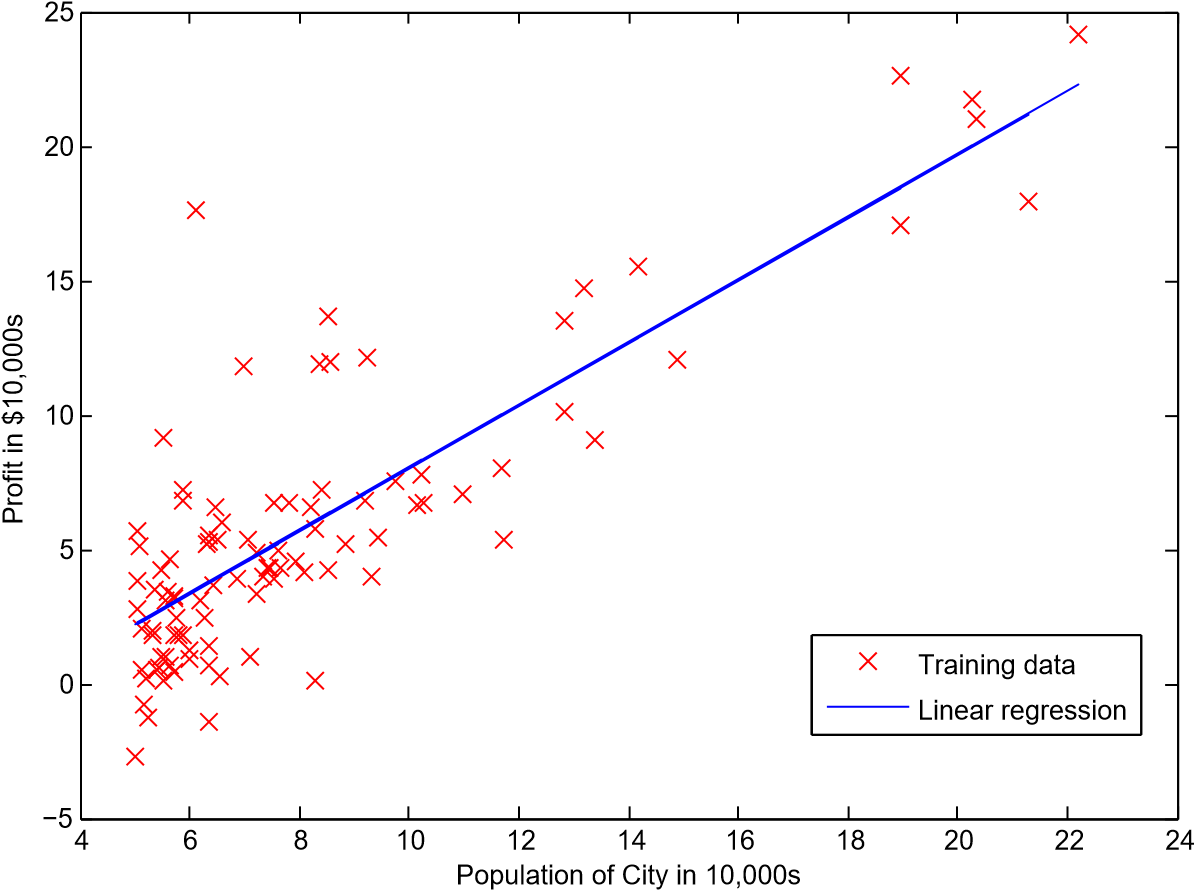


图 2：具有线性回归拟合的训练数据

* 默认情况下，Octave/MATLAB 将数学运算符解释为矩阵运算符。这是大小不兼容错误的常见来源。如果您不需要矩阵乘法，则需要添加“点”表示法以将其指定为Octave / MATLAB。例如， A\*B 执行矩阵乘法，而 A.\*B 执行按元素乘法。

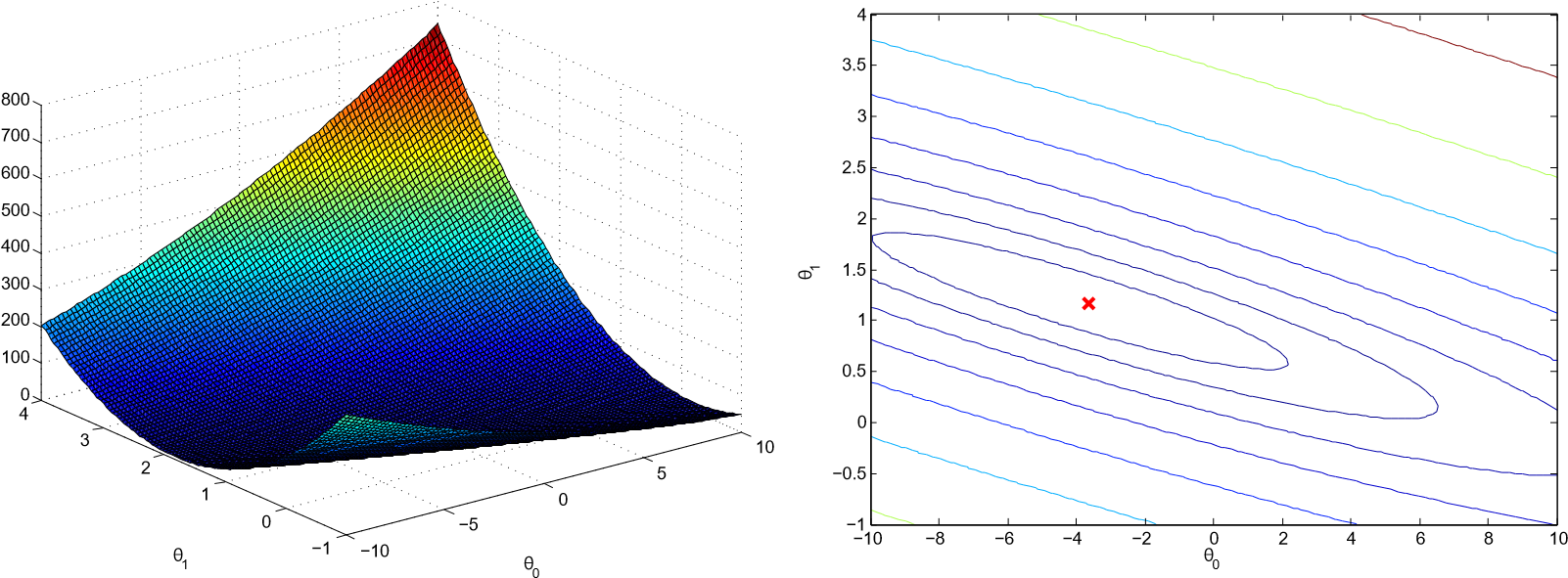
## 可视化 *J*（*θ*）

为了更好地理解成本函数 *J*（*θ*），您现在将在 θ 0 和 *θ*1  值的二维网格上绘制成本。你不需要为th是一部分编写任何新代码，但你应该了解你已经编写的代码是如何创建这些图像的。

在 ex1.m 的下一步中，设置了代码以使用您编写的 computeCost 函数在值网格上计算 *J*（*θ*）。

|  |
| --- |
| % 将 J 值初始化为 0 的矩阵  J vals = zeros（length（theta0vals）， length（theta1vals））;  % Fill out J vals for i = 1：length（theta0vals） for j = 1：length（theta1vals） t = [theta0vals（i）; theta1vals（j）];  J vals（i，j） = computeCost（x， y， t）; 结束  结束 |

执行这些行后，您将拥有一个 *J*（*θ*） 值的二维数组。然后，脚本 ex1.m 将使用这些值通过 surf 和 contour 命令生成 *J*（*θ*） 的表面和等值线图。这些图应如图 3 所示：



（a） Surface （b） 轮廓，显示最小值

图 3：成本函数 *J*（*θ*）

这些图的目的是向您展示 *J*（*θ*） 如何随 θ 0 和 *θ*1 的变化而变化。成本函数 *J*（*θ*） 是碗形的，具有全局最小值。（这在等值线图中比在 3D 曲面图中更容易看到）。此最小值是 *θ*0 和 *θ*1 的最佳点，梯度下降的每个步骤都靠近此点。

# 可选练习

如果您已成功完成 上述材料，恭喜您！您现在了解了线性回归，并且应该能够开始在自己的数据集上使用它。

对于本编程练习的其余部分，我们包括以下可选练习。这些练习将帮助您 更深入地了解材料，如果您能够这样做，我们鼓励您也完成它们。

# 具有多个变量的线性回归

在这一部分中，您将使用多个变量实现线性回归来预测 房屋价格。假设您正在出售房屋，并且想知道一个好的市场价格。一种方法是首先收集有关最近售出房屋的信息，并制作房价模型。

文件 ex1data2.txt 包含俄勒冈州波特兰市房价的一组训练。第一列是房屋的大小（以平方英尺为单位），第二列是卧室的数量，第三列是房屋的价格。

ex1 multi.m 脚本的设置是为了帮助您逐步完成练习。

## 功能规范化

ex1 multi.m 脚本将首先加载并显示此数据集中的一些值。通过查看这些值，请注意房屋大小约为卧室数量的1000倍。当特征因 magnitude 阶数而异时，首先执行特征缩放可以使梯度下降收敛得更快。

您在这里的任务是完成 featureNormalize.m 中的代码

* 从数据集中减去每个要素的平均值。
* 减去平均值后， 另外按各自的“标准差”缩放（除以）特征值。

标准差是一种测量特定特征值范围变化量的方法（大多数数据点将位于平均值的±2个标准差范围内）;这是取值范围（max-min）的替代方法。在Octave/MATLAB中，您可以使用“std”函数来计算标准偏差。例如，在 featureNormalize.m 内部，数量 X（：，1） 包含训练集中 *x*1（房屋大小）的所有值，因此 std（X（：，1）） 计算房屋大小的标准差。在调用 featureNormalize.m 时，对应于 *x*0  = 1 的额外 1 列尚未添加到 X 中（有关 details，请参见 ex1 multi.m）。

您将为所有功能执行此操作，并且您的代码应适用于各种大小的数据集（任意数量的功能/示例）。请注意，矩阵 X 的每一列对应于一个特征。

*您现在应该提交您的解决方案。*

**实现说明：**在对要素进行归一化时，存储用于归一化的值非常重要 - 用于计算的*平均值*和*标准差*。从模型中学习参数后，我们经常想要预测我们以前从未见过的房屋价格。给定一个新的 **x** 值（客厅面积和卧室数量），我们必须首先使用之前从训练集中计算的平均值和标准差来规范化 **x**。

## 梯度下降

以前，you 在单变量回归问题上实现了梯度下降。现在唯一的区别是矩阵X中还有一个特征。假设函数和批次梯度下降更新规则保持不变。

您应该在 computeCostMulti.m 和 gradientDescentMulti.m 中完成代码，以实现具有多个变量的线性回归的成本函数和梯度下降。如果你在上一部分（单个变量）中的代码已经支持多个 v阿里可对象，你也可以在这里使用它。

确保您的代码支持任意数量的功能，并且具有良好的矢量化。您可以使用“size（X， 2）”来了解数据集中存在多少个要素。

*您现在应该提交您的解决方案。*

|  |
| --- |
| **实现说明：** 在 多变量情况下，成本函数也可以以以下矢量化形式编写：    哪里   （1））*T* —   *和*（1）   — （*x*   — （*x*（2））*T* —   *X* =  .. *。*      — (  当您使用 Octave / MATLAB等数值计算工具时，矢量化版本非常有效。如果您是矩阵运算方面的专家，则可以向自己证明这两种形式是等效的。 |

### 可选（未评分）练习：选择学习率

在本部分练习中，您将尝试数据集的差异学习速率，并找到快速收敛的学习速率。您可以通过修改 ex1 multi.m 并更改设置学习速率的代码部分来更改学习速率。

ex1 multi.m 中的下一阶段将调用您的 gradientDescent.m 函数，并以所选的学习速率运行梯度下降约 50 次迭代。该函数还应返回向量 *J 中 J*（*θ*） 值的历史记录。在最后一次迭代之后，ex1 multi.m 脚本将 J values 与迭代次数对应起来。

如果您选择的学习率在一个很好的范围内，您的绘图 将类似于 图 4。如果你的图看起来非常不同，特别是如果你的 *J*（*θ*）值增加甚至爆炸，请再次调整你的学习速率和try。我们建议尝试 *以* 大约 3 倍于前一个值（即 0.3、0.1、0.03、0.01 等）的乘法步长，在对数尺度上α学习速率值。您可能还想调整正在运行的迭代次数，如果这有助于您查看曲线中的整体趋势。

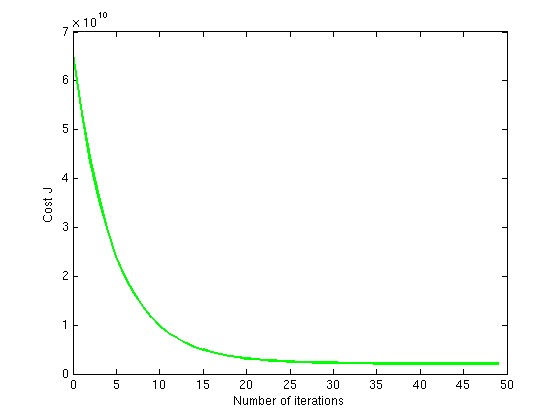


图4：梯度下降与适当学习速率的收敛性

**实现注意事项：**如果您的 学习速率太大，*J*（*θ*）可能会发散并“爆炸”，从而导致值太大而无法进行计算机计算。在这些情况下，Octave/MATLAB 将倾向于返回 NaNs。 NaN代表“不是一个数字”，通常是由涉及-∞和+∞的未定义运算引起的。

|  |
| --- |
| **Octave/MATLAB 提示：**要比较不同的学习 学习速率如何影响收敛，在同一 图上绘制多个学习速率的 J 是很有帮助的。在倍频程/MATLAB中，这可以通过在图之间“保持”c ommand多次执行梯度下降来完成。具体来说，如果您尝试了三个不同的 alpha 值（您可能应该尝试比这更多的值），并将成本存储在 J1、J2 和 J3 中，则可以使用以下命令将它们绘制在同一图上：  情节（1：50， J1（1：50）， 'b'）; 坚持住;  情节（1：50， J2（1：50）， 'r'）; 情节（1：50， J3（1：50）， 'k'）;  最后的参数 “b”、“r” 和 “k” 为绘图指定不同的颜色。 |

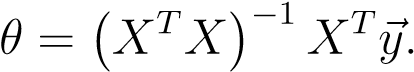
请注意，随着学习速率的变化，收敛曲线的变化。对于较小的学习 速率，您应该发现梯度下降需要很长时间才能收敛到最佳值。相反，在学习率大的情况下，梯度下降可能不会收敛，甚至可能发散！

使用找到的最佳学习速率，运行 ex1 multi.m 脚本以运行梯度下降，直到收敛以找到 θ 的最终值。接下来，使用此值 *θ* 来预测具有 1650 平方英尺和 3 间卧室的房屋的价格。稍后将使用 value 来检查正常 equations 的实现。当您进行此预测时，不要忘记规范化您的特征！

*您无需为这些可选（未评分）练习提交任何解决方案。*

## 正态方程

在讲座视频中，您了解到线性规则的闭式解是



使用此公式不需要任何特征缩放，您将在一次计算中获得精确的解决方案：没有像梯度下降那样的“收敛循环”。

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。

*您应该提交您的解决方案。*

*可选（未分级）练习：*现在，一旦您使用此方法找到*θ*，就可以使用它来预测带有3间卧室的1650平方英尺房屋的价格。您应该发现，给出的预测价格与您使用梯度下降拟合模型（在第 3.2.1 节中）计算的值相同。

# 提交和评分

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。以下是 本练习每个部分的评分方式的细分。

|  |  |  |
| --- | --- | --- |
| **部分** | **提交的文件** | **点** |
| 热身运动 | warmUpExercise.m | 10 积分 |
| 一个变量的计算成本 | computeCost.m | 40 积分 |
| 一个变量的梯度下降 | gradientDescent.m | 50 积分 |
| 总 积分 |  | 100 积分 |

## 可选练习

|  |  |  |
| --- | --- | --- |
| **部分** | **提交的文件** | **点** |
| 特征规范化 | featureNormalize.m | 0 积分 |
| 多个 计算 成本  变量 | computeCostMulti.m | 0 积分 |
| 多个变量的梯度下降 | gradientDescentMulti.m | 0 积分 |
| 正态方程 | normalEqn.m | 0 积分 |

您可以多次提交解决方案，我们将仅考虑最高分。

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。 [↑](#footnote-ref-1)