编程练习 5：

正则化线性回归和 偏差方差

机器学习

# 介绍

在本练习中，您将实现正则化线性回归，并使用它来研究具有不同偏差-方差属性的模型。在开始编程练习之前，我们强烈建议您观看视频讲座并完成 相关主题的评论问题。

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Octave/MATLAB中的cd命令更改为他的目录。

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

ex5.m - Octave/MATLAB 脚本，可引导您完成 练习 ex5data1.mat - 数据集

submit.m - 将您的解决方案发送到我们的服务器的提交脚本 功能Normalize.m - 特征归一化函数 fmincg.m - 函数最小化例程（类似于 fminunc） plotFit.m - 绘制多项式拟合 训练LinearReg.m - 使用您的成本函数训练线性回归 [*？*] linearRegCostFunction.m - 正则化线性回归成本函数

[*?*] learningCurve.m - 生成学习曲线

[*?*] polyFeatures.m - 将数据映射到多项式特征空间

[*?*] validationCurve.m - 生成交叉验证曲线

*?* 表示您需要完成的文件

在整个练习过程中，您将使用 脚本 ex5.m。 这些脚本为问题设置数据集，并调用您将编写的函数。您只需按照此赋值中的说明修改其他文件中的函数。

## 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。

# 正则化线性回归

在练习的前半部分中，您将实现正则线性回归，以使用水库中水位的变化来预测流出大坝的水量。在下一个 h alf 中，您将对调试学习算法进行一些诊断，并检查偏差 与方差的影响。

提供的脚本 ex5.m 将帮助您逐步完成本练习。

## 可视化数据集

我们将首先可视化包含有关水位变化 *x* 和流出大坝的水量 *y* 的历史记录的数据集。

此数据集分为三个部分：

* 模型将学习的训练集：X， y
* 用于确定正则化参数的**交叉验证**集：

Xval， yval

* 用于评估性能的测试集。这些是您的模型在训练期间没有看到的“看不见”的示例：Xtest，ytest

ex5.m 的下一步将绘制训练数据（图 1）。在以下部分中，您将实现线性回归，并使用它来拟合数据直线并绘制学习曲线。接下来，您将实现多项式回归以找到更适合数据的方法。

−50

−40

−30

−20

−10

0

10

20

30

40

0

5

10

15

20

25

30

35

40

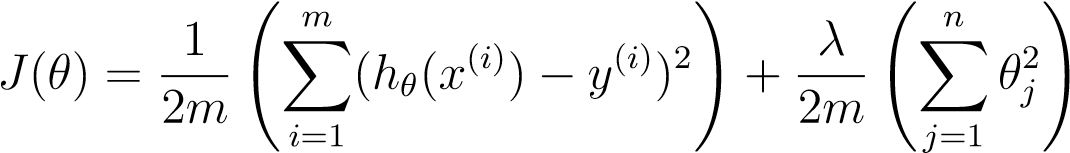
水位变化 （x）

流出大坝的水 （y）

图1：数据

## 正则化线性回归成本函数

回想一下，正则化线性回归具有以下成本函数：

 *,*

其中 *λ* 是控制正则化程度的正则化参数（因此有助于防止过度拟合）。正则化项对总体成本 *J* 进行了惩罚。随着模型参数 *θj* 的幅度增加，惩罚也会增加。请注意，不应正则化 *θ*0 项。（在 Octave/MATLAB 中， *θ*0 项表示为 theta（1），因为 Octave/MATLAB 中的索引从 1 开始）。

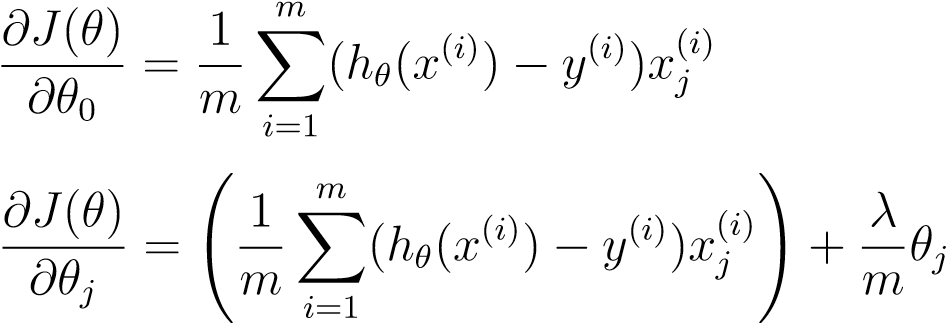
您现在应该完成文件 linearRegCostFunction.m 中的代码。您的任务是编写一个函数来计算正则化线性回归成本函数。如果可能，请尝试对代码进行矢量化，避免编写循环。完成后，ex5.m 的下一部分将使用初始化为 [1; 1] 的 theta 运行成本函数。您应该 expect 以查看

303.993.

*您现在应该提交您的解决方案。*

## 正则化线性回归梯度

相应地，正则化线性回归的*θj*成本的偏导数定义为

对于 *j* = 0

对于 *j* ≥ 1

在 linearRegCostFunction.m 中，添加代码以计算梯度，并在变量 grad 中返回梯度。完成后，ex5.m 的下一个 par t 将使用初始化为 [1; 1] 的 theta 运行梯度函数。您应该会看到渐变 [-15.30; 598.250]。

*您现在应该提交您的解决方案。*

## 拟合线性回归

一旦您的成本函数和梯度正确工作，ex5.m的下一部分将在trainLinearReg.m中运行代码 以计算θ的最优值。此训练函数使用 fmincg 来优化成本函数。

在这一部分中，我们将正则化参数 *λ* 设置为零。因为我们目前的线性回归实现试图拟合一个二维 *θ*，所以正则化对于 如此低维的θ不会有太大的帮助。在 exer cise 的后面部分中，您将使用具有正则化的多项式回归。

最后， ex5.m 脚本还应绘制最佳拟合线，从而生成类似于图 2 的图像。最佳拟合线告诉我们，模型不适合数据，因为 数据具有非线性模式。虽然如图所示可视化最佳拟合是调试学习算法的一种可能方法，但可视化数据和模型并不总是那么容易。在下一节中，您将实现一个函数来生成学习曲线，即使您不容易可视化数据，它也可以帮助您调试学习算法。

−50

−40

−30

−20

−10

0

10

20

30

40

−5

0

5

10

15

20

25

30

35

40

水位变化 （x）

流出大坝的水 （y）

图 2：线性拟合

# 偏置方差

机器学习中的一个重要概念是偏差-方差权衡。具有高偏差的模型对于数据来说不够复杂，并且往往欠拟合，而具有高方差的模型对训练数据过度拟合。

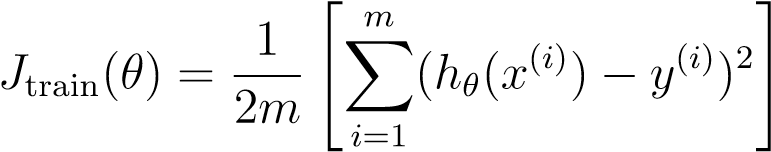
在本部分练习中，您将在学习曲线上对错误进行训练和测试，以诊断偏差方差问题。

## 学习曲线

现在，您将实现代码以生成学习曲线，这些曲线在调试学习算法时非常有用。回想一下，学习曲线将训练 和交叉验证误差绘制为训练集大小的函数。您的工作是填写 learningCurve.m ，以便它返回训练集和交叉验证集的错误向量。

为了绘制学习曲线，我们需要针对不同训练集大小的*训练*和交叉验证集误差。要获得不同的训练集大小，应使用原始训练集 X 的不同子集。具体来说，对于 i 的训练集大小，应使用第一个 i 示例（即 X（1：i，:) 和 y（1：i））。。

您可以使用 trainLinearReg 函数来查找 *θ* 参数。请注意， lambda 作为参数传递给 learningCurve 函数。学习 *θ* 参数后，应计算 训练集和交叉验证集上的误差。回想一下，数据集的训练误差定义为

 *.*

特别注意，训练错误不包括正则化项。计算训练误差的一种方法是使用现有的 cost 函数，并且  *仅在*  使用它来计算训练误差和交叉验证误差时才将 λ 设置为 0。计算训练集错误时，请确保在训练子集上计算它（即 X（1：n，:) 和 y（1：n））（而不是整个 training 集）。但是，对于交叉验证错误，应在整个交叉验证集上计算该错误 。应将计算出的错误存储在向量错误序列和错误值中。

完成后，ex5.m 将打印学习曲线并生成类似于图 3 的绘图。

*您现在应该提交您的解决方案。*

在图 3 中，您可以观察到， 当训练示例的数量增加时，训练误差和交叉验证误差都很高。这反映了模型中**的高偏差**问题 - 线性回归模型是

线性回归的学习曲线

0

2

4

6

8

10

12

0

50

100

150

训练示例数

错误

火车

交叉验证

图 3：线性回归学习曲线

太简单了，无法很好地拟合我们的数据集。在下一节中，您将实现多项式回归，以适合此数据集的更好模型。

# 多项式回归

我们的线性模型的问题在于它对于数据来说太简单了，导致欠拟合（高偏倚）。在本部分练习中，您将通过添加更多功能来解决此问题。

对于多项式回归，我们的假设具有 form：

*hθ*（*x*） = *θ*0 + *θ*1 ∗ （水水平） + *θ*2 ∗ （水水平）2 + ··· + *θp* ∗ （水水平）*p* = *θ*0 + *θ*1*x*1 + *θ*2*x*2 + *...* + *θpxp.*

请注意，通过定义 *x*1 = （waterLevel）*，x*2 = （waterLevel）2*,...,xp* = （waterLevel）*p*，我们得到一个线性回归模型，其中特征是原始值 （waterLevel） 的各种幂。

现在，您将使用数据集中现有要素 *x*  的更高倍数添加更多要素。本部分中的任务是完成 polyFeatures.m 中的代码 ，以便函数将大小为 *m*×1 的原始训练集 X 映射到其更高的幂。具体来说，当将大小为 *m*×1 的训练集 X 传递到函数中时，该函数应返回 *m*×*p* 矩阵 X poly，其中列 1 保存 X 的原始值，列 2 保存 X 的值。^2，列 3 保存 X.^3 的值。，依此类推。请注意，您不必考虑此函数中的零 eth 幂。

现在，您有一个将特征映射到更高维度的函数，ex5.m 的第 6 部分会将其应用于训练集、测试集和交叉验证集（您尚未使用）。

*您现在应该提交您的解决方案。*

## 学习多项式回归

完成 polyFeatures.m 后，ex5.m 脚本将继续使用线性回归成本函数训练多项式回归。

请记住，即使我们在特征向量中有多项式项，我们仍然在解决线性回归优化问题。多元项只是简单地变成了我们可以用于线性回归的特征。我们使用的成本函数和梯度与您在本练习的前面部分编写的成本函数和梯度相同。

在本部分练习中，您将使用 degr ee 8 的多项式。事实证明，如果我们直接在投影数据上运行训练，则不会很好地工作，因为特征将严重扩展（例如， *x* = 40的示例现在将具有特征 *x*8 = 408 = 6*。*5 × 1012）。因此，您将需要使用无 rmalization 功能。

在学习多项式回归的参数*θ*之前，ex5.m将首先调用特征规范化和归一化训练集的特征，分别存储mu，sigma参数。我们已经为 y ou 实现了这个函数，它与第一个练习中的函数相同。

学习参数 *θ* 后，您应该看到为 *λ* = 0 的多项式回归生成的两个图（图 4，5）。

从图 4 中，您应该看到多项式拟合能够很好地跟踪 datap软膏 - 从而获得较低的训练误差。然而，多项式拟合非常复杂，甚至在极端情况下下降。这是多项式回归模型过度拟合训练数据并且不能很好地泛化的迹象。

为了更好地理解未映射 （*λ* = 0） 模型的问题，您可以看到学习曲线（图 5）显示了相同的效果，其中 低训练误差较低，但 交叉验证 误差较高。训练与交叉验证错误之间存在差距，表明存在高方差问题。

−80

−60

−40

−20

0

20

40

60

80

−60

−50

−40

−30

−20

−10

0

10

20

30

40

水位变化 （x）

流出大坝的水 （y）

多项式回归拟合（lambda = 0.000000）

图 4：多项式拟合， *λ* = 0

多项式回归学习曲线（lambda = 0.000000）

0

2

4

6

8

10

12

0

10

20

30

40

50

60

70

80

90

100

训练示例数

错误

火车

交叉验证

图 5：多项式学习曲线， *λ* = 0

解决过拟合（高方差）问题的一种方法是向模型添加正则化。在下一节中，您将了解不同的 *λ* 参数，以了解正则化如何导致更好的模型。

## 可选（未评分）练习：调整正则化参数

在本节中，您将观察正则化参数如何影响正则化多项式回归的偏差方差。您现在应该在 ex5.m 中修改 lambda 参数，并尝试 *λ* = 1*，*100。对于这些值中的每一个，脚本都应生成数据的多项式拟合以及学习曲线。

对于 *λ* = 1，您应该看到一个很好地遵循数据趋势的多项式拟合（图 6）和一条学习曲线（图 7），显示 cross 验证和训练误差都收敛到一个相对较低的值。这表明 *λ* = 1 正则多项式回归模型没有 高比值 或高方差问题。实际上，它在偏差和方差之间实现了良好的权衡。

对于 *λ* = 100，您应该看到一个多项式拟合（图 8），它不能很好地跟随数据。在这种情况下，正则化太多，模型无法拟合训练数据。

*您无需为此可选（未评分）exercise* 提交任何解决方案。

−80

−60

−40

−20

0

20

40

60

80

0

20

40

60

80

100

120

140

160

水位变化 （x）

流出大坝的水 （y）

多项式回归拟合 （lambda = 1.000000）

图 6：多项式拟合， *λ* = 1

多项式回归学习曲线 （lambda = 1.000000）

0

2

4

6

8

10

12

0

10

20

30

40

50

60

70

80

90

100

训练示例数

错误

火车

交叉验证

图 7：多项式学习曲线， *λ* = 1

−80

−60

−40

−20

0

20

40

60

80

−10

−5

0

5

10

15

20

25

30

35

40

水位变化 （x）

流出大坝的水 （y）

多项式回归拟合（lambda = 100.000000）

图 8：多项式拟合， *λ* = 100

## 使用交叉验证集选择 *λ*

从练习的前面部分，您观察到 *λ*  的值会显著影响正则多项式回归在训练和交叉验证集上的结果。特别是，没有正则化 （*λ* = 0） 的模型很好地拟合了 training 集，但没有泛化。相反，具有太多正则化 （*λ* = 100） 的模型不能很好地拟合训练集和测试集。λ的良好选择 （例如，*λ* = 1）可以提供与数据的良好拟合。

在本节中，您将实现 一个自动方法来选择 *λ* 参数。具体而言，您将使用交叉验证集来评估每个 *λ* 值的优劣。 使用交叉验证集选择最佳λ值后，我们可以评估测试集上的模型，以估计模型在实际未见过的数据上的表现。

您的任务是完成 validationCurve.m 中的代码。具体来说，您应该使用 trainLinearReg 函数来训练使用不同的 λ 值的模型 ，并将训练误差计算为nd 交叉验证误差。

您应该在以下范围内尝试 *λ*：{0*，*0*。*001*，*0*.*003*，*0*.*01*，*0*.*03*，*0*.*1*，*0*.*3*，*1*，*3*，*10}.

0

1

2

3

4

5

6

7

8

9

10

0

2

4

6

8

10

12

14

16

18

20

λ

错误

火车

交叉验证

图 9：使用交叉验证集选择 *λ*

完成代码后，ex5.m的下一部分将运行您的函数可以绘制误差v.s的交叉验证曲线。 *λ* 允许您选择要使用的 *λ* 参数。您应看到类似于图 9 的绘图。在此图中，我们可以看到 λ 的最佳值 在 3 左右。由于数据集的定型和验证拆分的随机性，交叉验证错误有时可能低于训练误差。

*您现在应该提交您的解决方案。*

## 可选（未评分）练习：计算测试集错误

在本练习的前一部分中，您实现了代码来计算正则化参数 λ 的各种值的交叉验证误差。但是，为了更好地指示模型在现实世界中的表现，重要的是在训练的任何部分中未使用的测试集上评估“最终”模型（即，它既不用于选择*λ*参数，也不用于学习模型参数*θ*）。

对于此可选（未评分）练习，应使用找到的 *λ* 的最佳值来计算检验误差。在我们的交叉验证中，我们获得了λ = 3的测试误差3.8599 。

*您无需为此可选（未评分）练习提交任何解决方案。*

## 可选（未评分）练习：使用随机选择的示例绘制学习曲线

在实践中，特别是对于小型训练集，当您绘制学习曲线来调试算法时，在多个随机选择的示例集之间求平均值通常有助于确定训练误差和交叉验证误差。

具体而言，要确定 *i* 示例的训练错误和交叉验证错误，应首先 从训练集中随机选择 *i* 示例，从交叉验证集中随机选择 i 示例。然后，您将使用随机选择的训练集学习参数 *θ*，并在随机选择的训练集和交叉验证集上评估参数 θ。然后，上述步骤应重复多次（例如50），并且应使用平均误差来确定i示例的训练误差和交叉验证误差 。

对于此可选（未评分）练习，应实施上述策略来计算学习曲线。作为参考，图 10 显示了我们在 *λ* = 0 的情况下获得的多项式回归的学习曲线*。*01.由于随机选择示例，您的 figure 可能会略有不同。

*您无需为此可选（未评分）练习提交任何解决方案。*

多项式回归学习曲线 （lambda = 0.010000）

0

2

4

6

8

10

12

0

10

20

30

40

50

60

70

80

90

100

训练示例数

错误

火车

交叉验证

图 10：可选（未评分）练习：随机选择示例的学习曲线

# 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。下面的g是本练习每个部分的评分方式的细分。

|  |  |  |
| --- | --- | --- |
| **部分** | **提交的文件** | **点** |
| 正则化线性回归成本  功能 | linearRegCostFunction.m | 25 积分 |
| 正则化线性回归梯度 | linearRegCostFunction.m | 25 积分 |
| 学习曲线 | learningCurve.m | 20 积分 |
| 多项式特征映射 | polyFeatures.m | 10 积分 |
| 交叉验证曲线 | validationCurve.m | 20 积分 |
| 总积分 |  | 100 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)