编程练习 2：逻辑回归

机器学习

# 介绍

在本练习中，您将实现逻辑回归并将其应用于两个不同的数据集。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。

在开始练习时，您需要下载入门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，请在启动此 exe rcise 之前，使用 Octave/MATLAB 中的 cd 命令更改为此目录。

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

ex2.m - Octave/MATLAB 脚本，用于引导您完成练习 ex2 reg.m - Octave/MATLAB script 用于练习的后续部分 ex2data1.txt - 练习前半部分的训练集 ex2data2.txt - 练习后半部分的训练集 submit.m - 将您的解决方案发送到我们服务器的提交脚本 mapFeature.m - Func生成多项式特征 plotDecisionBoundary.m - 绘制分类器决策边界的函数

[*?*] plotData.m - 绘制 2D 分类数据的函数 [*？*] sigmoid.m - Sigmoid Function

[*?*] costFunction.m - Logistic Regression Cost Function

[*?*] predict.m - 逻辑回归预测函数

[*?*] 成本函数Reg.m - 正则化逻辑回归成本

*?* 表示您需要完成的文件

在整个练习过程中，您将使用 脚本 ex2.m 和 ex2 reg.m。这些脚本为问题设置数据集，并调用您将编写的函数。您无需修改其中任何一个。您只需按照本作业中的说明修改其他文件中的 functi ons。

## 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。

# 逻辑回归

在本部分练习中，您将构建一个逻辑回归模型，以预测学生是否被大学录取。

假设您是一所大学部门的管理员，并且您希望根据每个申请人在两次考试中的成绩来阻止他们被录取的机会。您拥有以前申请人的历史数据，可将其用作逻辑回归的训练集。对于每个培训示例，您都有两个艾迈斯半导体的申请人分数和录取决定。

您的任务是构建一个分类模型，该模型根据这两项考试的分数估计申请人的入学概率。本大纲和 ex2.m 中的框架代码将指导您完成本练习。

## Vi对数据进行可视化

在开始实施任何学习算法之前，如果可能的话，将数据可视化总是好的。在 ex2.m 的第一部分中，代码将加载数据，并通过调用函数 plotData 将其显示在二维图上。

现在，您将在 plotData 中完成代码 ，以便它显示如图 1 所示的数字，其中轴是两个考试分数，正反示例使用不同的标记显示。



图 1：训练数据的散点图

为了帮助您更熟悉绘图，我们将 plotData.m 留空，以便您可以尝试自己实现它。但是，这是一个*可选（未评分）练习*。我们还在下面提供了我们的实现，因此您可以复制它或引用它。如果您选择复制我们的示例，请确保您通过查阅Octave / MATLAB文档来了解其每个命令的作用。

|  |
| --- |
| % 查找正负示例的指数 pos = find（y==1）; neg = find（y == 0）;  % 绘图 示例 绘图（X（pos， 1）， X（pos， 2）， 'k+'，'LineWidth'， 2， ...  “标记大小”， 7）;plot（X（neg， 1）， X（neg， 2）， 'ko'， 'MarkerFaceColor'， 'y'， ...  “标记大小”， 7）; |

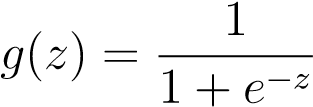
## 实现

### 热身运动：乙状结肠功能

在开始使用实际成本函数之前，回想一下逻辑回归假设定义为：

*hθ*（*x*） = *g*（*θTx*）*，*

其中函数 *g* 是乙状结肠函数。Sigmoid 函数定义为：

*.*

第一步是在 sigmoid.m 中实现此函数，以便程序的其余部分可以调用它。完成后，尝试通过在 Octave/MATLAB 命令行调用 sigmoid（x） 来测试几个值。对于 x 的大正值，sigmoid 应接近 1，而对于较大的负值，sigmoid 应接近 0。评估 sigmoid（0） 应该正好得到 0.5。您的代码还应该与 vectors 和矩阵一起使用。 **对于矩阵，您的函数应在每个元素上执行 sigmoid 函数。**

您可以通过在 Octave/MATLAB 命令行中键入 submit 来提交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。您可以从作业的网页获取提交令牌。

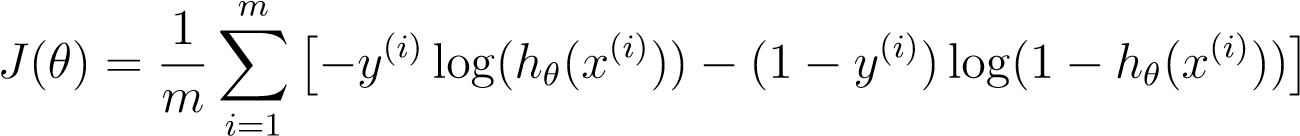
*您现在应该提交您的解决方案。*

### 成本函数和梯度

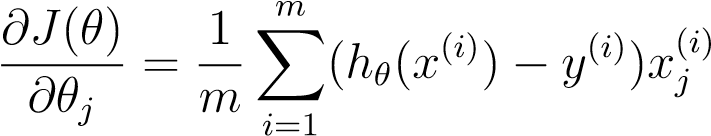
现在，您将实现逻辑回归的成本函数和梯度。

在 costFunction.m 中完成代码以返回成本和梯度。

回想一下，逻辑回归中的成本函数为

*,*

成本的梯度是 same 长度的向量，作为 *θ*，其中第 *j*个元素 （对于 *j* = 0*，*1*,...，n*） 定义如下：



请注意，虽然此梯度看起来与线性回归梯度相同，但公式实际上是不同的，因为线性回归和逻辑回归具有不同的 *hθ*（*x*） 定义。

完成后，ex2.m 将使用 *θ* 的初始参数调用您的 costFunction。您应该看到成本约为 0.693。

*您现在应该提交您的解决方案。*

### 使用 fminunc 学习参数

在前面的赋值中，您可以通过实现分级下降来计算线性回归模型的最优参数 。您编写了一个成本函数并计算其梯度，然后相应地采取了梯度下降步骤。这一次，您不是采取梯度下降步骤，而是使用一个名为fminunc的Octave / MATLAB内置函数。

Octave/MATLAB 的 fminunc 是一个优化求解器，用于查找无约束函数的最小值[[2]](#footnote-2)。对于逻辑回归，您希望使用参数 *θ* 优化成本函数 *J*（*θ*）。

具体而言，您将使用 fminunc 在给定固定数据集（*X* 和 *y*  值）的情况下，为逻辑回归成本函数 找到最佳参数 *θ*。You 将传递以下 输入：

* 我们尝试优化的参数的初始值。
* 一个函数，当给定训练集和特定 *θ* 时，计算数据集 （*X*， *y*） 相对于 *θ* 的逻辑回归成本和梯度

在 ex2.m 中，我们已经编写了代码来调用 具有正确参数的 fminunc。

|  |
| --- |
| % Set options for fminunc options = optimset（'GradObj'， 'on'， 'MaxIter'， 400）;  % 运行 fminunc 以获得最优的 θ % 此函数将返回 θ 和 成本 [theta， cost] = ...  fminunc（@（t）（costFunction（t， X， y））， initialtheta， options）; |

在此代码片段中，我们首先定义了要与 fminunc 一起使用的选项。具体来说，我们将GradObj选项设置为on，这告诉fminunc我们的函数同时返回成本和梯度。这允许 fminunc 在最小化函数时使用梯度。此外，我们将 MaxIter 选项设置为 400，以便 fminunc 在终止之前最多运行 400 步。

为了指定我们正在最小化的实际函数，我们使用“简写”来指定带有@（t）（costFunction（t，X，y））的函数。这将创建一个参数为 t 的函数，该函数调用您的 costFunction。这使我们能够包装成本函数以与fminunc一起使用。

如果您正确完成了成本函数，fminunc 将收敛于正确的优化参数，并返回成本和 θ 的最终值。请注意，通过使用fminunc，您不必自己编写任何循环，也不必像梯度下降那样设置学习速率。这一切都是由fminunc完成的：你只需要提供一个计算成本和梯度的函数。

一旦 fminunc 完成，ex2.m 将使用 *θ* 的最优参数调用您的 costFunction 函数。你看到成本是左右

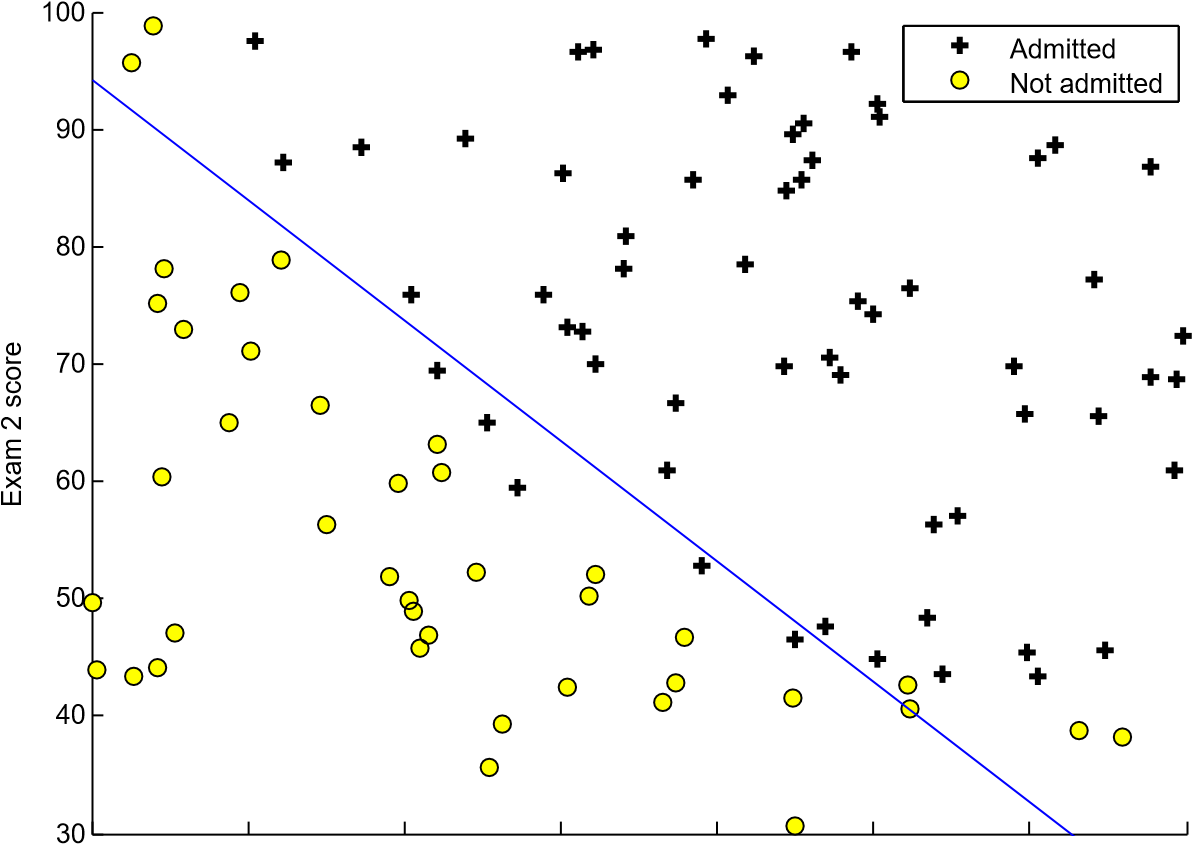
0.203.

然后，此最终 *θ* 值将用于在训练数据上绘制决策边界，从而生成类似于图 2 的数字。我们还鼓励您查看 plotDecisionBoundary.m 中的代码，了解如何使用 *θ* 值绘制这样的边界。

### 评估逻辑回归

了解参数后，您可以使用该模型来预测特定学生是否会被录取。对于考试 1 分数为 45 且考试 2 分数为 85 的学生，您应该 期望看到 0.776 的录取概率。

评估我们发现的参数质量的另一种方法是查看学习模型在我们的训练集上的预测程度。在此



+ 30 40 50 60 70 80 90 100 元

考试 1 分数

图 2：使用决策边界训练数据

部分，您的任务是完成predict.m中的代码。预测 函数将在给定数据集和学习参数向量 θ 的情况下生成“1”或“0”预测。

在 predict.m 中完成代码后， ex2.m 脚本将继续通过计算分类器正确示例的百分比来报告分类器的训练准确性。

*您现在应该提交您的解决方案。*

# 正则化逻辑回归

在本部分练习中，您将实现 正则逻辑回归，以预测来自制造工厂的微芯片是否通过质量保证 （QA）。在QA期间，每个微芯片都经过各种测试，以确保其正常运行。

假设您是factory的产品经理，并且您在两个不同的测试中获得了某些微芯片的测试结果。从这两个测试中，您想确定微芯片是应该被接受还是被拒绝。为了帮助您做出决定，您在 pas t 微芯片上有一个测试结果数据集，您可以从中构建逻辑回归模型。

您将使用另一个脚本 ex2 reg.m 来完成这部分练习。

## 可视化数据

与本练习的前几部分类似，plotData 用于生成如图 3 类似的图形，其中轴是两个测试分数，正 （*y* = 1，接受）和负 （*y* = 0， 被拒绝） 示例使用不同的标记显示。

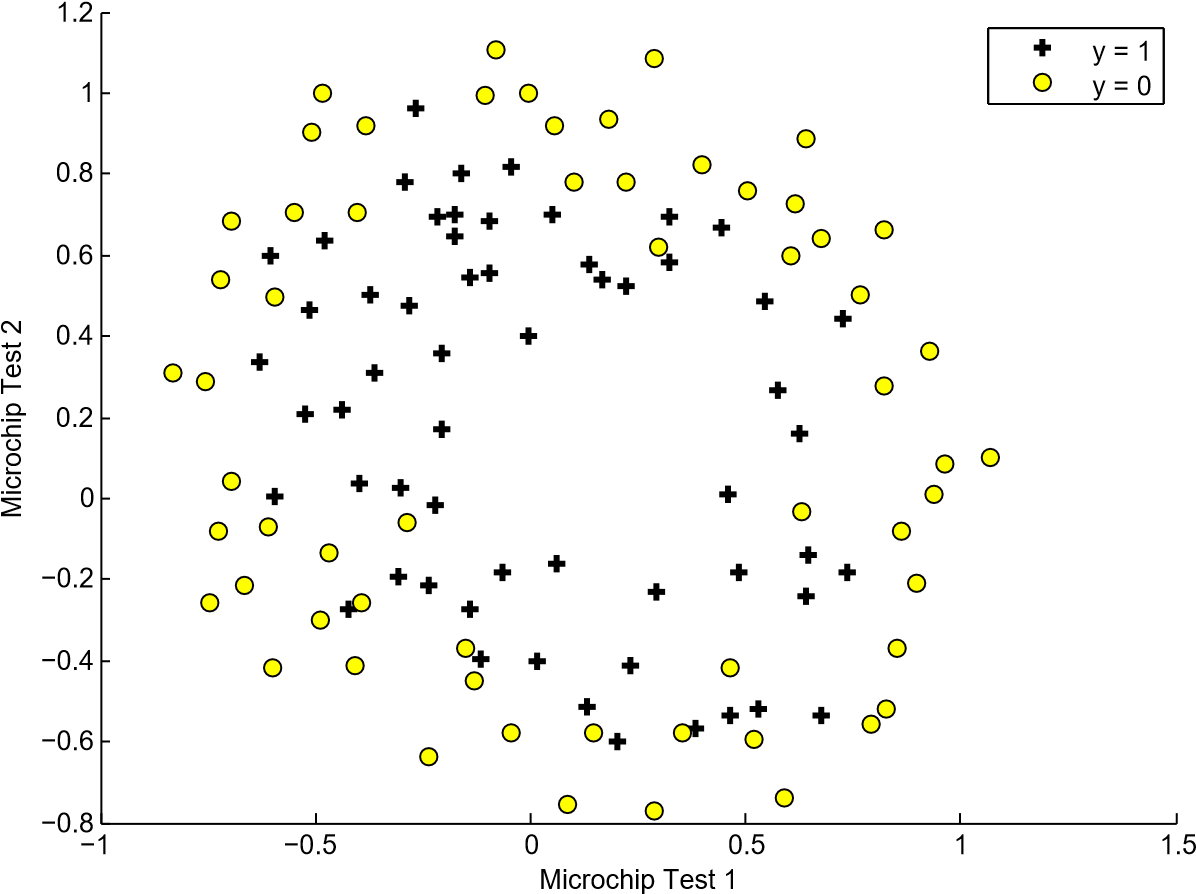


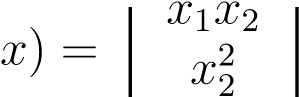
Figure 3：训练数据图

图 3 显示，我们的数据集不能通过一条直线穿过图来划分为正反示例。因此，逻辑回归的直接应用在此数据集上表现不佳，因为逻辑回归只能找到线性决策边界。

## 特征映射

更好地拟合数据的一种方法是从每个数据点创建更多要素。在提供的函数 mapFeature.m 中，我们将特征映射到 *x*1 和 *x*2  的所有多边形项中，直到第六次方。

|  |  |  |
| --- | --- | --- |
|            | 1 *x*1 *x*2  *x*21 |            |

  

地图功能（

 3 

 *x*1 

 ... 





 *x*1*x*52  

*x*62

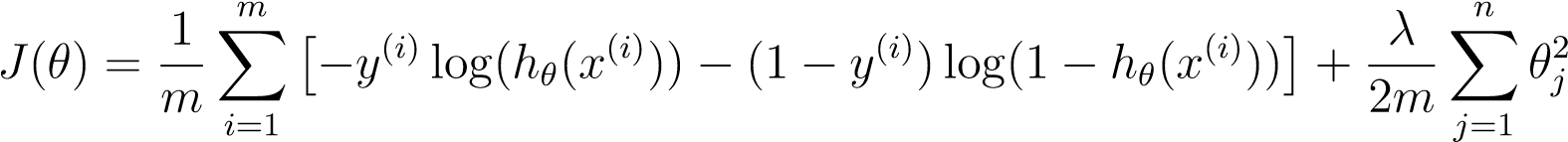
作为这种映射的结果，我们的两个特征向量（两个QA测试的分数）已经转换为28维向量。在此高维特征向量上训练的逻辑回归分类器将具有更复杂的决策边界，并且在我们的二维图中绘制时将显示为非线性。

虽然特征映射允许我们构建更具表现力的分类器，但它也更容易受到过拟合的影响。在本练习的下一部分中， 您将实现正则化的 logistic 回归来拟合数据，并亲眼看看正则化如何帮助解决过拟合问题。

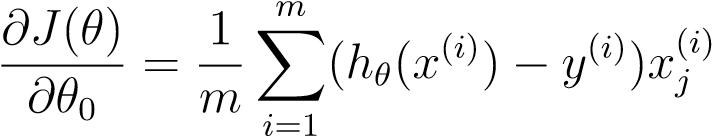
## 成本函数和梯度

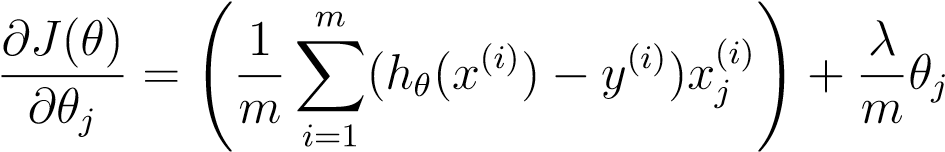
现在，您将实现代码来计算正则化逻辑回归的成本函数和梯度。在 costFunctionReg.m 中完成代码以返回成本和梯度。

回想一下，逻辑回归中的正则化成本函数为

*.*

注意 that 你不应该正则化参数 *θ*0。在 **Octave/MATLAB**中，回想一下索引从1开始，因此，您不应该在代码中正则化 θ（1） 参数（对应于 *θ*0）。成本函数的梯度是一个向量，其中 the *j*th 元素定义如下：

 对于 *j* = 0

 对于 *j* ≥ 1

完成后，ex2 reg.m 将使用 *θ*  的初始值（初始化为全部为零）调用 costFunctionReg 函数。您应该看到成本约为 0.693。

*您现在应该提交您的解决方案。*

### 使用 fminunc 学习参数

与前面的部分类似，您将使用 fminunc 来学习最佳参数 *θ*。如果您已正确完成正则化逻辑回归 （costFunctionReg.m） 的成本和梯度，则应该能够逐步完成 ex2 reg.m 的下一部分，以使用 fminunc 学习参数 *θ*。

## 绘制决策边界

为了帮助你可视化这个分类器学习的模型，我们提供了函数plotDecisionBoundary.m，它绘制了分隔正和负示例的（非线性）决策边界。在 plotDecisionBoundary.m 中，我们通过计算分类器在均匀分布网格上的预测来绘制非线性十进制边界，然后绘制预测从 *y* = 0 到 *y* = 1 变化的等值线图。

在学习了参数*θ*之后，ex reg.m中的下 一步将绘制一个类似于图4的决策边界。

## 可选（未评分）练习

在本部分练习中，您将尝试数据集的不同正则化参数，以了解正则化如何防止过度拟合。

请注意，当您改变 *λ* 时，决策边界的变化。使用较小的*λ*，您应该发现分类器几乎每个训练示例都正确，但绘制了一个非常复杂的边界，从而过度拟合了数据（图5）。这不是一个好的决策边界：例如，它预测 *x*  处的点 = （−0*.*25*，*1*.*5） 被接受 （*y* = 1），考虑到训练集，这似乎是一个不正确的决定。

使用较大的 *λ*，您应该看到一个图，该图显示了 一个 更简单的决策边界，该边界仍然很好地分离了正数和负数。H欠法，如果 *λ* 设置为太高的值，您将无法获得良好的拟合，并且决策边界不会很好地遵循数据，从而欠拟合数据（图

6）。

*您无需为这些可选（未评分）练习提交任何解决方案。*

lambda = 1

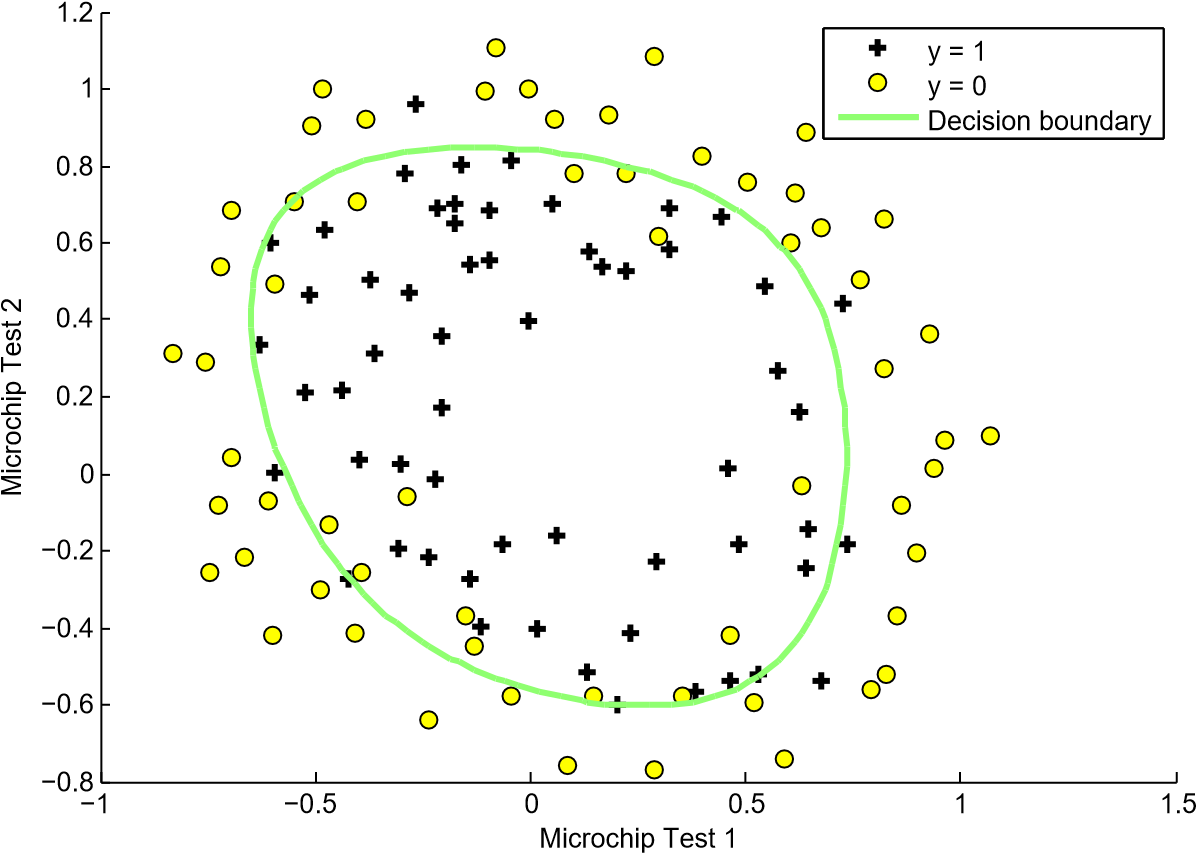


图 4：具有决策边界的训练数据（*λ* = 1）

lambda = 0

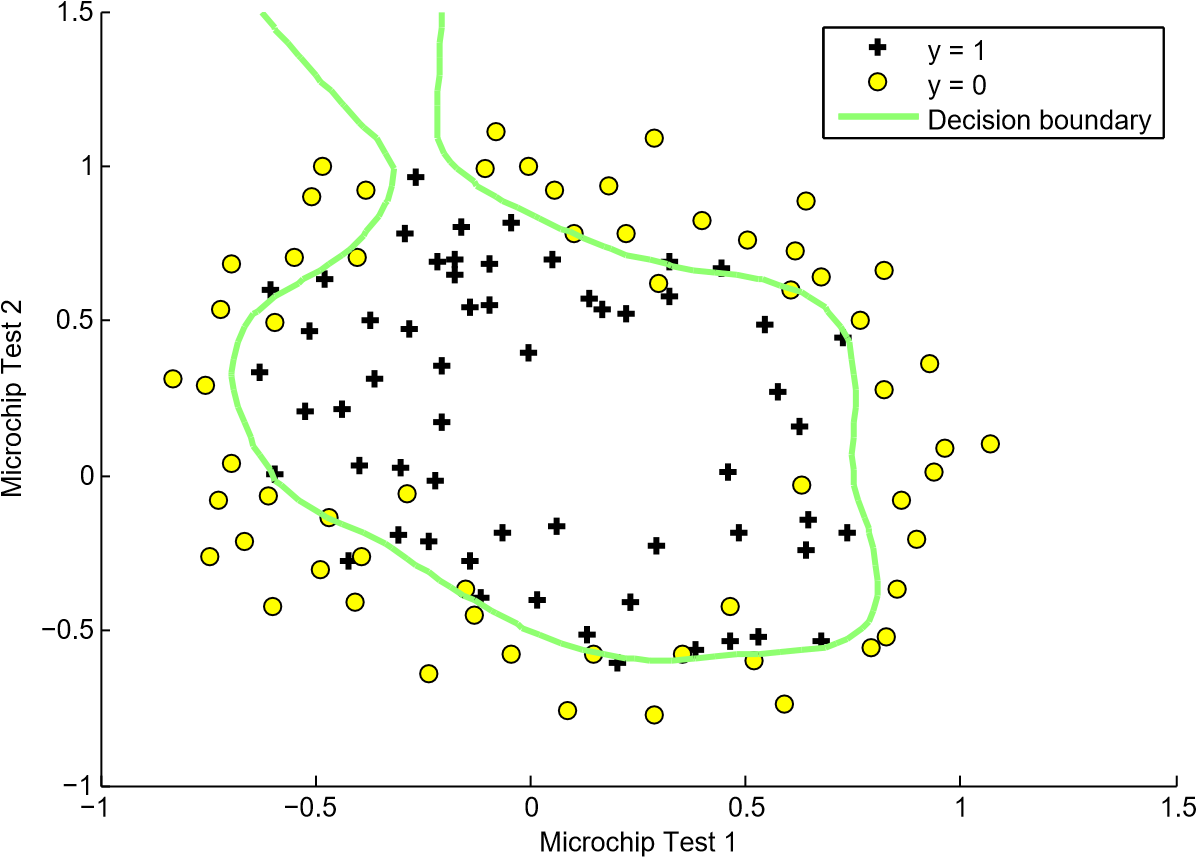


图 5：无正则化（过拟合）（*λ* = 0）

λ = 100

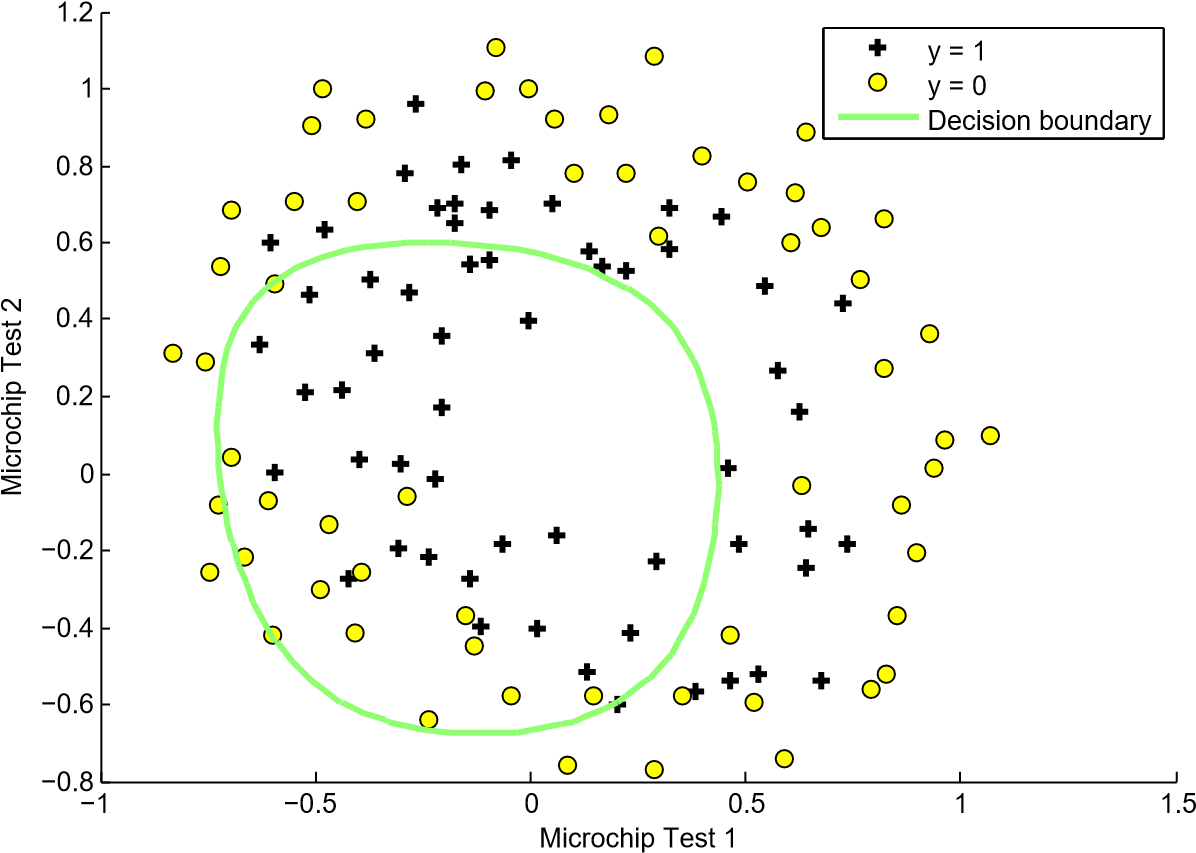


图 6：正则化过多（欠拟合）（*λ* = 100）

# 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** |
| 乙状结肠功能 | sigmoid.m | 5 积分 |
| 逻辑回归的计算成本 | costFunction.m | 30 积分 |
| 逻辑回归的梯度 | costFunction.m | 30 积分 |
| 预测函数 | 预测.m | 5 积分 |
| 正则化 LR 的计算成本 | costFunctionReg.m | 15 积分 |
| 正则化 LR 的渐变 | costFunctionReg.m | 15 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)
2. 优化中的约束通常是指对参数的约束，例如，约束可能值的约束*我*可以ke （例如，*我*≤1）.逻辑回归没有这样的约束，因为*我*允许采用任何实际值。 [↑](#footnote-ref-2)