实验报告-ex2

编程软件：MATLAB，notepad++

编程环境：Windows10

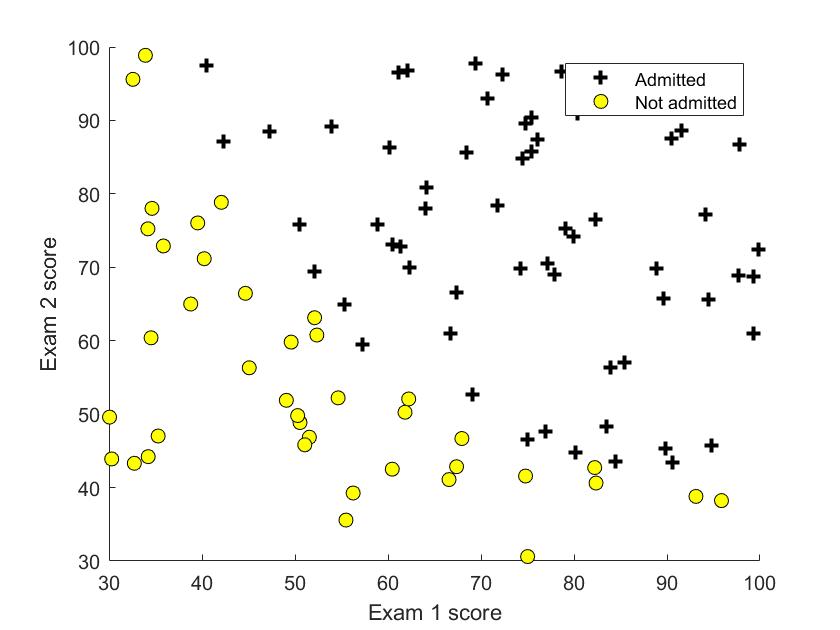
作业任务：本次实验旨在了解logistic回归的应用。有ex2和ex2\_reg两个函数，函数中分别套用了相应的函数，所以需要根据这两个目录函数的指示和实验手册，完成对应的四个函数。

1 Logistic Regression

本次实验目的是用logistic回归来预测一个学生是否可以被大学录取，从描述中可知这是一个classification问题。

1.1 Visualizing the data

在回归算法前，先对data set有一个可视化的了解。由于我对MATLAB的plot功能不是很了解，所以这一部分选择参考实验手册上给出的方法。结果如下：



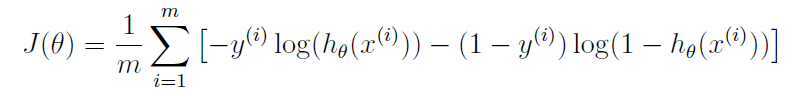
1.2 Implementation

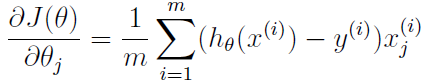
1.2.1 Warmup exercise: sigmoid function

这一部分要求实现sigmoid函数，前面已知函数定义g(z)=，对于logistic回归，z=θTx。现在需要实现的就是g(z)。实现过程较为简单，只需要考虑到矩阵需要一个个元素地计算，所以需要一个二层循环来逐个计算。

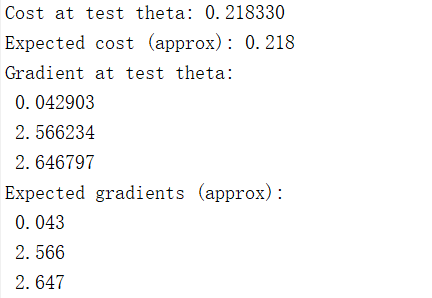
1.2.2 Cost function and gradient

这部分实现的是logistic回归中两个重要的函数，cost function和gradient，函数表达式如下：



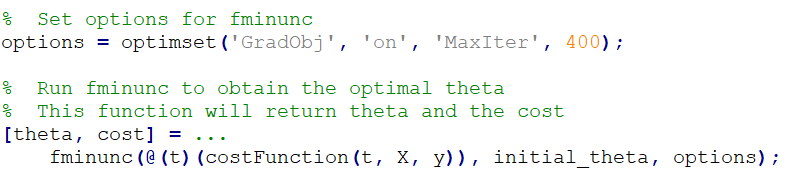


这部分的核心其实是h(x)的实现，因为J实际上就是套用公式即可，而偏导数也已经在ex1中用向量化实现。而h(x)由于前面sigmoid函数的实现也可直接写出。结果如下：



1.2.3 Learning parameters using fminunc

这部分运用了优化算法fminunc，由于之前已经完善了cost function的算法，故此处不需要添加代码。以下是对fminun函数运用的解释：



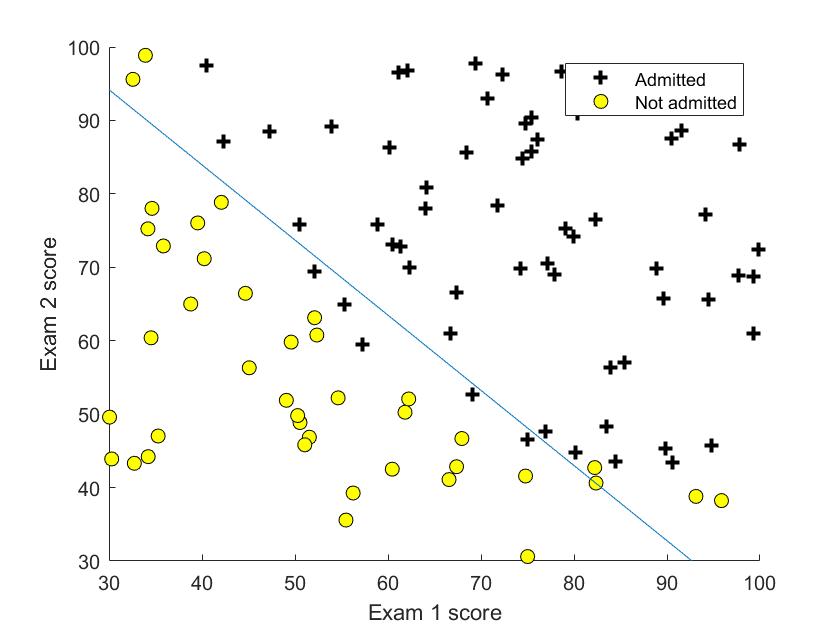
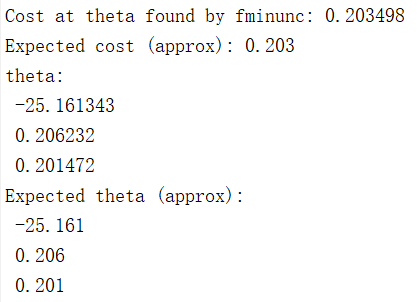
要使用fminun函数的如下形式：[x,fval,exitflag] = fminunc(fun,x0,options); 该方法需要定义3个输入参数fun,x0和options。

①其中@(t)(costFuction(t,X,y))是fun。@是句柄，句柄的使用格式：变量名=@(输入参数列表)运算表达式。这里的t相当于θ

②第二个输入参数x0定义为 initial\_Theta 该参数为一个(n+1)×1的矩阵。该列向量为用户自定义梯度下降法的输出参数，使用前要预先初始化。

③第三个输入参数为options。该参数的的作用包括是否使用用户自定义的梯度下降公式（GradObj）以及迭代次数（MaxIter）。本题中使用了自定义的梯度下降公式（costFunction中的gradien返回值），并且把迭代次数设为迭代400次。

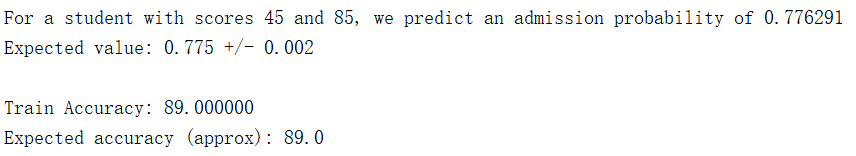
以下为输出结果：



1.2.4 Evaluating logistic regression

第一部分是对已知新数据进行结果预测。这里用到了sigmoid函数，只需要算出其假设函数的值即可。

第二部分需要完善predict函数。这个函数用来计算出training set中所有数据的假设函数值，然后在ex2中取平均值来看回归函数的准确程度。方法较简单，只需要对每一个数据判断是否≥0.5即可。以下为结果：

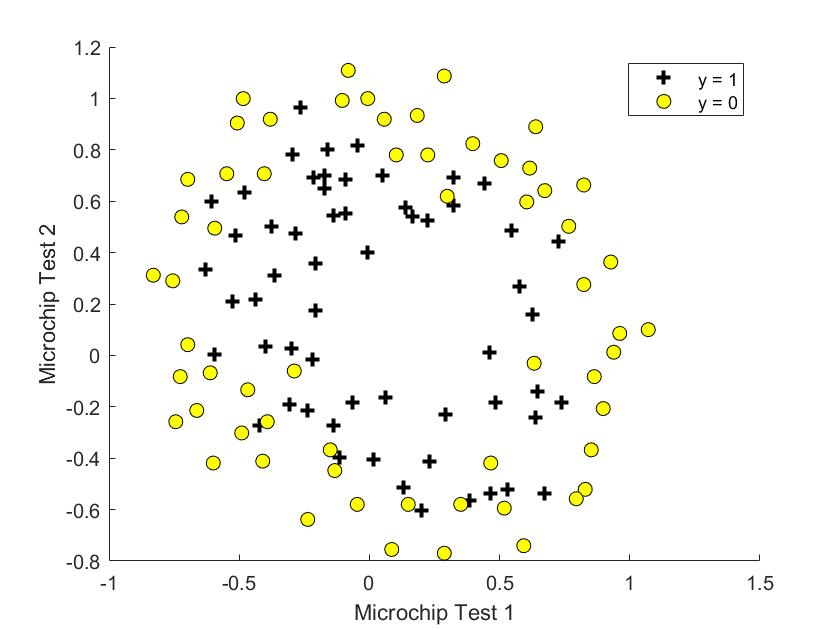


2 Regularized logistic regression

这一部分更进一步，需要用带有正则化的logistic回归来预测加工厂的微芯片是否通过质检。

2.1 Visualizing the data

这一部分是ex2\_reg函数中已经给出的，对已有的training set进行可视化，得到如下图：



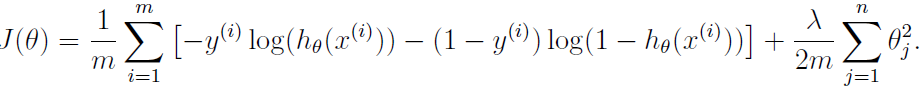
从图中可以看出，这个数据组不可以用线性函数来划分，所以我们需要多项式回归。

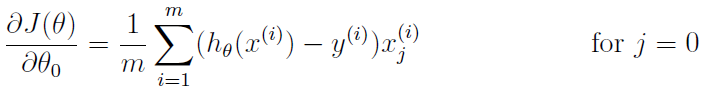
2.2 Feature mapping

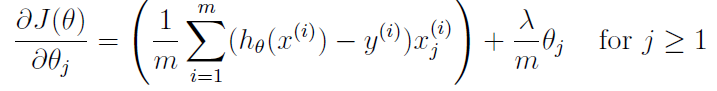
这一部分就是直接将非线性的函数转化为线性函数，每一个1-6次方的x都赋予了θ。因此，这种情况会造成overlapping的问题。

2.3 Cost function and gradient

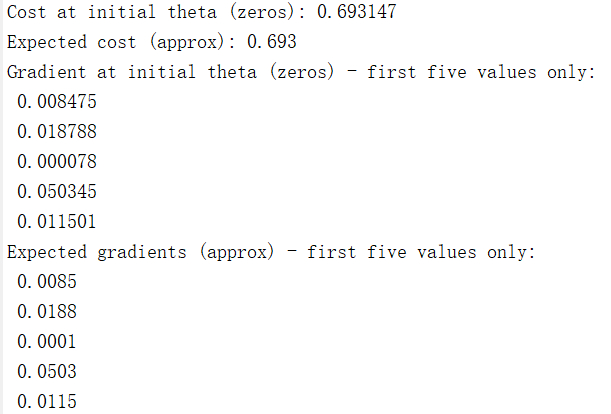
这一部分与ex2函数中调用的costfunction函数类似，但是多了一个惩罚项，需要在代码中体现。Cost function和偏导数式子如下：





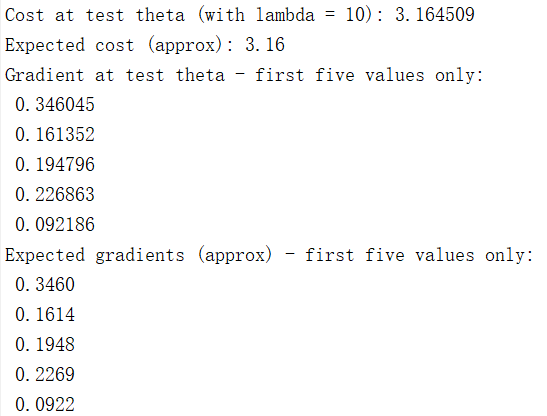


整体思路较简单，与ex2中方法一致，只需要按照公式加上惩罚项。但是需要注意，无论是J还是grad，都不能将theta(1)，即θ0考虑进去，即该项不需要正则化。所以我选择在J中减去theta(1)^2，而对于grad(1)进行单独运算。



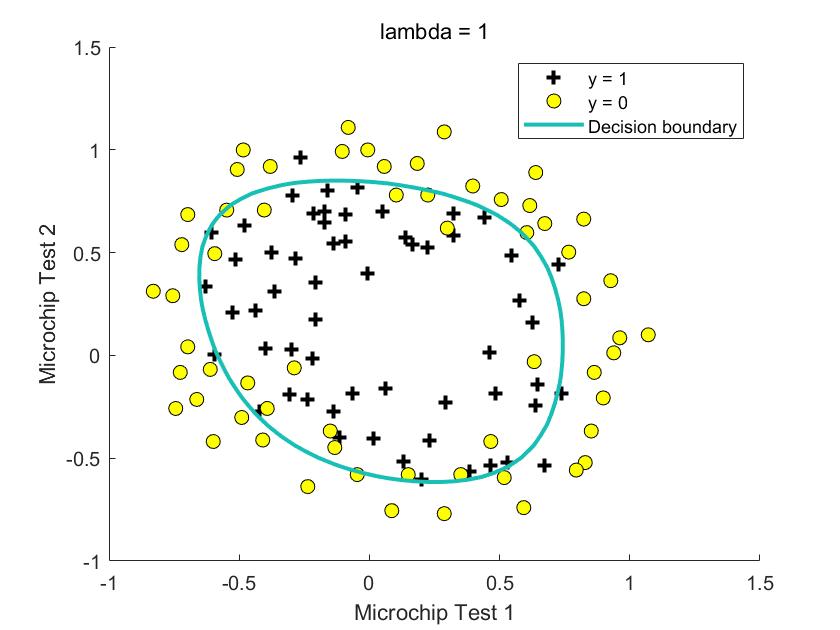
2.3.1 Learning parameters using fminunc

fminunc是建立在cost function的基础上的，所以一旦cost function和gradient都完成后，fmimunc与是否正则化无关。结果如下：



2.4 Plotting the decision boundary

同样画出decision boundary就好了~



○至此提交实验部分全部完成！

2.5 Optional (ungraded) exercises

这部分探索了不同的λ对回归算法的影响，通过图像显示：

