《K-Means解题报告》

姓名：刘丰恺

班级：软件1412

学号：201492167

1. **简介：**

很多时候我们不只需要做回归学习，也需要做任务分类。当我们需要做任务分类的时候，要如何把它转化为回归模型呢：只需要找一个单调可微函数将分类任务的真实标记与线性回归模型的预测值联系起来。

考虑一个二分类任务，我们将他的输出标记抽象成一个类似的样式，此时线性回归模型产生的预测值是一个实数，于是，我们需要将实数 转换为0/1值。最理想的函数是一种被称为“单位阶跃函数”的函数：

此时若预测值大于0就判为正例，小于0判为反例，预测值为临界值则任意判别。

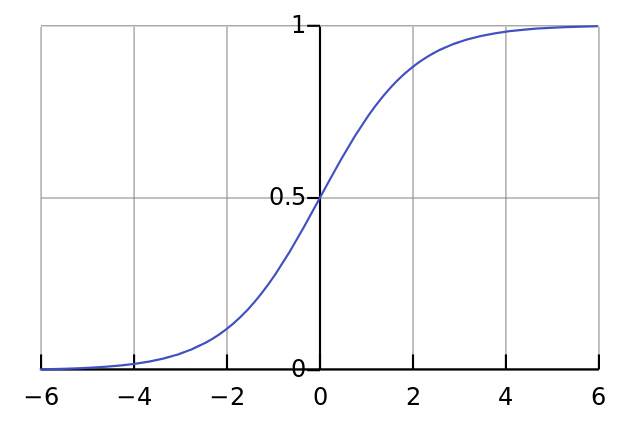


图1：单位阶跃函数

从图中，我们可以看出，单位阶跃函数不连续，因此不能直接用作单调可微函数。我们希望能找到一个近似于单位阶跃函数的替代函数，并希望它单调可微。而常用于替代的函数就是以下这个函数：

这个函数将转化为一个接近0或1的值，并且在附近输出值变化很陡，将该函数代入，可得到：

该函数可变为：

若将视为样本作为正例的可能性，则是其反例可能性，两者的比值为：

将该数称为“几率”，则几率取对数就得到

由此可以看出，函数实际上是在用线性回归模型的预测结果去逼近真实标记的对数几率，因此，其对应的模型称为“对数几率回归”。

为了确定和，可将式中的视为类后验概率估计，则可以把对数几率函数重写成

则：

于是，就可以使用“极大似然法”来估计和。给定数据集，对率回归模型最大化“对数似然”

令每个样本属于其真实标记的概率越大越好，则：

将式子代入，得到：

为了简便，我们将简写为，并定义：

将其带入到线性回归的代价函数，并引入梯度下降法，可以得到：

对此函数求的偏微分，可以得到：

当此式等于0时，则可求得极小值。

但是，当比较复杂时，我们就要用到非线性的回归函数，此时的方程为：

偏微分为：

1. **结果展示及功能描述：**

**2.1代码：**

costFuction.m

功能：对上文中提到的计算和其关于的偏微分两个公式的实现，并且以返回相应的结果。

参数：theta - 初始化的theta向量

X – X向量

y – y向量

返回：J - 最小化代价

grad – J关于的偏微分

function [J, grad] = costFunction(theta, X, y)

%COSTFUNCTION Compute cost and gradient for logistic regression

% J = COSTFUNCTION(theta, X, y) computes the cost of using theta as the

% parameter for logistic regression and the gradient of the cost

% w.r.t. to the parameters.

% Initialize some useful values

m = length(y); % number of training examples

% You need to return the following variables correctly

J = 0;

grad = zeros(size(theta));

% ====================== YOUR CODE HERE ======================

% Instructions: Compute the cost of a particular choice of theta.

% You should set J to the cost.

% Compute the partial derivatives and set grad to the partial

% derivatives of the cost w.r.t. each parameter in theta

%

% Note: grad should have the same dimensions as theta

%

J =- 1 / m \* sum(y .\* log(sigmoid(X \* theta)) + (1-y) .\* log(1 - sigmoid(X \* theta)));

grad = 1 / m \* X' \* (sigmoid(X \* theta) - y);

% =============================================================

end

costFunctionReg.m

功能：实现非线性条件下的和其关于的偏微分两个公式的计算，并且返回相应结果。

参数: theta - 初始化的theta向量

X – X向量

y – y向量

lambda – 边界决策参量

返回：J - 最小化代价

grad – J关于的偏微分

function [J, grad] = costFunctionReg(theta, X, y, lambda)

%COSTFUNCTIONREG Compute cost and gradient for logistic regression with regularization

% J = COSTFUNCTIONREG(theta, X, y, lambda) computes the cost of using

% theta as the parameter for regularized logistic regression and the

% gradient of the cost w.r.t. to the parameters.

% Initialize some useful values

m = length(y); % number of training examples

% You need to return the following variables correctly

J = 0;

grad = zeros(size(theta));

% ====================== YOUR CODE HERE ======================

% Instructions: Compute the cost of a particular choice of theta.

% You should set J to the cost.

% Compute the partial derivatives and set grad to the partial

% derivatives of the cost w.r.t. each parameter in theta

predictions = sigmoid(X\*theta); % m x 1 predictions of hypothesis on all m examples

J = 1 / m \* (-y'\*log(predictions) - (1 - y)' \* log(1 - predictions)) + 1 / (2 \* m) \* lambda \* (theta' \* theta - (theta(1, 1))^2); % cost function

n = size(theta);

grad(1,1) = (1 / m \*(predictions - y)' \* X(:,1));

%size(grad(2:n,1));

%(1/m \*lambda\*theta(2:n,1))

grad(2:n, 1) = (1 / m \* (predictions - y)' \* X(:,2:n) )'+ 1 / m \* lambda \* theta(2:n,1);%+

% =============================================================

end

predict.m

功能：计算单位阶跃函数，并将结果转化为0与1的形式

参数：theta – theta向量

X – X向量

返回：p – 计算出来的预测向量

function p = predict(theta, X)

%PREDICT Predict whether the label is 0 or 1 using learned logistic

%regression parameters theta

% p = PREDICT(theta, X) computes the predictions for X using a

% threshold at 0.5 (i.e., if sigmoid(theta'\*x) >= 0.5, predict 1)

m = size(X, 1); % Number of training examples

% You need to return the following variables correctly

p = zeros(m, 1);

% ====================== YOUR CODE HERE ======================

% Instructions: Complete the following code to make predictions using

% your learned logistic regression parameters.

% You should set p to a vector of 0's and 1's

%

p(sigmoid(X\*theta)>0.5)=1;

% =========================================================================

end

sigmoid.m

功能：计算对数几率函数

参数：z – 要计算的值

返回：g – 计算结果

function g = sigmoid(z)

%SIGMOID Compute sigmoid functoon

% J = SIGMOID(z) computes the sigmoid of z.

% You need to return the following variables correctly

g = zeros(size(z));

% ====================== YOUR CODE HERE ======================

% Instructions: Compute the sigmoid of each value of z (z can be a matrix,

% vector or scalar).

g = 1 ./ (1 + exp(-z));

% =============================================================

end

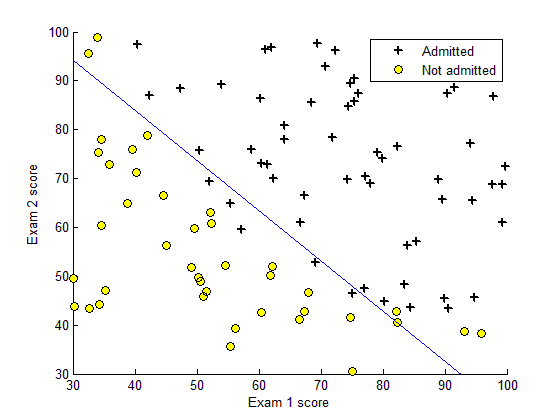
**2.2成果:**

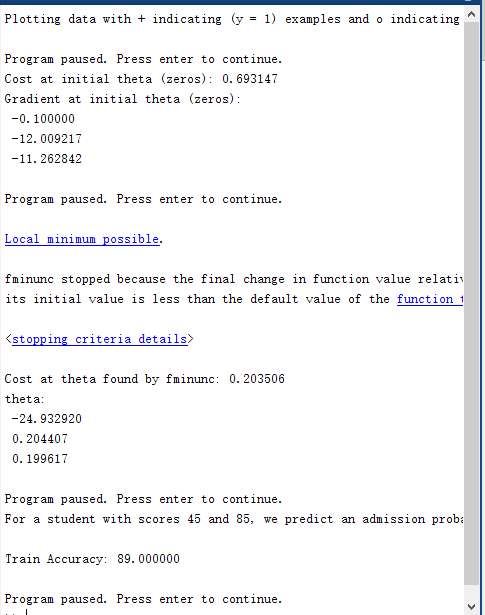
第一题：

首先，在拿到数据的时候绘制当前图像，根据所给数据的positive与negative情况使用不同的点进行标记，我们可以非常明显的看出，两种数据明显存在一条线性分界线，通过costFunction计算出在时，初始的距离，与其的偏微分。

将其导入最小化函数中，得到相应的与其的偏微分，就可以绘制出一条所有样本到其的欧氏距离之和最小的直线，可以得到当前决策边界并将其绘制在图上。

在得到这个之后，就可以开始预测相应情况了，我们导入数据exam1 为45，exam2为85，并且其为positive，可以看出，我们的算法给出了他的通过率为77.4%，本次训练的准确度为89%。

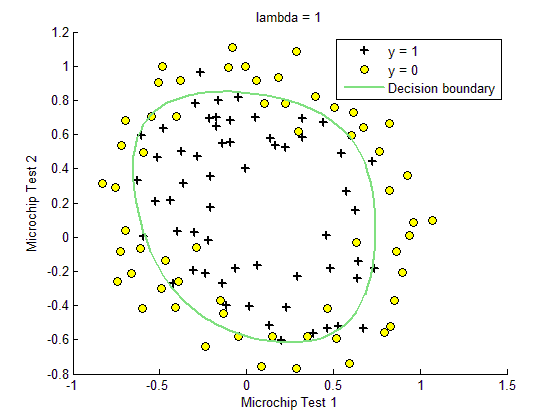


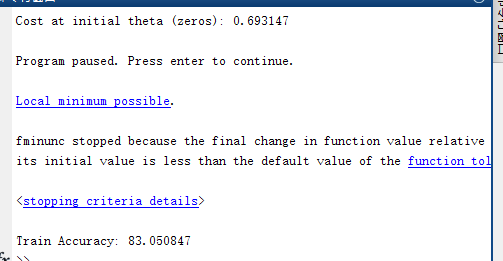


第二题：

在我们拿到数据并将其绘制至图中后，我们可以发现，这批数据并不符合线性分类的情况，我们只能通过公式去构建一个非线性的分类情况。

我们首先利用mapFeature函数提取出所有的X的多项式组合，并将该数据更新至X的值，然后透过公式计算他们的决策边界，可以看到，公式的初始为0.63147，训练准确率约为83.1%。

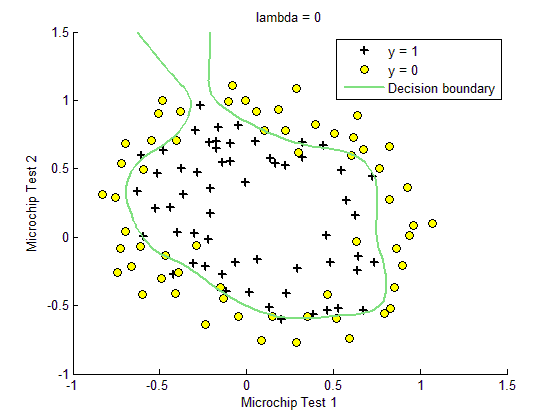


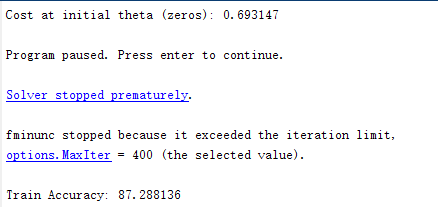


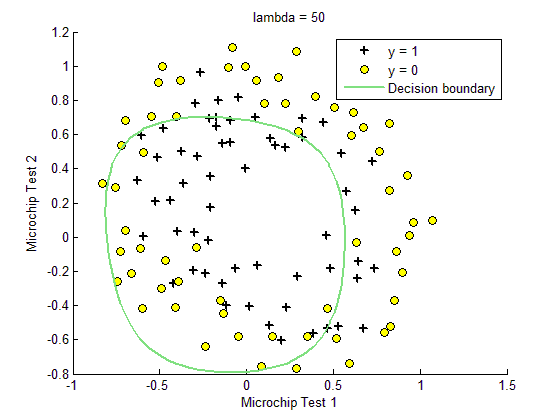
在完成以上实验后，还需要对不同的lambda取值进行分析结果，我选取了lambda = 0与lambda = 50 时进行分析。

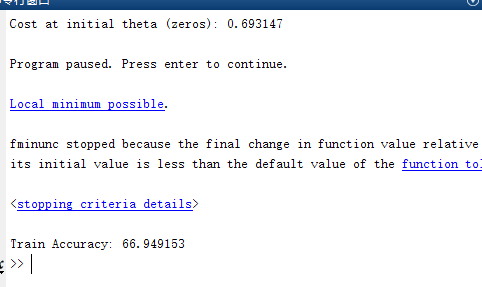
上文中我们已经提到，lambda的值主要影响数据的拟合情况。我们可以从图像中看出来，在lambda = 0的时候，所绘制的决策边界非常复杂而且曲折，同时它的拟合准确率也上升到了87.3%，但是由于边界实在太过复杂，存在过度拟合的情况。

当lambda = 50的时候，图像展示给我们一种更为简单且平滑的决策边界。然而，如果lambda设置的过于大，数据的决策边界就不是那么适合了。我们可以看到，训练的准确度降到了66.9%，当前决策边界已经不能很好的适应数据了，用该边界去预测结果，也很难得到我们想要的结果。









1. **小结：**

本次实验我们学习了逻辑回归，学会了将数据分类并且计算决策边界的方法，对机器学习有的一点初步的了解，并且巩固了课堂中学到的知识。