

**模式识别大作业**

题 目 logistic regression对数据分类

学 院 信息科学与工程

专 业 信息与通信工程

组 员 杜嘻嘻

指导教师 赵海涛

**完成日期： 2018 年 10 月22日**

**模式识别作业报告——logistic regression对数据分类**

组员：杜嘻嘻

最初选修模式识别这门课程，我并不了解模式识别是什么。不过，通过这几次赵海涛老师的辛勤教学，我对模式识别有了一定的了解，并通过本次针对logistic regression对数据分类的实验来巩固所学内容。

经过一段时间的程序编写及调试，最终完成了logistic regression对数据的分类。下面将详细说明我的解决过程。

**一、Logistic Regression简介**

[Logistic regression](https://baike.baidu.com/item/logistic%E5%9B%9E%E5%BD%92/2981575)主要在流行病学中应用较多，比较常用的情形是探索某疾病的危险因素，根据危险因素预测某疾病发生的概率，等等。例如，想探讨胃癌发生的危险因素，可以选择两组人群，一组是胃癌组，一组是非胃癌组，两组人群肯定有不同的体征和生活方式等。这里的因变量就是是否胃癌，即“是”或“否”，为两分类变量，[自变量](https://baike.baidu.com/item/%E8%87%AA%E5%8F%98%E9%87%8F/6895256)就可以包括很多了，例如年龄、性别、饮食习惯、幽门螺杆菌感染等。自变量既可以是连续的，也可以是分类的。通过logistic regression分析，就可以大致了解到底哪些因素是胃癌的危险因素。

Logistic regression也是当前业界比较常用的机器学习方法，用于估计某种事物的可能性。之前在经典之作《数学之美》中也看到了它用于广告预测，也就是根据某广告被用户点击的可能性，把最可能被用户点击的广告摆在用户能看到的地方，然后叫他“你点我啊！”用户点了，你就有钱收了。这就是为什么我们的电脑现在广告泛滥的原因了。

**二、Logistic Regression推导过程**

**1.基本原理**

Logistic Regression按照我自己的理解，可以简单的描述为这样的过程：

（1）找一个合适的预测函数，一般表示为*h*函数，该函数就是我们需要找的分类函数，它用来预测输入数据的判断结果。这个过程时非常关键的，需要对数据有一定的了解或分析，知道或者猜测预测函数的“大概”形式，比如是线性函数还是非线性函数。

（2）构造一个Cost函数（损失函数），该函数表示预测的输出（*h*）与训练数据类别（*y*）之间的偏差，可以是二者之间的差（*h-y*）或者是其他的形式。综合考虑所有训练数据的“损失”，将Cost求和或者求平均，记为*J(θ)*函数，表示所有训练数据预测值与实际类别的偏差。

（3）显然，*J(θ)*函数的值越小表示预测函数越准确（即*h*函数越准确），所以这一步需要做的是找到*J(θ)*函数的最小值。找函数的最小值有不同的方法，Logistic Regression实现可以用梯度下降法。

**2.具体过程**

Logistic Regression的具体过程，包括：选取预测函数，求解Cost函数和*J(θ)*，梯度下降法求*J(θ)*的最小值。

2.1构造预测函数

Logistic Regression虽然名字里带“回归”，但是它实际上是一种分类方法，用于两分类问题（即输出只有两种）。首先，需要先找到一个预测函数（*h*），显然，该函数的输出必须是两个值（分别代表两个类别），所以利用了Logistic函数，函数形式为：

 （1）

接下来需要确定数据划分的边界类型，接下来我们只讨论线性边界的情况。

对于线性边界的情况，边界形式如下：

 （2）

构造预测函数为：

 （3）

函数的值有特殊的含义，它表示结果取1的概率，因此对于输入x分类结果为类别1和类别0的概率分别为：

 （4）

2.2构造Cost函数

Cost函数和函数是基于最大似然估计推导得到的。下面详细说明推导的过程。（4）式综合起来可以写成：

 （5）

取似然函数为：

 （6）

对数似然函数为：

 （7）

最大似然估计就是要求得使取最大值时的，求得的就是要求的最佳参数，即：

 （8）

综上，可得：

 （9）

 （10）

2.3  梯度下降法求的最小值

求的最小值可以使用梯度下降法，根据梯度下降法可得的更新过程：

 （11）

式中为学习步长，下面来求偏导：

 （12）

上式求解过程中用到如下的公式：

 （13）

因此，（11）式的更新过程可以写成：

 （14）

因为式中本来为一常量，所以一般将省略，所以最终的更新过程为：

 （15）

另外，补充一下，2.2节中提到求得取最大值时的也是一样的，用梯度上升法求（7）式的最大值，可得：

 （16）

观察上式发现跟（14）式是一样的，所以，采用梯度上升发和梯度下降法是完全一样的，这也是《机器学习实战》中采用梯度上升法的原因。

**三、整体解决方案**

（1）logistic回归原理

我们先制造一份示例数据来看看logistic 回归是如何工作的。

首先调用模块：

import numpy as np

from sklearn import linear\_model, datasets

import matplotlib.pyplot as plt

from scipy.stats import norm

然后生成模拟数据并绘制图形：

np.random.seed(3)

n = 40

X = np.hstack((norm.rvs(loc=2, size=n, scale=2), norm.rvs(loc=8, size=n, scale=3)))

y = np.hstack((np.zeros(n),np.ones(n)))

plt.figure(figsize=(10, 4))

plt.xlim((-5, 20))

plt.scatter(X, y, c=y)

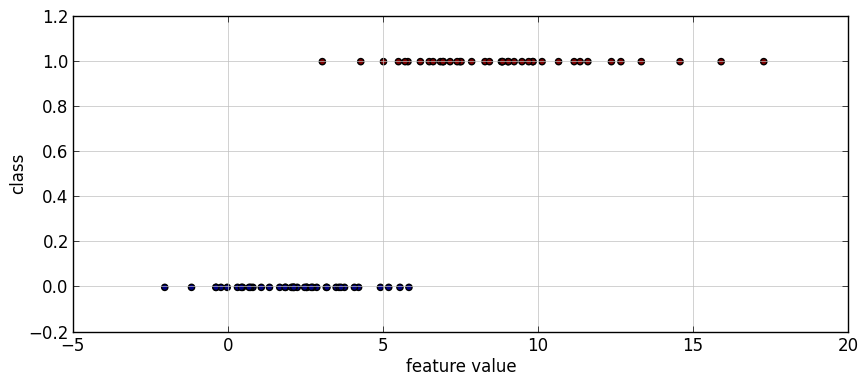
plt.xlabel("feature value")

plt.ylabel("class")

plt.grid(True, linestyle='-', color='0.75')

plt.savefig("E:/wulingfei/logistic\_classification/logistic\_classify1.png", bbox\_inches="tight")

我们就看到这样一张图：



上图有80个数据点，在模拟X的时候，我们使用了两个正态分布，分别制定各自的均值和方差，生成40个点。模拟Y的时候我们直接生成40个0和40个1。两个正态分布均值虽然不同，但在方差作用下，在X上的分布有重叠。我们假设这是数据中某个feature的表现，要求使用一个模型，根据这个feature来判别所属的Y类别。

（2）使用logistic regression处理iris数据

我们使用基于python的sklearn这个机器学习的包，其中自带Iris数据。我们要先读入这个数据：

from matplotlib import pyplot as plt

from sklearn.datasets import load\_iris

import numpy as np

试着把数据画出来：

data = load\_iris()

features = data['data']

feature\_names = data['feature\_names']

target = data['target']

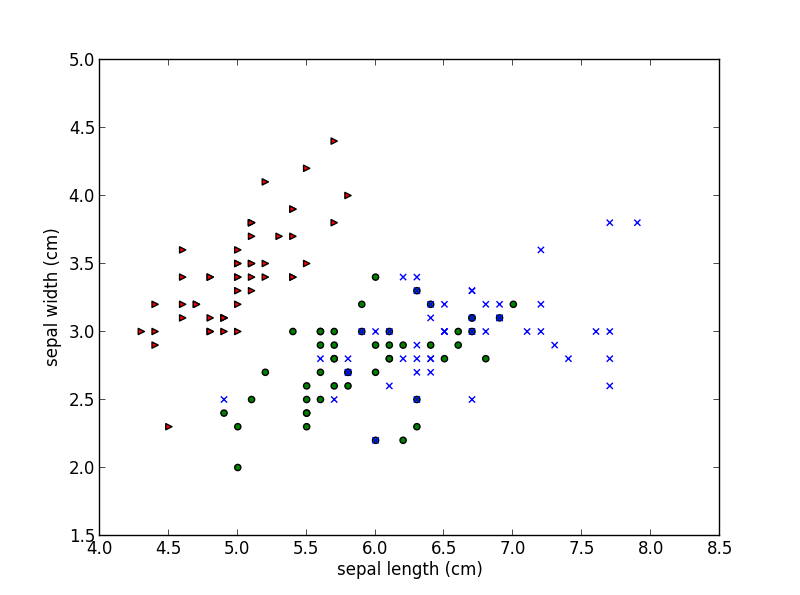
for t,marker,col in zip(xrange(3),">ox","rgb"):

plt.scatter(features[target == t,0], features[target == t,1], marker=marker,c=col)

plt.xlabel(feature\_names[0])

plt.ylabel(feature\_names[1])

使用上述命令，我们可以把数据集根据其两个特征画出来：



其中，三角形代表Setosa，圆形代表Versicolor，叉叉代表Virginica。

我们当然也可以写一个函数来画图，把四个特征的所有组合都画出来看一下数据在特征空间的分布，一共有六个二维投影：

def plotIrisData(x,y):

for t,marker,col in zip(xrange(3),">ox","rgb"):

plt.scatter(features[target == t,x], features[target == t,y], marker=marker,c=col)

plt.xlabel(feature\_names[x])

plt.ylabel(feature\_names[y])

figure(num=None, figsize=(12, 8), dpi=80, facecolor='w', edgecolor='k')

plt.subplot(231)

plotIrisData(0,1)

plt.subplot(232)

plotIrisData(0,2)

plt.subplot(233)

plotIrisData(0,3)

plt.subplot(234)

plotIrisData(1,2)

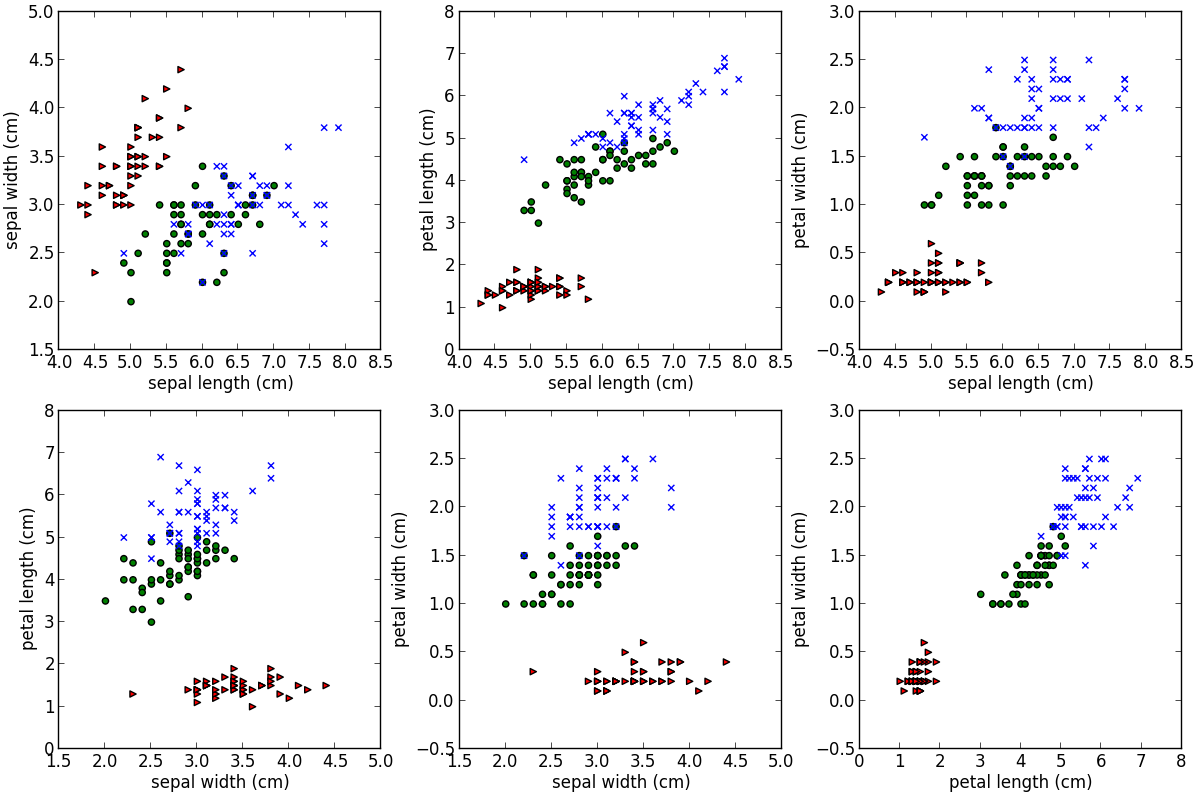
plt.subplot(235)

plotIrisData(1,3)

plt.subplot(236)

plotIrisData(2,3)

plt.tight\_layout(pad=0.4, w\_pad=0, h\_pad=1.0)



这张图展示了4维特征空间中的数据所有的二维投影。

这里的情况比之前的模拟数据复杂，应为我们现在的自变量有两个而不是一个（我们从四个维度中选取两个），因变量有三个而不是两个。不过幸好sklearn的fit函数和predict函数让我们处理起这些问题来变得很容易。

*#-----preprare data----------*

from sklearn import datasets

iris = datasets.load\_iris()

X = iris.data[:, :2]

Y = iris.target

*#-----fit data--------------------*

logclf = LogisticRegression(C=1e5)

logclf.fit(X, Y)

*# Plot the decision boundary. For that, we will assign a color to each*

*# point in the mesh [x\_min, m\_max]x[y\_min, y\_max].*

h = .02 *# step size in the mesh*

x\_min, x\_max = X[:, 0].min() - .5, X[:, 0].max() + .5

y\_min, y\_max = X[:, 1].min() - .5, X[:, 1].max() + .5

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h), np.arange(y\_min, y\_max, h))

Z = logclf.predict(np.c\_[xx.ravel(), yy.ravel()])

*# Put the result into a color plot*

Z = Z.reshape(xx.shape)

pl.figure(1, figsize=(8, 6))

pl.pcolormesh(xx, yy, Z, cmap=plt.get\_cmap('Spectral'))

*# Plot also the training points*

for t,marker,col in zip(xrange(3),"o^D","rgb"):

plt.scatter(X[Y == t,0], X[Y == t,1], marker=marker,c=col)

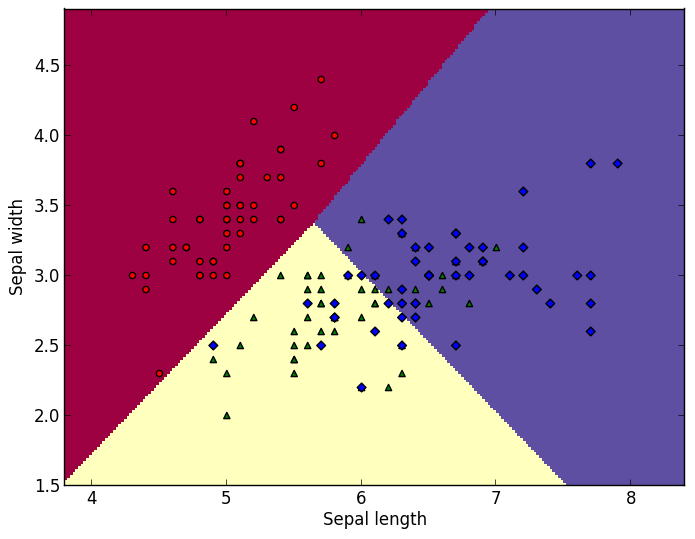
pl.xlabel('Sepal length')

pl.ylabel('Sepal width')

pl.xlim(xx.min(), xx.max())

pl.ylim(yy.min(), yy.max())

plt.savefig("E:/wulingfei/logistic\_classification/logistic\_classify4.png", bbox\_inches="tight")



**四、小组分工**

程序设计及编写：杜嘻嘻

程序调试：杜嘻嘻

实验报告：杜嘻嘻

**五、作业总结**

由于本科阶段几乎没有自己亲自动手编写过程序，虽然研究生开学以来一直在自学编程，但是刚入门感觉还是很艰难，所以要完成本次实验对我来说有点难度。通过本次实验，自己亲自完成一个实验的经历，我学习到了很多。以前都是拼命地学习理论知识，没有把理论运用到编程的实际操作中，这就导致自己的编程能力没有得到锻炼。我也认识到了自己的编程的道路很艰辛，但是我相信通过自己的努力，在不久的将来，我可以独挡一面的自己独立完成程序的编写。

通过学习模式识别这门课程，我可以毫不夸张地说赵老师真的是我见过的最认真负责的老师之一。每次上课，赵老师到教室的时间比大多数学生都早，然后就开始准备上课的教学内容，每节课都保质保量的教给我们很多知识。我们应该学习赵老师的这种精神，我相信只要拿出赵老师的这种对待学术的态度，我们还怕学不好编程？感谢赵老师的教学帮助，并且在实验过程中给了我们大家很多指导。