

**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。

还有类似的某用户购买某商品的可能性，某病人患有某种疾病的可能性等等。这个世界是随机的（当然了，人为的确定性系统除外，但也有可能有噪声或产生错误的结果，只是这个错误发生的可能性太小了，小到可以忽略不计而已），所以万物的发生都可以用可能性或者几率来表达。“几率”指的是某事物发生的可能性与不发生的可能性的比值。 Logistic regression可以用来回归，也可以用来分类，主要是二分类。

**二、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)

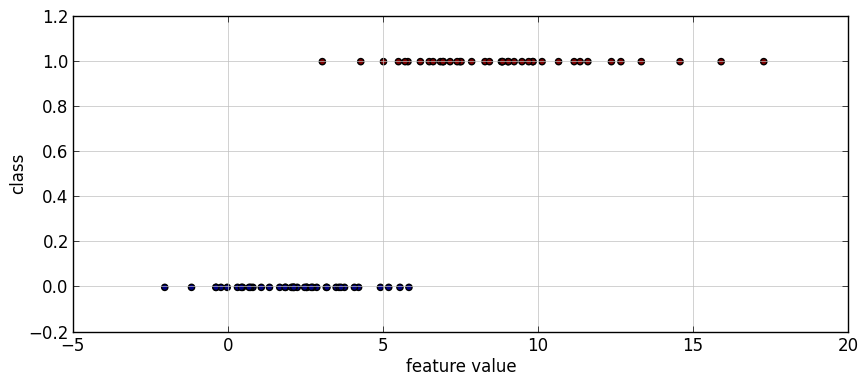
plt.xlabel("feature value")

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']

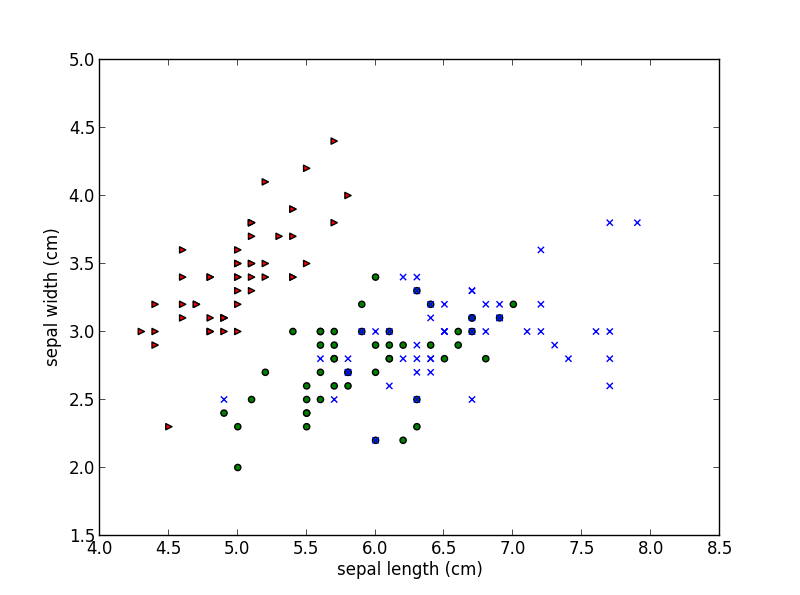
for t,marker,col in zip(xrange(3),">ox","rgb"):

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)

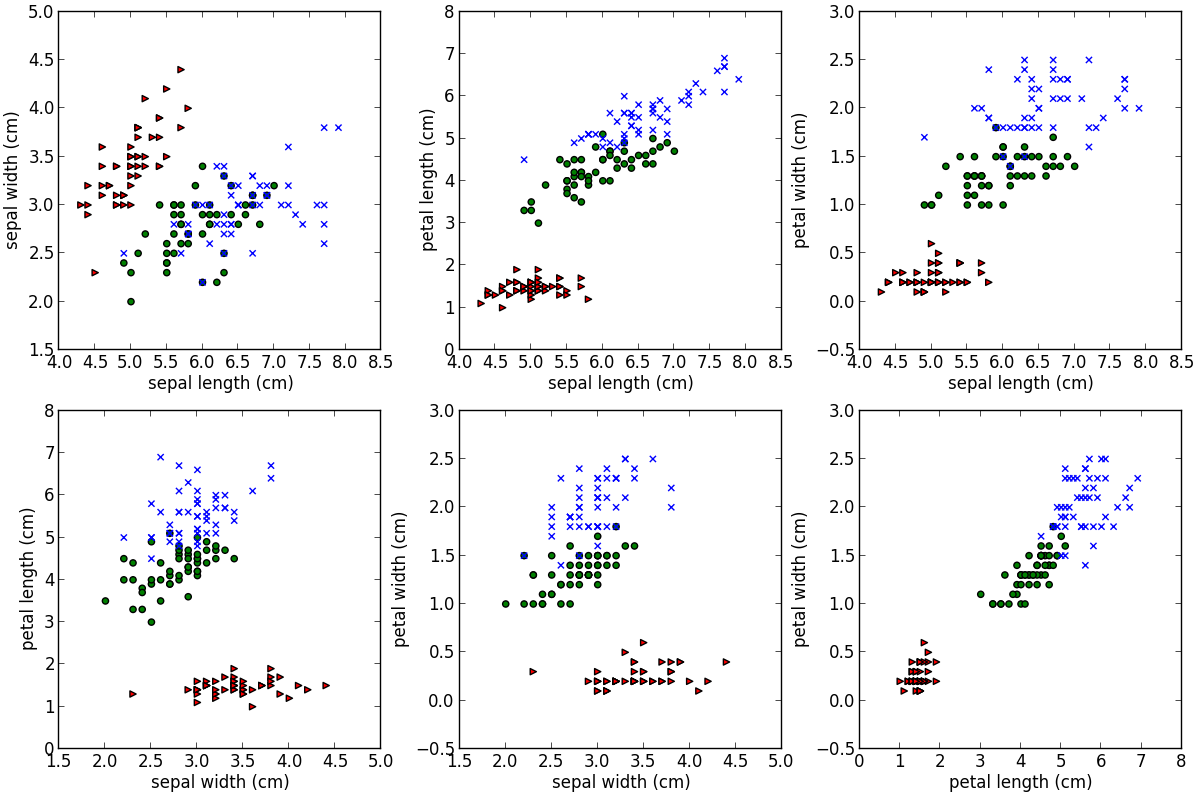
plt.subplot(235)

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)

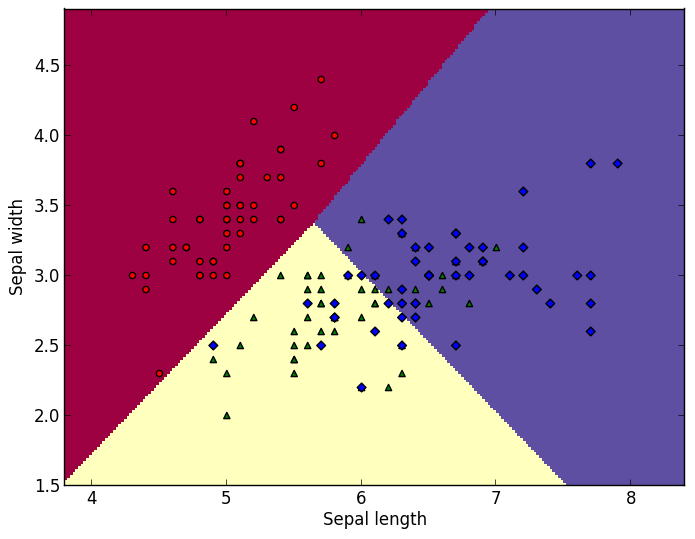
pl.xlabel('Sepal length')

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")



**三、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

**四、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。以我现在的水平也只能在现有的程序上改动一下。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开学以来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。