Python编程及人工智能应用

第四章 逻辑斯蒂分类及Python实现

内容提要

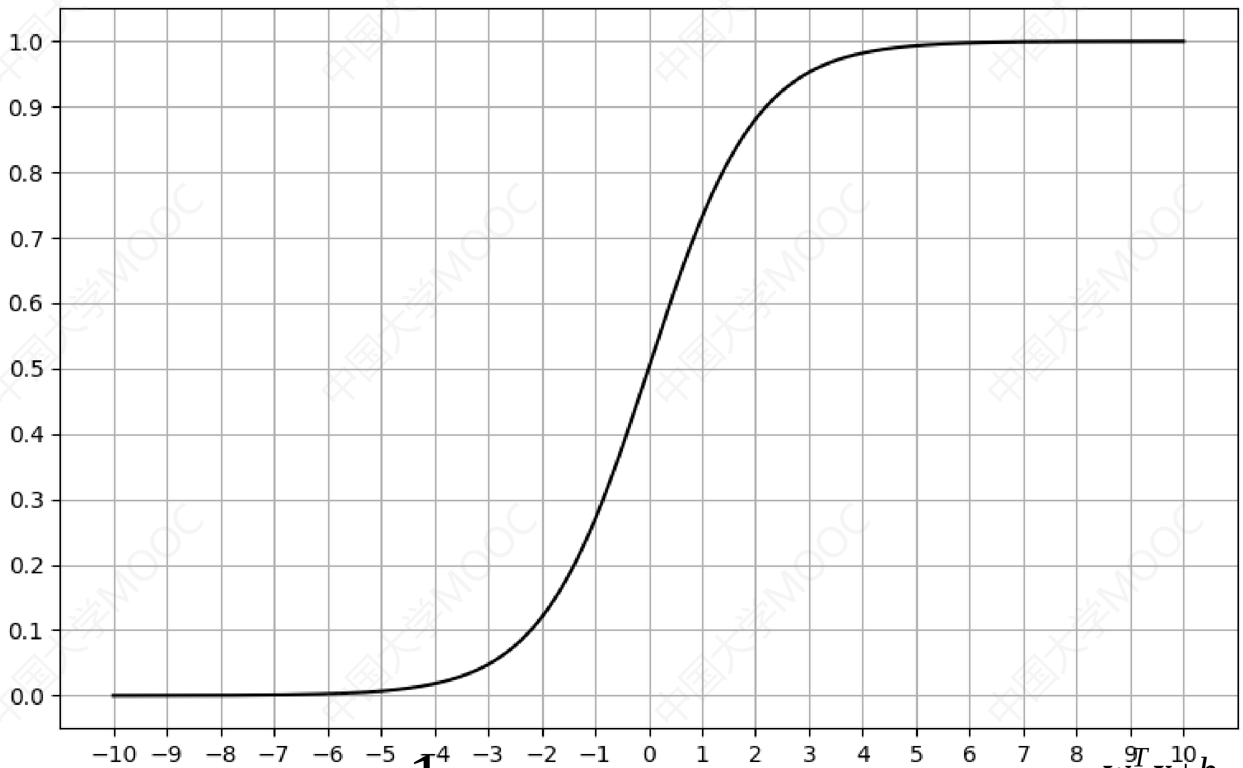
- 逻辑斯蒂分类简介
- 二分类逻辑斯蒂分类问题
- 基于Scikit-learn库求解二分类逻辑斯蒂分类
- 基于梯度下降法求解二分类逻辑斯蒂分类
- 分类模型的评价
- 非线性分类问题
- 正则化问题
- 多类别逻辑斯蒂分类问题

分类问题简介

- 分类问题的预测值是**离散**的
  - 根据晚风和晚霞预测明天**是否晴天**
  - 根据户型、面积、价格等因素预测房子**是否好卖**
  - 根据气色、打喷嚏、食欲等估算**是否感冒**
  - 根据西瓜的外观、敲瓜响声判断西瓜**是否甜**
  - 根据餐馆的飘香、入座情况等判断菜品**是否好吃**
- 分类对人类来说是一个基本能力
- 让人工智能学习分类是一个复杂的过程，需要**优秀的模型**、**海量的数据**和**高性能的硬件支持**

逻辑斯蒂分类简介

- 逻 …），
  是 …
- 逻 … 斯蒂
  函 … $\frac{1}{e^{-x}}$
- f(x …
- 对 …
- 给 …



$$P(y=1|\mathbf{x})=\frac{1}{1+e^{\mathbf{w}^T\mathbf{x}+b}} \qquad P(y=0|\mathbf{x})=\frac{e^{\mathbf{w}^T\mathbf{x}+b}}{1+e^{\mathbf{w}^T\mathbf{x}+b}}$$

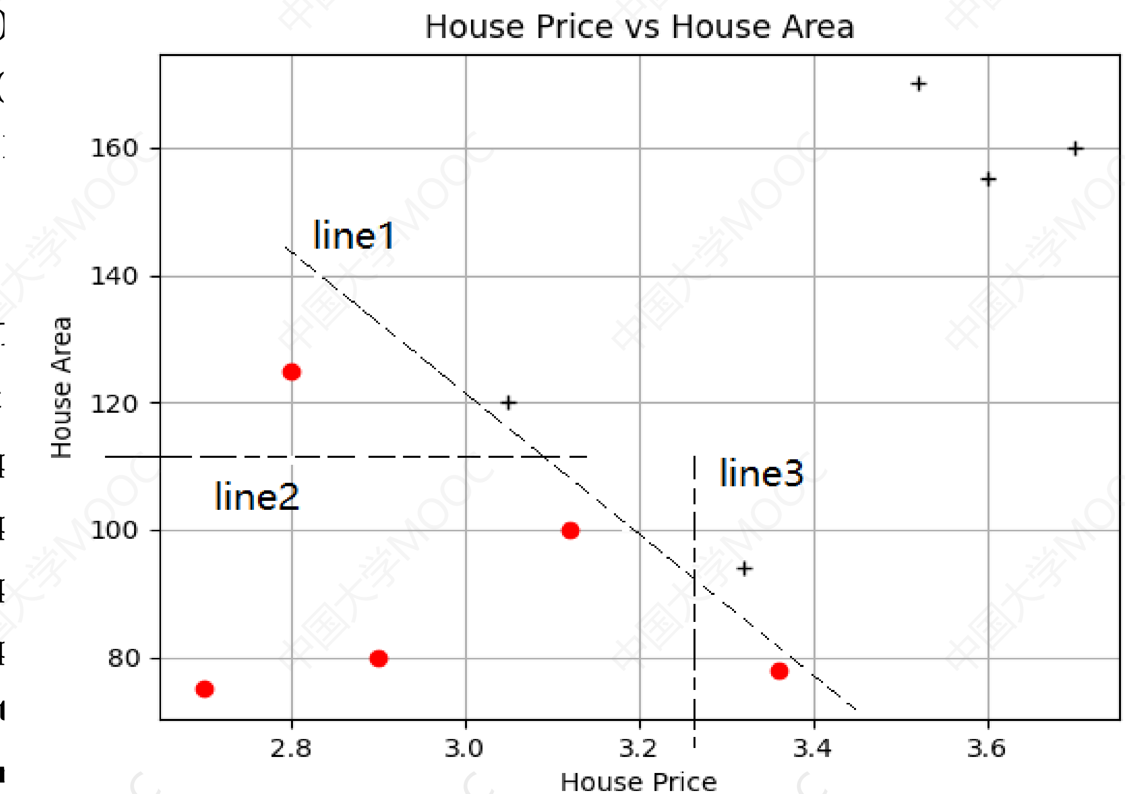
二分类逻辑斯蒂分类问题

- 当逻辑斯蒂分类类别数量只有两个时（即y的取值 … 型
- 案例
  - 根 … 挂售 半 … 售半 年 … 跟房 屋 … 系。 下 … 练样 本 … 售房 屋， … 据表中的训练数据，中介要判断该房屋是否好卖。

样本	房屋面积	房屋单价（万元/平米）	是否好卖
训练样本1	78	3.36	是
训练样本2	75	2.70	是
训练样本3	80	2.90	是
训练样本4	100	3.12	是
训练样本5	125	2.80	是
训练样本6	94	3.32	否
训练样本7	120	3.05	否
训练样本8	160	3.70	否
训练样本9	170	3.52	否
训练样本10	155	3.60	否
测试样本1	100	3.00	是
测试样本2	93	3.25	否
测试样本3	163	3.63	是
测试样本4	120	2.82	是
测试样本5	89	3.37	是

案例分析


```
#代码4.1 表4.1中房屋销售数据的可视化展示代码
import numpy as np
import matp
def initPlot()
    plt.figure(
    plt.title ( '
    plt.xlabel
    plt.ylabel
    plt.grid ( 
    return plt
xTrain0 = np                                            的样本
yTrain0 = np
xTrain1 = np                                            本
yTrain1 = np
plt = initPlot
plt.plot ( xTr
plt.plot ( xTrain1 [ :, 0 ], xTrain1 [ :, 1 ], 'ro' ) #r表示红色，o表示点的形状为圆形
plt.show ()
```



LogisiticRegression类

- **使用Scikit-learn库的LogisticRegression类解决逻辑斯蒂分类问题**

from sklearn.linear_model import LogisiticRegression

model=LogisticRegression(penalty='l2', dual=False, tol=0.0001, C=1.0, fit_intercept=True, intercept_scaling=1, class_weight=None, random_state=None, solver='liblinear', max_iter=100, multi_class='ovr', verbose=0, warm_start=False, n_jobs=1)

  - **penalty**：正则化参数，可选值为“L1”和“L2”
  - **solver**：优化算法选择参数
    - **liblinear**：使用坐标轴下降法来迭代优化损失函数
    - **lbfgs**：拟牛顿法的一种
    - **newton-cg**：也是牛顿法家族的一种
    - **sag**：随机平均梯度下降
  - **multi_class**：分类方式选择参数
  - **class_weight**：类别权重参数
  - **fit_intercept**：是否存在截距
  - **max_iter**：算法收敛的最大迭代次数
- **拟合函数fit（x, y）、预测函数predict（x）、评价分数值score（x, y）**

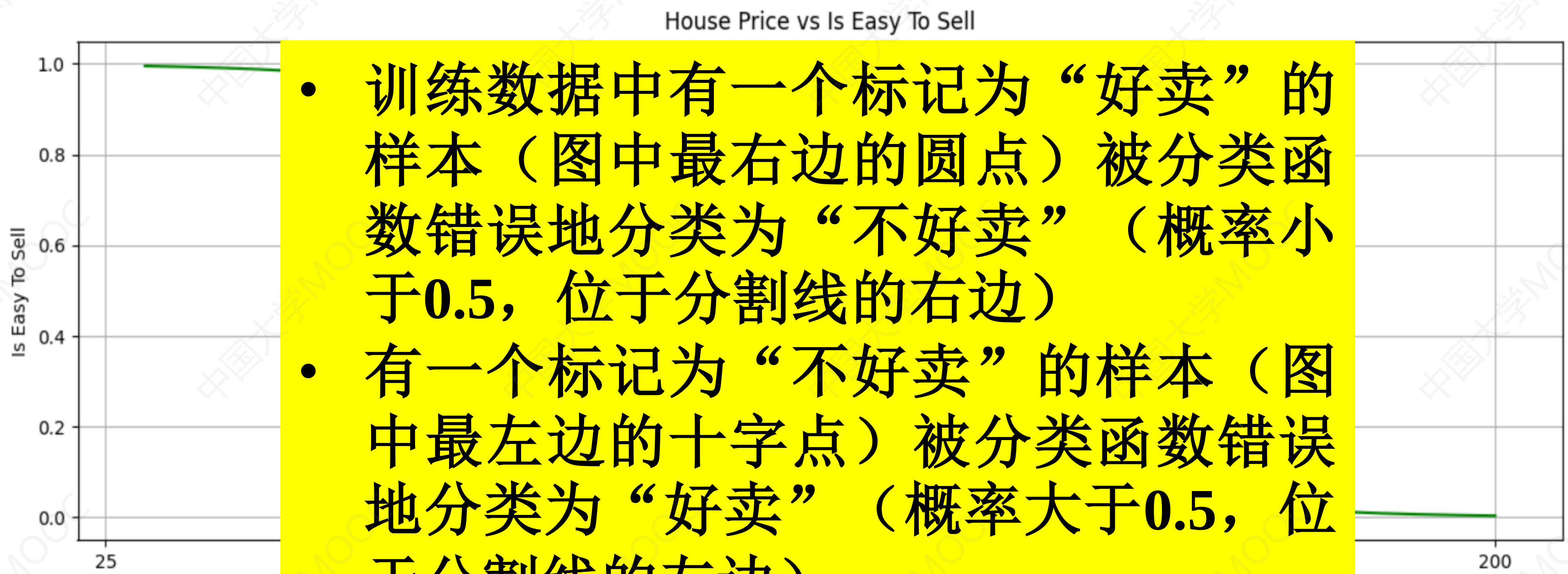
求解步骤

南京邮电大学
Nanjing University of Posts and Telecommunications

- 第一步：准备训练数据
  - xTrain = np.array ([[94], [120], [160], [170], [155], [78], [75], [80], [100], [125]])
  - yTrain = np.array ([0, 0, 0, 0, 0, 1, 1, 1, 1, 1])
- 第二步：创建LogisticRegression对象并拟合
  - from sklearn.linear_model import LogisticRegression #导入类
  - model = LogisticRegression (solver = "lbfgs") #创建对象，默认优化算法是L-BFGS
- 第三步：执行拟合
  - model.fit (xTrain, yTrain) #执行拟合
  - print (model.intercept_) #输出截距
  - print (model.coef_) #输出斜率
- 第四步：对新数据执行预测
  - newX = np.array ([[100], [130]]) #定义新样本
  - newY = print (model.predict (newX)) #输出斜率

编码实现

- 运行演示代码4.2



- 训练数据中有一个标记为“好卖”的样本（图中最右边的圆点）被分类函数错误地分类为“不好卖”（概率小于0.5，位于分割线的右边）
- 有一个标记为“不好卖”的样本（图中最左边的十字点）被分类函数错误地分类为“好卖”（概率大于0.5，位于分割线的左边）。

- 拟合得到函数：$P(y=1|x)=f(x)=\dfrac{1}{1+e^{-(0.06426704x+7.23982418)}}$

  - 当x=112.65时，分母中的指数部分为零 $P(y=1|x=112.65)=0.5$

梯度下降法优化目标

- **逻辑斯蒂分类的判别函数** $P(y=1|\mathbf{x})=f(\mathbf{x})=\frac{e^{\mathbf{w}^T\mathbf{x}}}{1+e^{\mathbf{w}^T\mathbf{x}}}=\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$
  - **其中：** $\mathbf{w}^T=[w_0,w_1,w_2,\ldots w_d]$ $\mathbf{x}^T=[1,x_1,x_2,\ldots x_d]$
- **训练数据中有*m*个样本，$y^{(i)}$=0表示第i个样本的实际类别为第0类，$y^{(i)}$=1表示该样本的实际类别为第1类。**
- **M_0为实际类别为0的样本子集，M_1为实际类别为1的样本子集**
  - **对于一个M_0中的样本i，其预测概率为** $P(y=0|\mathbf{x}^{(i)})=1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}$**，要尽量使得这个预测概率最大化，通常对这个函数取对数后进行优化**

$$\max imize\frac{1}{|M_0|}\sum_{i\in M_0}\log(1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})$$

  - **对于M_1中任一个样本i，其预测概率为** $P(y=1|\mathbf{x}^{(i)})=\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}$**，要尽量使得这个预测概率最大化，同样地，取对数后可得到**

$$\max imize\frac{1}{|M_1|}\sum_{i\in M_1}\log(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})$$

梯度下降法优化目标

- 将以上两类样本的优化目标合并之后，可以得到总的优化目标如公式

$$\max imize \frac{1}{m}\sum_{i=1}^{m} y^{(i)}\log\left(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}\right)+(1-y^{(i)})\log\left(1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}\right)$$

- 左半部分是用实际类别为1的训练样本进行优化，左半部分是用实际类别为0的训练样本进行优化
- 优化目标一般是进行最小化而不是最大化，**L(w)**也被称为损失函数（**Loss Function**）

$$L(\mathbf{w})=\frac{1}{m}\sum_{i=1}^{m} -y^{(i)}\log\left(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}\right)-(1-y^{(i)})\log\left(1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}\right)$$

$$\min imize \quad L(\mathbf{w})$$

梯度计算

- 梯度下降法需要根据梯度更新参数，更新公式如下

$$w_j = w_j - \alpha * \frac{\partial L(\mathbf{w})}{\partial w_j}$$

- 偏导数的求解如下（演示推导过程）

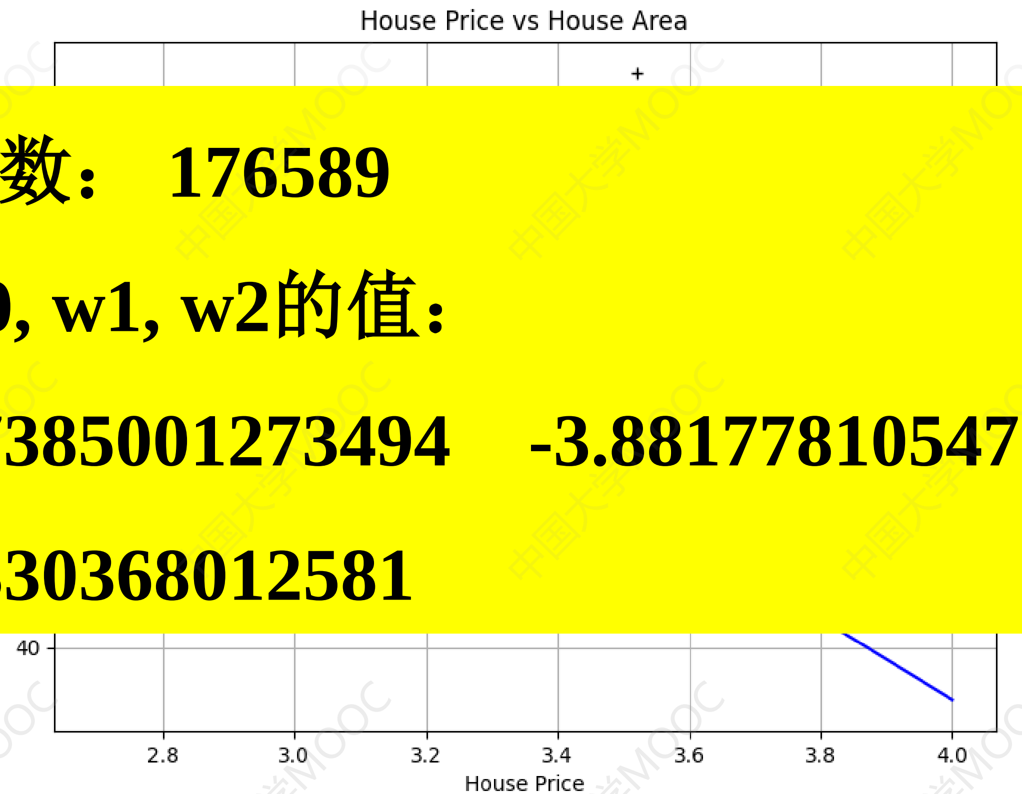
$$f(\mathbf{x}^{(i)}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}$$

$$\frac{\partial L(\mathbf{w})}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m} -y^{(i)} \frac{1}{f(\mathbf{x}(i))} \bullet \frac{\partial f(\mathbf{x}^{(i)})}{\partial \mathbf{w}_j} - (1-y^{(i)}) \frac{1}{1-f(\mathbf{x}^{(i)})} \bullet \frac{-\partial f(\mathbf{x}^{(i)})}{\partial w_j}$$

$$\frac{\partial L(\mathbf{w})}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m} (-y^{(i)} + \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}}) \bullet x_j^{(i)}$$

Python编码实现

- **演示运行代码4.3：基于梯度下降求解房价预测问题的Python实现**



迭代次数： 176589

参数w0, w1, w2的值：

-1.9407385001273494 -3.8817781054764713

-4.408330368012581

输出结果的说明

- 该代码采用批量梯度下降法相同的实现，即bgd_optimizer函数。该函数需要传入成本函数（目标函数）、梯度函数、参数初始值、学习率等通用参数。具体到逻辑斯蒂分类问题，其成本函数和梯度函数已经在前面定义并用Python实现，作为参数传入bgd_optimizer函数即可。
- 由于“房屋面积”与“房屋单价”这两个属性具有不同取值范围，取值范围差异大，在样本数据传入梯度下降函数进行训练之前先进行归一化操作

$$x\_norm_i = \frac{x_i - \overline{x_i}}{std(x_i)}$$

- 训练数据的两个属性分别对应优化参数w1和w2，由于参数向量也包含w0，而w0与1对应，因此为了便于向量运算，在训练数据属性向量中增加一个值为1的量，对应代码中的make_ext()函数。
- 根据输出文字结果，梯度下降法总共迭代了176589次，得到的w0、w1、w2三个参数值约为-1.94、-3.88、-4.41，得到的逻辑斯蒂分类函数为

$$f(x) = \frac{1}{1+e^{-(-1.94-3.88x_1-4.41x_2)}}$$

分类模型的评价方法

南京邮电大学 Nanjing University of Posts and Telecommunications

- 对于一些疾病（如癌症、艾滋等），人群中只有约0.5%的人患有这种疾病。因此，完全可以设计一个极端的方法：对于任何样本，永远预测y=0，即该样本没有该疾病。这样的简单方法只有0.5%的错误率，根据这种错误率是否能判定这样的极简模型是好模型呢？
- 简单地使用错误率或正确率（Accuracy）来判定模型好坏不一定是一种合适的做法。

分类模型的评价方法

- **正确率（Accuracy）：** $$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$
- **精准率（Precision）：** $$Precision = \frac{TP}{TP + FP}$$
- **召回率（Recall）：** $$Recall = \frac{TP}{TP + FN}$$
- **F1指数（F1-Score）：** $$F1Score = \frac{2PR}{P + R}$$

设定测试样本总数为1000个，其中有5个样本为真实确诊病例（y=1），如果根据极端分类方法将所有样本预测为y=0，则TP=0，TN=995，FP=0，FN=5

		真实类别（Actual Class）	
		1	0
预测类别（Predicted Class）	1	真阳（True Positive）	假阳（False Positive）
	0	假阴（False Negative）	真阴（True Negative）

代码4.4


```
# 代码4.4 手动计算房屋好卖预测的各种评价指标
# 该代码不能独立执行，请粘贴到代码4.3的plt.show()语句之前再运行
xTest = np.array ( [ [ 3.00, 100 ] , [ 3.25, 93 ], [ 3.63, 163 ], [ 2.82, 120 ], [ 3.37, 89 ] ] )
xTest_norm = normalize ( xTest, mean, std )
xTest
yTest
yTest
yTest
yTest
accur
precis                                                                 隹确率
recall
f1scor
print
print ( "实际值： ", yTest )
print ( "正确率（Accuracy）： ", accuracy )
print ( "准确率（Pecision）： ", precision )
print ( "召回率（Recall）： ", recall )
print ( "F1Score： ", f1score )
```

预测值： [True True False True False]

实际值： [1 0 1 1 1]

正确率（Accuracy）： 0.4

准确率（Pecision）： 0.6666666666666666

召回率（Recall）： 0.5

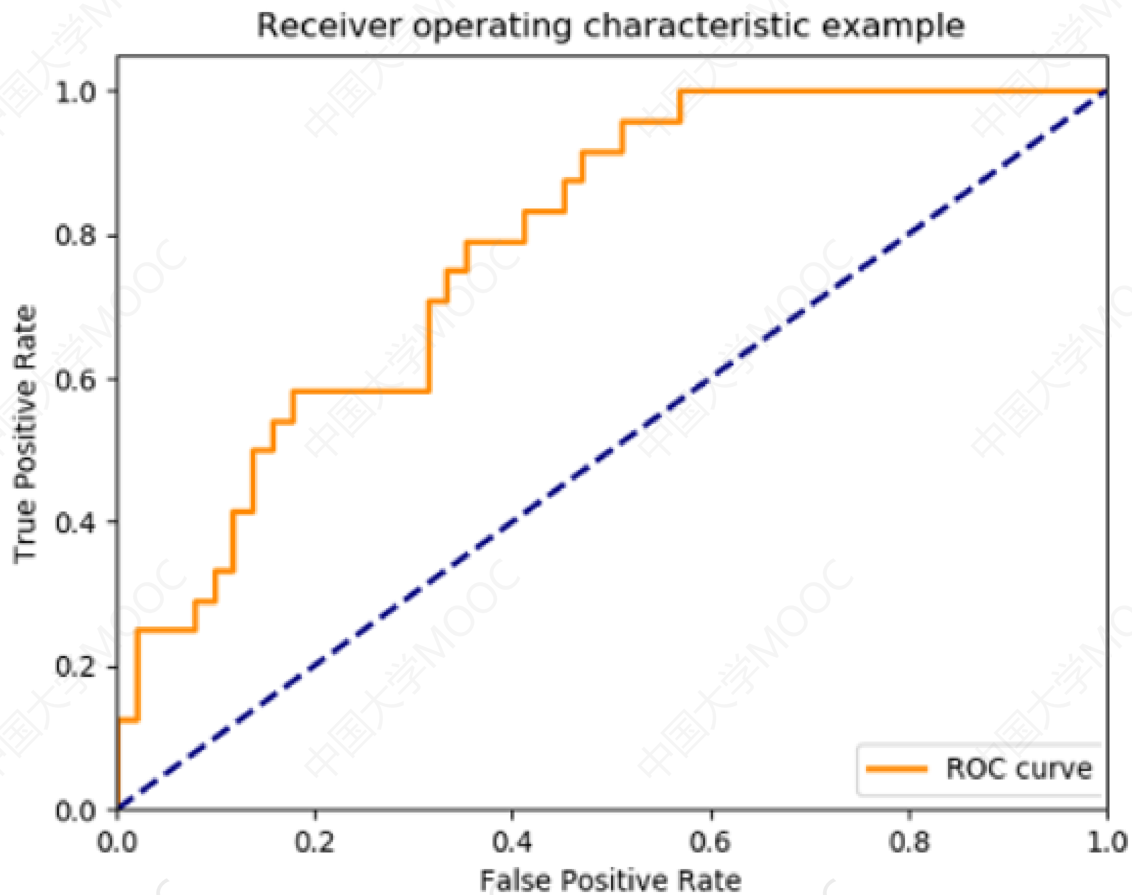
F1Score： 0.5714285714285715

运行结果说明

- 测试数据集中共有5个样本，并给出真实y值用于评价；计算预测y值前，先将属性值归一化，再根据分类模型预测y值。
- 语句"yTest_real_pred = yTestPredicted == yTest"是将预测值与真实值逐项比较，相同则对应的项为1，不相同则对应的项为0，注意正例和负例都进行了比较，此时预测值为[1, 1, 0, 1, 0]，真实值为[1, 0, 1, 1, 1]
- 逻辑斯蒂分类模型计算出来的是概率值，判定时需设定一个阈值K，当概率值大于等于K时判定为正例，小于K时判定为负例
- 思考：2020年新冠疫情爆发以来，我国成为全球防疫最成功的国家，但是受国外输入疫情影响，局部偶有暴发，比如2021年7月份南京禄口机场的输入疫情，为此南京全城加强排查，各个小区进行多次核酸检测筛选。请分析，在核酸检测筛选问题中，更应看重正确率、准确率和召回率中的哪些指标？

ROC曲线

- 接受者操作特性曲线（receiver operating characteristic c urve，
  sitivity cu rve），
- ROC曲
  FPR）为纵 轴所组
  时候其真阳 性率和
- 理想的



代码4.5

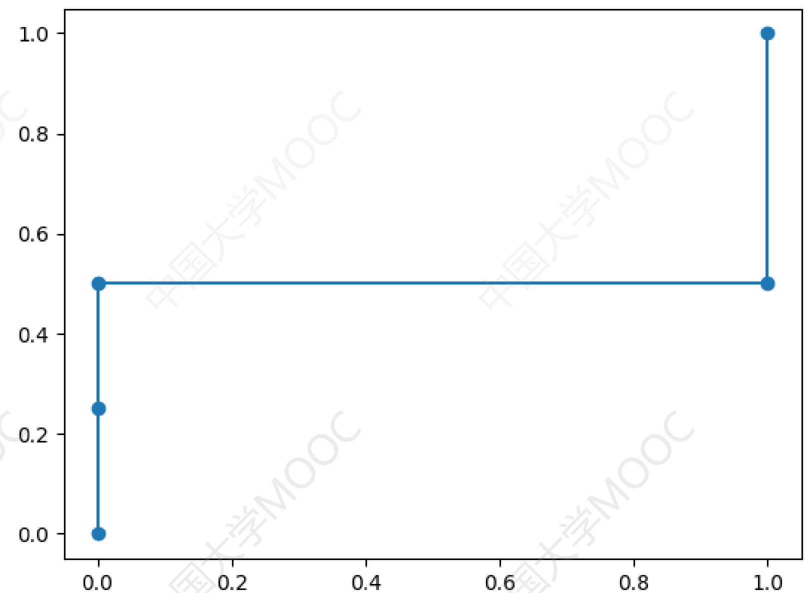
南京邮电大学
Nanjing University of Posts and Telecommunications

K值： [1.92641428e+00 9.26414276e-01 8.88241401e-01 6.19132228e-01 2.15106810e-06]

fpr： [0. 0. 0. 1. 1.]

tpr： [0. 0.25 0.5 0.5 1.]

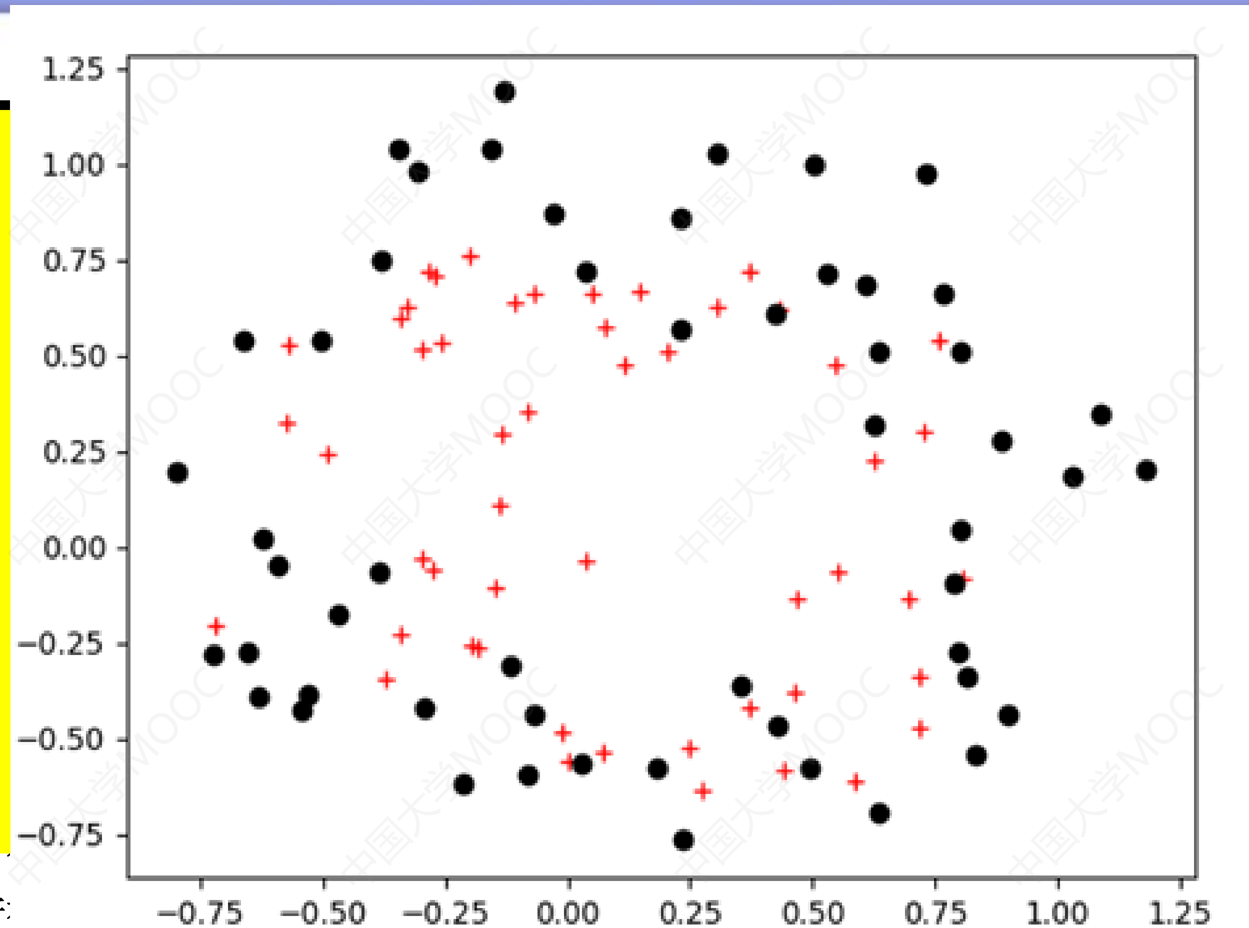
```
# 代码
# 该
xTest
xTest
xTest
yTest
yTest
from sklearn import metrics
fpr, tpr, thresholds = metrics.
print ( "K值： ", thresholds )
print ( "fpr： ", fpr )
print ( "tpr： ", tpr )
plt.scatter ( fpr, tpr )
plt.plot ( fpr, tpr )
```



非线性分类问题

- 在很多情况下，

无法找到一条可以将这些正负样本较好分开的直线，这种情况称为“线性不可分”，但从图中可以看出应该可以找到一条曲线将这些样本分隔开




```
X01 = X [ X [ :, 2 ] == 0, 0 ] #获取所
X02 = X [ X [ :, 2 ] == 0, 1 ] #获取所
plt.plot ( X11, X12, "r+" ) #绘制类别为1的样本点
plt.plot ( X01, X02, "ko" ) #绘制类别为0的样本点
plt.show ( )
```

基于Scikit-learn库求解

- 上述数据必须采用高阶曲线才可能进行分割
- 可尝试采用如下逻辑斯蒂分类函数

$$f(\mathbf{x}) = \frac{1}{1+e^{g_w(\mathbf{x})}}$$

- 其中

$$\begin{aligned} g_w(\mathbf{x}) &= w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_1 x_2 + w_5 x_2^2 + w_6 x_1^3 + w_7 x_1^2 x_2 + w_8 x_1 x_2^2 + w_9 x_2^3 + \cdots \\ &+ w_{21} x_1^6 + w_{22} x_1^5 x_2^1 + w_{23} x_1^4 x_2^2 + w_{24} x_1^3 x_2^3 + w_{25} x_1^2 x_2^4 + w_{26} x_1 x_2^5 + w_{27} x_2^6 \end{aligned}$$

- x_1和x_2是样本的两个属性，进行多种乘方组合变化，可以得到共28组不同的特征

代码4.7和代码4.8

南京邮电大学 Nanjing University of Posts and Telecommunications

输出：

model.coef_： [[6.75648906 15.61204317 42.60187485 -61.49758007

-35.68695213 -89.18474107 -13.76242629 -193.87613648

-301.02643472 -244.9038965 96.59149095 132.7132875

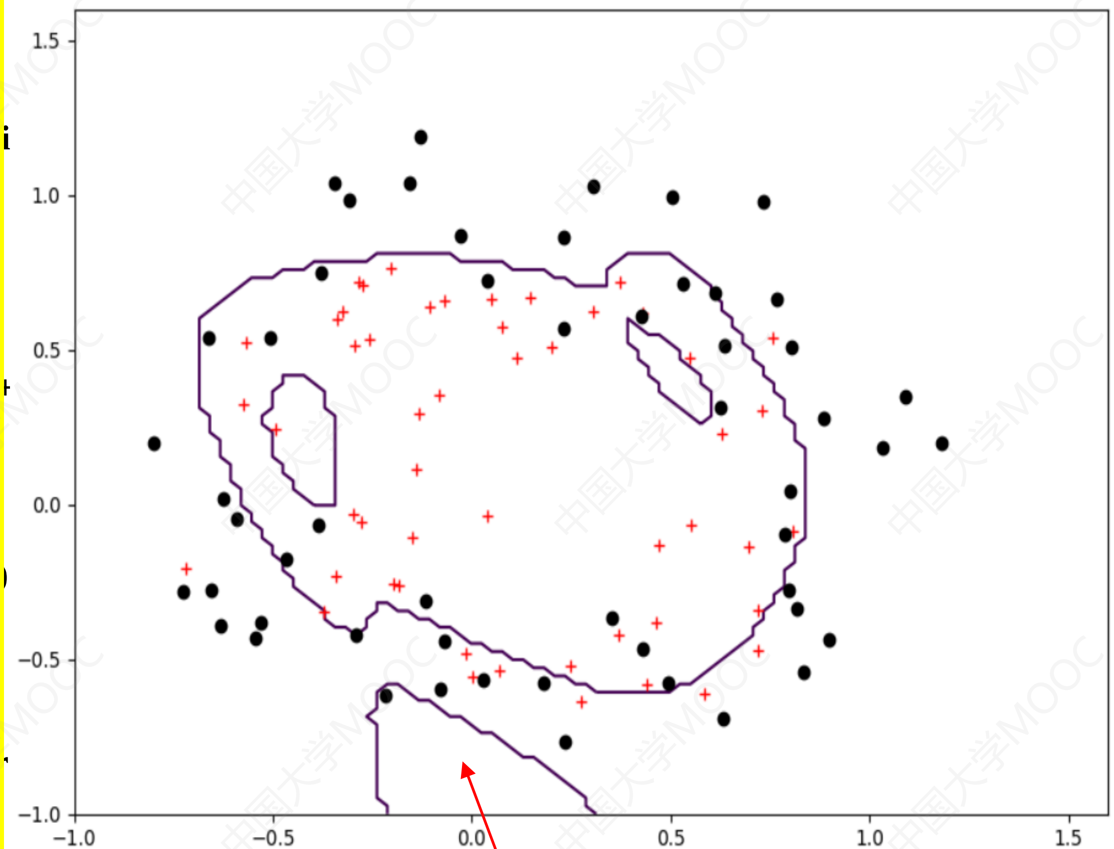
425.84753516 254.50022587 307.12501577 -23.98634476

146.08263767 504.59891556 823.70400268 772.78806942

386.27226822 -46.62328301 -74.27162823 -517.93456299

-778.26775291 -1226.28898173 -729.0797253 -444.26963462]]

model.intercept_： [6.75648906]



“过拟合” 问题

```
, penalty = "n
ay ( [ newX2 [ j ] ] ) ) #扩展特征
) #计算类别
plt.contour ( newX1, newX2, Z, levels = [ 0 ] )
plt.show ( )
print ( "model.coef_: ", model.coef_ )
print ( "model.intercept_: ", model.intercept_ )
```

正则化问题

- 为了防止过拟合问题，正则化方法被提出来了
- 核心思想是：由于特征数量维度多而造成权重参数也很多，应尽可能使每个权重参数的值小，以降低模型复杂度和不稳定程度，从而避免过拟合的危险。
- 采用的方法是在成本函数中加入一个惩罚项。

$$L(\mathbf{w})=\frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})-(1-y^{(i)})\log(1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})]+\frac{\lambda}{2m}\sum_{j=1}^{d}w_j^2$$

正则化项说明

$$L(\mathbf{w})=\frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}\log(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})-(1-y^{(i)})\log(1-\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})]+\frac{\lambda}{2m}\sum_{j=1}^{d}w_j^2$$

- $\frac{\lambda}{2m}\sum_{j=1}^{d}w_j^2$**是惩罚项，也被称为penalty项，是对参数数量和大小的约束**
- **惩罚项是有范式的，以上是二范式项；也可以根据需要替换成零范式（要么有参数*w_j*，要么没有参数*w_j*）、一范式、三范式**
- λ **是超参数，用于调整惩罚项的权重**

正则化项说明

- 惩罚项的梯度是 $\dfrac{\partial Penalty}{\partial w_j} = \dfrac{\lambda}{m} w_j$
- 得到成本函数的梯度如下

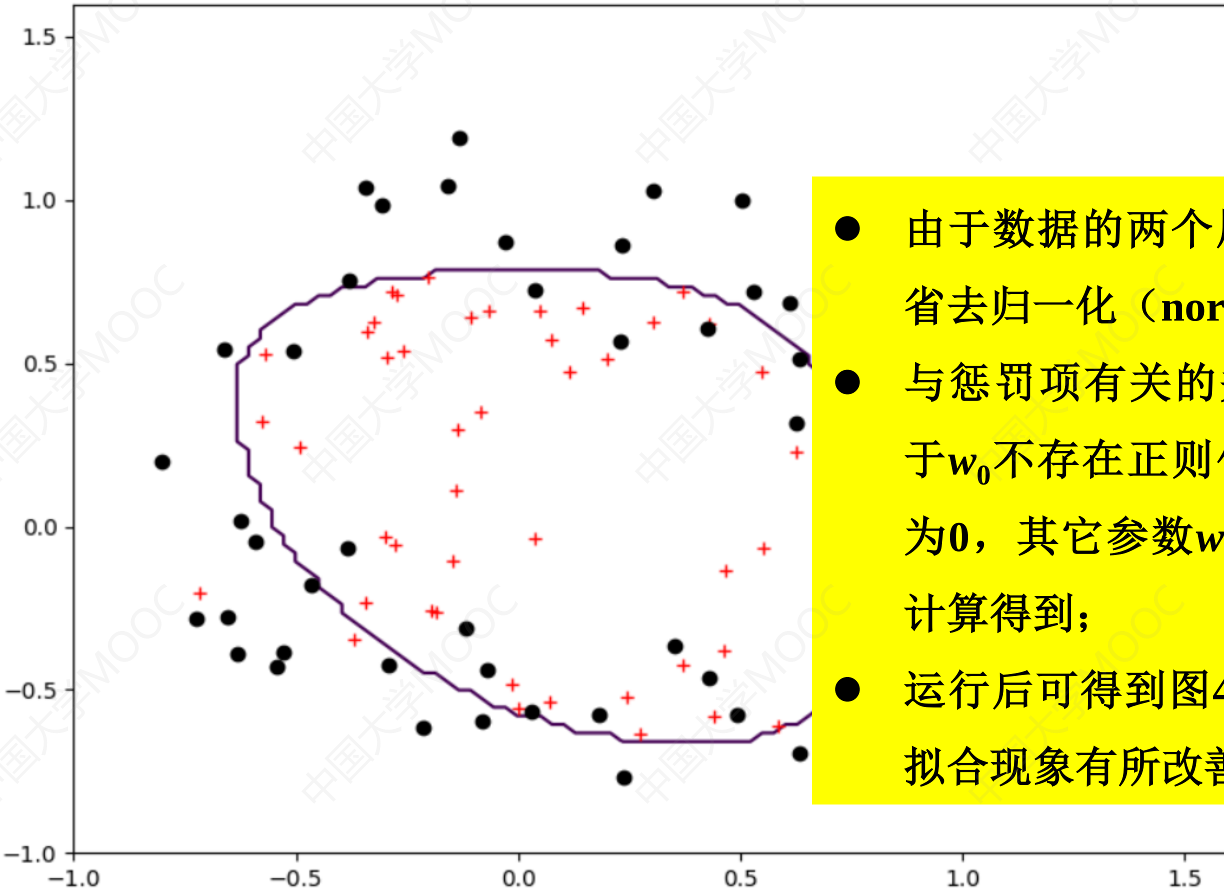
$$\frac{\partial L(\mathbf{w})}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m}[(-y^{(i)} + \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}^{(i)}}})\bullet x_j^{(i)}] + \frac{\lambda}{m} w_j$$

- 该式适用于除 w_0 以外的参数 w_j 的更新

正则化问题的求解实现

南京邮电大学 Nanjing University of Posts and Telecommunications

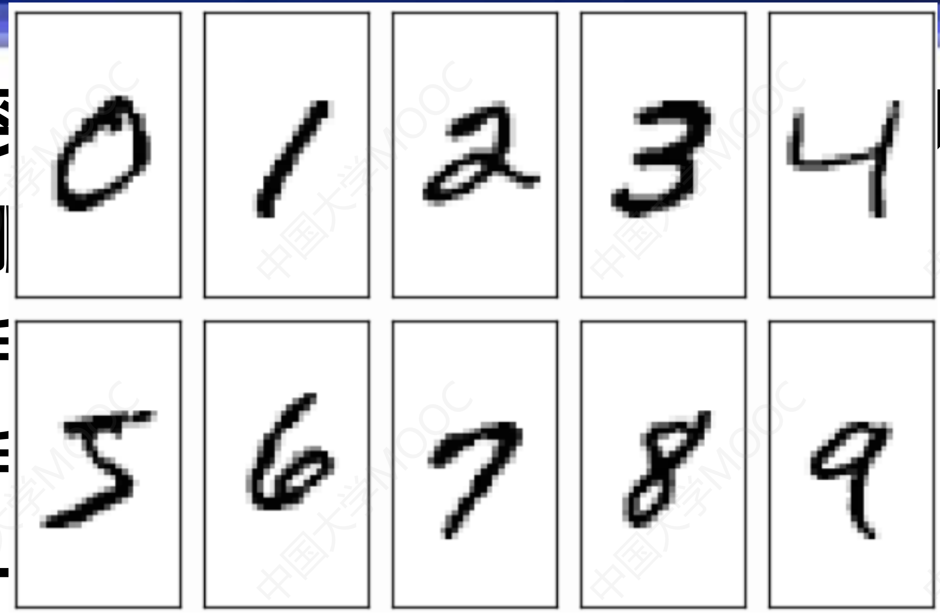
语句修改成如下



- 由于数据的两个属性值取值范围相同，因此可以省去归一化（**normalize**）部分；
- 与惩罚项有关的关键部分在**grad_fun**函数中，由于w_0不存在正则化项，因此w_0对应的惩罚项梯度为**0**，其它参数w对应的惩罚项梯度根据公式**4.12**计算得到；
- 运行后可得到图**4.8**所示的图，其中分届曲线的过拟合现象有所改善。

多类别逻辑斯蒂分类

- 现实真 … 以上的
  - 判断哪 …
  - 判断手 … 哪一个
  - 判断手 …
  - 判断一 …



- 手写阿拉伯数字识别
  - 本质上是一个有10个类别的多分类问题
  - 对大量已知图片数据进行训练，得出相应的逻辑斯蒂分类模型参数

求解步骤

- 第一步：准备训练数据和测试数据
  - 训练数据包含5000张手写数字图片
  - 测试数据包含500张手写数字图片
  - 每张图片标注好对应的正确数字，称为标签
  - 每张图片分辨率统一为28*28
  - 每张图片都是灰度图，即像素值取值为0~255，0表示纯白，255表示纯黑，中间值为黑度不同的灰色，省去了处理RGB彩色图片的负担
  - 数据在arab_digits_for_training.txt和arab_digits_for_testing.txt文件

求解步骤

- 第二步：计算特征矩阵用于模型的训练
  - 将每个像素当做一个特征，每张图对应一维数组，将图片的每个像素点值依次连续放在数组中，共得到28*28个数组元素；图片的标签放在数组最后；最终一维数组的大小是28*28+1=785
  - 5000个训练图片对应一个二维数组，大小是5000*785，构成一个训练矩阵
  - 因为像素值取值范围是0~255，因此对训练数据进行归一化处理（normalization）
- 第三步：训练逻辑斯蒂分类模型并评估

基于Scikit-learn库实现

南京邮电大学 Nanjing University of Posts and Telecommunications

```
#代码4.10 使用LogisicRegession实现多类别分类
import numpy as np
import matplotlib.pyplot as
from sklearn.linear_model
def batch_normalize ( X, co
    return ( X - col_means ) /
#加载训练数据
trains = np.loadtxt ( "arab
t" )
trainX = trains [ :, 1: ] #第0
trainY = trains [ :, 0 ]
col_means = np.mean ( trainX, 0 ) #每个属性的平均值
trainX = batch_normalize ( trainX, col_means )
model = LogisticRegression ( solver = "lbfgs", multi_class = "
multinomial", max_iter = 500 )
model.fit ( trainX, trainY )
```

```
#加载测试数据
g.txt", delimiter = "\t
eans )
dictY )
errors, np.shape ( test
```

预测错误数是：54/500

代码中的归一化函数batch_normalize( )，其参数是一个二维numpy数组，对其中的每个列（属性）都要进行归一化；其目的是为了防止计算结果经过层层乘法运算后得到的数值过大而导致成本函数溢出

原则上采用前面章节的归一化公式 $x_i = \dfrac{x_i - \bar{x}_i}{std(x_i)}$

基于梯度下降法求解

南京邮电大学 Nanjing University of Posts and Telecommunications

- 由于原始的逻辑斯蒂回归解决的是二分类的问题，为了解决手写数字识别的多分类问题，可以把多分类问题转化为二分类问题求解
  - 为每个类别计算一个双类别逻辑斯蒂分类模型，其中该类别为正，其它所有类别为负，得到10个具有不同参数的双类别逻辑斯蒂模型
  - 在预测时，使用这10个模型对每个类别计算得到不同的预测概率，取其中概率最大的类别为预测类别
- 演示代码4.11
  - 预测错误数是：53/500

本章小结

- 本章介绍了逻辑斯蒂分类的定义，包括双类别逻辑斯蒂分类和多类别逻辑斯蒂分类；
- 介绍了分类评价的指标，包括正确率、准确率、召回率、ROC曲线等；求解逻辑斯蒂分类问题的多种方法，包括sklearn库函数求解法、梯度下降法等。
- 通过本章的学习，读者对逻辑斯蒂分类问题有了一个基本的了解，并掌握基于Python编码求解分类问题的基本方法。

输入理想的程序

输出快乐的人生