# 1. 逻辑回归思想

逻辑回归用sigmod函数：

训练集

对于二分类模型

“正类”为：

“负类”为：

# 2. 用极大似然法估计对多分类对数做几率回归

可以得到极大似然函数：

要选择合适的参数使输入参数和输出关系更为紧密，即使得最大，

将这两类模型统一起来：

其中

令，算符为，将极大似然函数代入可得，

分离分母，

又已知，该问题等价于极小化下式，

对于二分类，目标函数为

等价于最小化

# 3. 二分类算法思路：

1. 输入训练集

2. 初始化，令

3. 求解参数

1）牛顿法：

计算：

迭代，得到。

2）梯度下降法：

等价于最小化：

迭代，得到。

4. 用测试样本集去通过概率大小分类，计算正确率。

# 4. 代码实现：

选用包含LogisticRegression函数的sklearn进行逻辑回归分类

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression,SGDRegressor,Ridge,LogisticRegression

from sklearn.metrics import mean\_squared\_error, classification\_report

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import mean\_squared\_error

import pandas as pd

import numpy as np

def logistic():

"""

逻辑回归做二分类进行乳腺癌预测，用数据集的0.7作为训练集，用数据集的0.3作为测试集。Pytorch进行二分类要分每个维度分别进行，这里有sklearn内置逻辑回归函数对数据进行分类。首先读取在线数据集，然后进行数据分割，按0.7/0.3分割为训练集和测试集。

"""

# 构造列标签名字

column = ['Sample code number', 'Clump Thickness', 'Uniformity of Cell Size', 'Uniformity of Cell Shape',

'Marginal Adhesion', 'Single Epithelial Cell Size', 'Bare Nuclei', 'Bland Chromatin', 'Normal Nucleoli',

'Mitoses', 'Class']

# 读取在线数据集

data = pd.read\_csv(

"https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data",

names=column)

# print(data)

# 对存在缺失值进行处理，替换为nan

data = data.replace(to\_replace='?', value=np.nan)

data = data.dropna()

# 输出data的数据量和维度。

print("数据量，维度：\n", data.shape)

print("数据集：\n", data)

# 进行数据的分割，用数据集的0.7作为训练集，用数据集的0.3作为测试集

x\_train, x\_test, y\_train, y\_test = train\_test\_split(data[column[1:10]], data[column[10]], test\_size=0.3)

# 输出训练样本两个类分别的样本量。

print("训练样本类，数据量：\n", y\_train.value\_counts())

# 输出测试样本两个类分别的样本量。

print("测试样本类，数据量：\n", y\_test.value\_counts())

# 进行标准化处理

std = StandardScaler()

x\_train = std.fit\_transform(x\_train)

x\_test = std.transform(x\_test)

# 逻辑回归来对训练集进行训练

lg = LogisticRegression(C=1.0)

# 求得训练集X的均值、方差、最大值、最小值等固有属性

lg.fit(x\_train, y\_train)

print('回归系数\n', lg.coef\_)

# 训练后返回预测结果

y\_predict = lg.predict(x\_test)

# 输出预测结果计算出的决定系数R^2

print("拟合优度：", lg.score(x\_test, y\_test))

# classification\_report函数用于显示主要分类指标．显示每个类的精确度，召回率，F1值等

print("分类指标：", classification\_report(y\_test, y\_predict, labels=[2, 4], target\_names=["良性", "恶性"]))

if \_\_name\_\_ == "\_\_main\_\_":

logistic()

# 5. 输出结果：



