



机器学习

第4章 逻辑回归

朱桂祥 (9120201070@nufe.edu.cn)

南京财经大学信息工程学院

江苏省电子商务重点实验室

电子商务信息处理国家级国际联合研究中心

电子商务交易技术国家地方联合工程实验室

<https://github.com/zgx881205/Machine-Learning/tree/main>



南京财经大学

NANJING UNIVERSITY OF FINANCE & ECONOMICS

1.分类问题

2

01 分类问题

02 Sigmoid函数

03 逻辑回归求解

04 逻辑回归代码实现

分类问题

3

监督学习的最主要类型

标签离散

✓ 分类 (Classification)

- ✓ 身高1.85m, 体重100kg的男人穿什么尺码的T恤?
- ✓ 根据肿瘤的体积、患者的年龄来判断良性或恶性?
- ✓ 根据用户的年龄、职业、存款数量来判断信用卡是否会违约?

输入变量可以是离散的, 也可以是连续的。

分类问题

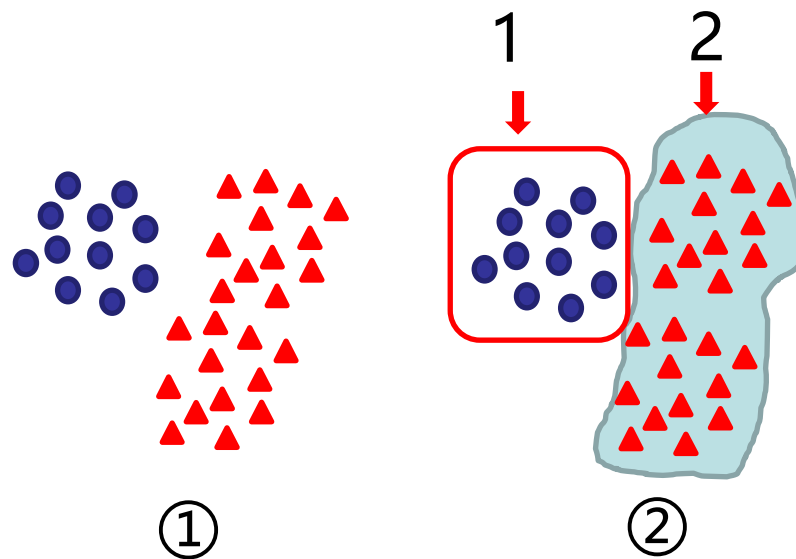
4

二分类

我们先从用蓝色圆形数据定义为类型1，其余数据为类型2；

只需要分类1次

步骤：①->②



二分类

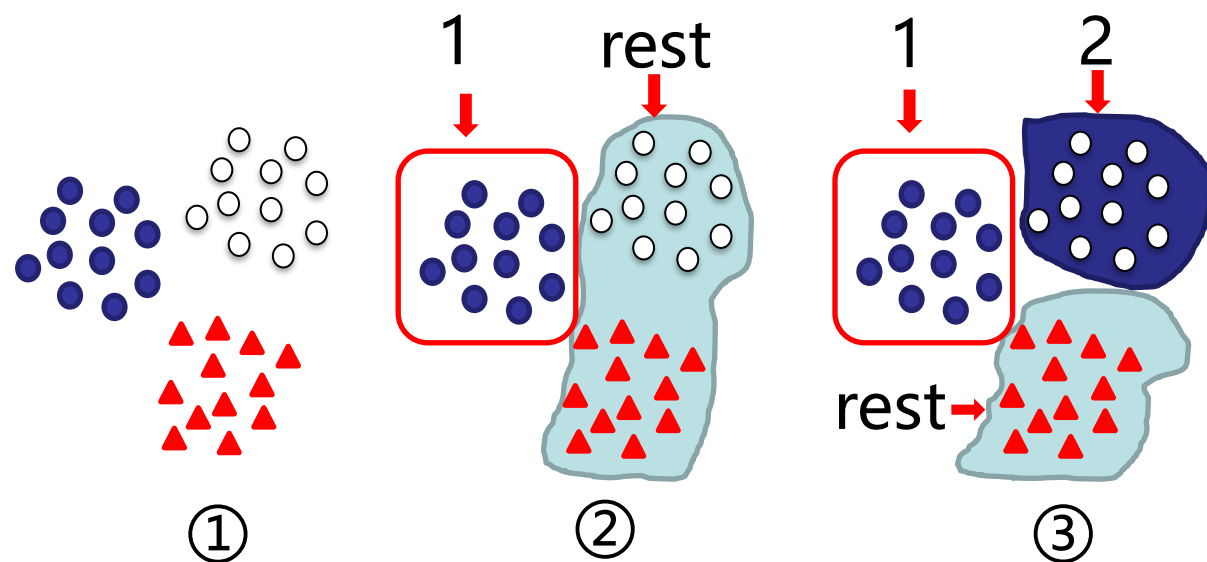
分类问题

5

多分类

我们先定义其中一类为类型1（正类），其余数据为负类（rest）；接下来去掉类型1数据，剩余部分再次进行二分类，分成类型2和负类；如果有 n 类，那就需要分类 $n-1$ 次

步骤：①->②->③->.....



One-vs-All (One-vs-Rest)

一对多 (一对余)

2.Sigmoid函数

6

01 分类问题

02 Sigmoid函数

03 逻辑回归求解

04 逻辑回归代码实现

2.Sigmoid函数

7

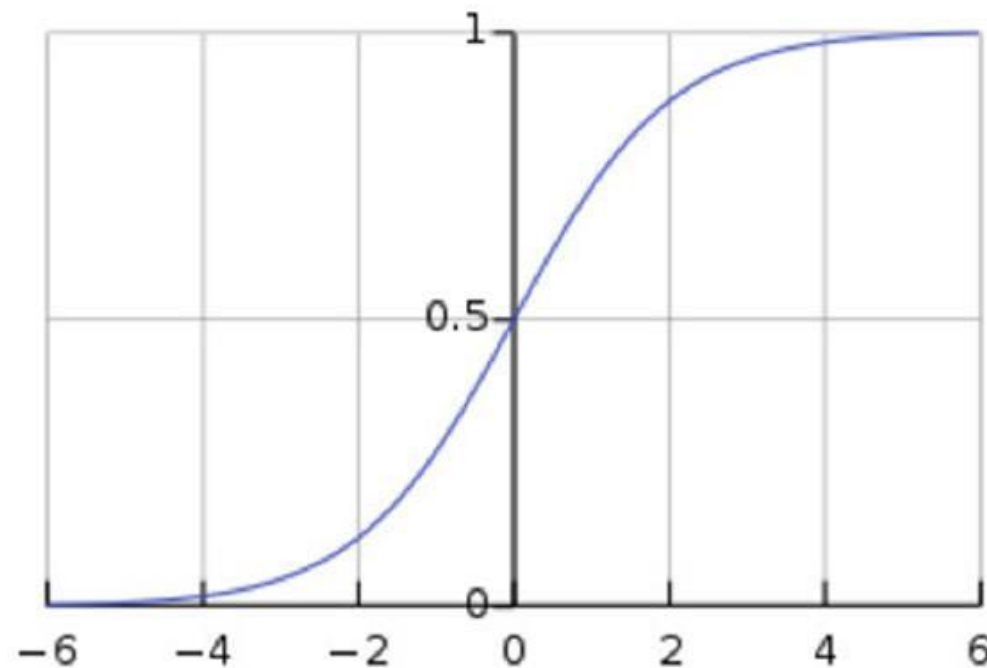
Sigmoid 函数

$\sigma(z)$ 代表一个常用的逻辑函数 (logistic function) 为S形函数 (Sigmoid function)

$$\text{则: } \sigma(z) = g(z) = \frac{1}{1+e^{-z}} \quad z=w^T x + b$$

合起来, 我们得到逻辑回归模型的假设函数:

$$L(\hat{y}, y) = -y\log(\hat{y}) - (1 - y)\log(1 - \hat{y})$$



当 $\sigma(z)$ 大于等于0.5时, 预测 $y=1$

当 $\sigma(z)$ 小于0.5时, 预测 $y=0$

注意: 若表达式 $h(x) = z = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n + b = w^T x + b$, 则 b 可以融入到 w_0 , 即: $z=w^T x$

2.Sigmoid函数

8

线性回归的函数 $h(x) = z = w^T x$, 范围是 $(-\infty, +\infty)$ 。

而分类预测结果需要得到 $[0,1]$ 的概率值。

在二分类模型中, 事件的几率odds: 事件发生与事件不发生的概率之比为 $\frac{p}{1-p}$,

称为事件的发生比 (the odds of experiencing an event)

其中 p 为随机事件发生的概率, p 的范围为 $[0,1]$ 。

取对数得到: $\log \frac{p}{1-p}$, 而 $\log \frac{p}{1-p} = w^T x = z$

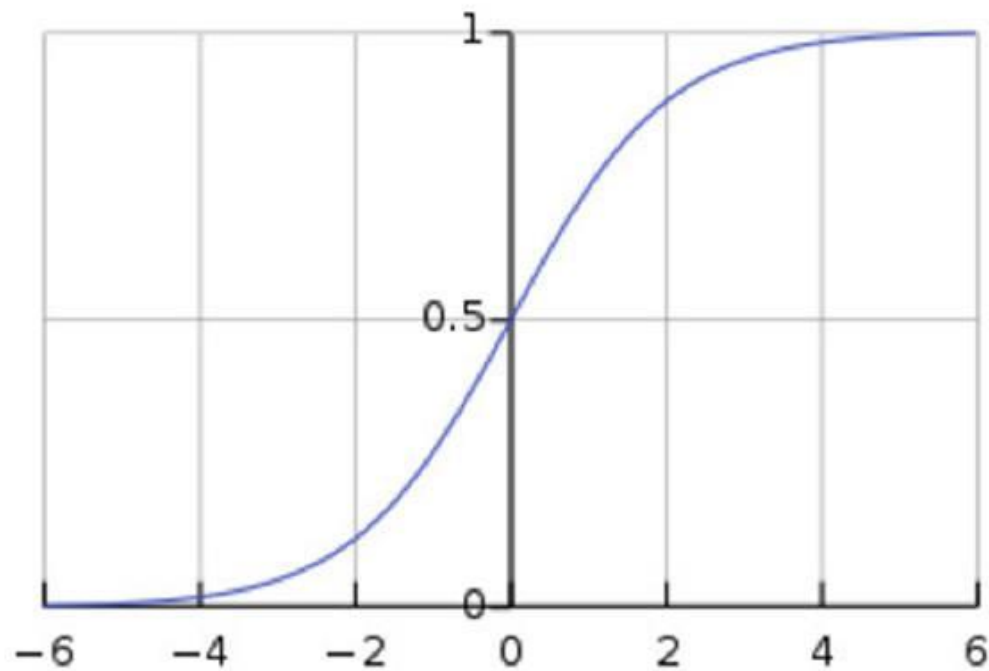
求解得到: $p = \frac{1}{1+e^{-w^T x}} = \frac{1}{1+e^{-z}}$

2.Sigmoid函数

9

将 z 进行逻辑变换: $g(z) = \frac{1}{1+e^{-z}}$

$$\begin{aligned} g'(z) &= \left(\frac{1}{1+e^{-z}} \right)' \\ &= \frac{e^{-z}}{(1+e^{-z})^2} \\ &= \frac{1+e^{-z}-1}{(1+e^{-z})^2} \\ &= \frac{1}{(1+e^{-z})} \left(1 - \frac{1}{(1+e^{-z})} \right) \\ &= g(z)(1-g(z)) \end{aligned}$$



$g(z)$

3.逻辑回归求解

10

01 分类问题

02 Sigmoid函数

03 逻辑回归求解

04 逻辑回归代码实现

3.逻辑回归求解

11

假设一个二分类模型：

$$p(y = 1|x; w) = h(x)$$

$$p(y = 0|x; w) = 1 - h(x)$$

则：

$$p(y|x; w) = (h(x))^y (1 - h(x))^{1-y}$$

逻辑回归模型的假设是： $h(x) = g(w^T x) = g(z)$

其中 $z = w^T x$, 逻辑函数 (**logistic function**)公式为：

$$g(z) = \frac{1}{1+e^{-z}}, \quad g'(z) = g(z)(1 - g(z))$$

3.逻辑回归求解

12

损失函数

$$L(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$

\hat{y} 表示预测值 $h(x)$

y 表示真实值

为了衡量算法在全部训练样本上的表现如何，我们需要定义一个算法的代价函数，算法的代价函数是对 m 个样本的损失函数求和然后除以 m ：

代价函数

$$J(w) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)}) = \frac{1}{m} \sum_{i=1}^m (-y^{(i)} \log \hat{y}^{(i)} - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}))$$

3.逻辑回归求解

13

求解过程：

似然函数为： $L(w) = \prod_{i=1}^m P(y^{(i)} | x^{(i)}; w) = \prod_{i=1}^m (h(x^{(i)}))^{y^{(i)}} (1 - h(x^{(i)}))^{1-y^{(i)}}$

似然函数两边取对数，则连乘号变成了连加号：

$$l(w) = \log L(w) = \sum_{i=1}^m (y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})))$$

代价函数为：

$$J(w) = -\frac{1}{m} l(w) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})))$$

3.逻辑回归求解

14

梯度下降求解过程：

$$w_j := w_j - \alpha \frac{\partial J(w)}{\partial w}$$

$$J(w) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})))$$

$$\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$\text{则： } w_j := w_j - \alpha \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

3.逻辑回归求解

15

求解过程： $\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_j^{(i)}$ 的推导过程：

$$J(w) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})))$$



$$\begin{aligned} & y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})) \\ &= y^{(i)} \log\left(\frac{1}{1 + e^{-w^T x^{(i)}}}\right) + (1 - y^{(i)}) \log\left(1 - \frac{1}{1 + e^{-w^T x^{(i)}}}\right) \\ &= -y^{(i)} \log(1 + e^{-w^T x^{(i)}}) - (1 - y^{(i)}) \log(1 + e^{w^T x^{(i)}}) \end{aligned}$$

3.逻辑回归求解

16

求解过程： $\frac{\partial}{\partial w_j} J(w) = \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_j^{(i)}$ 的推导过程：

$$\begin{aligned} \frac{\partial}{\partial w_j} J(w) &= \frac{\partial}{\partial w_j} \left(-\frac{1}{m} \sum_{i=1}^m \left(-y^{(i)} \log(1 + e^{-w^T x^{(i)}}) - (1 - y^{(i)}) \log(1 + e^{w^T x^{(i)}}) \right) \right) \\ &= -\frac{1}{m} \sum_{i=1}^m \left(-y^{(i)} \frac{-x_j^{(i)} e^{-w^T x^{(i)}}}{1 + e^{-w^T x^{(i)}}} - (1 - y^{(i)}) \frac{x_j^{(i)} e^{w^T x^{(i)}}}{1 + e^{w^T x^{(i)}}} \right) \\ &= -\frac{1}{m} \sum_{i=1}^m (y^{(i)} - h(x^{(i)})) x_j^{(i)} \\ &= \frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)}) x_j^{(i)} \end{aligned}$$

3.逻辑回归求解

17

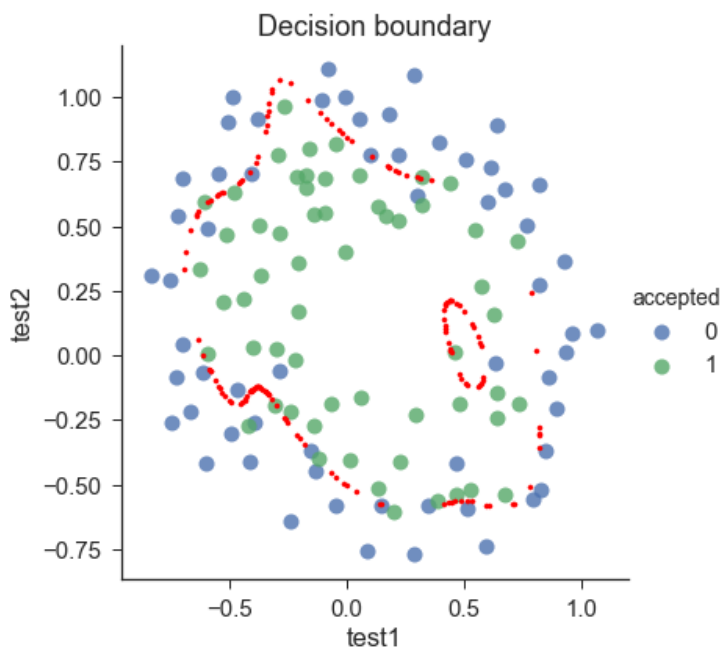
正则化：目的是为了**防止过拟合**

$$J(w) = \frac{1}{m} \sum_{i=1}^m \left[-y^{(i)} \log(h(x^{(i)})) - (1 - y^{(i)}) \log(1 - h(x^{(i)})) \right] +$$

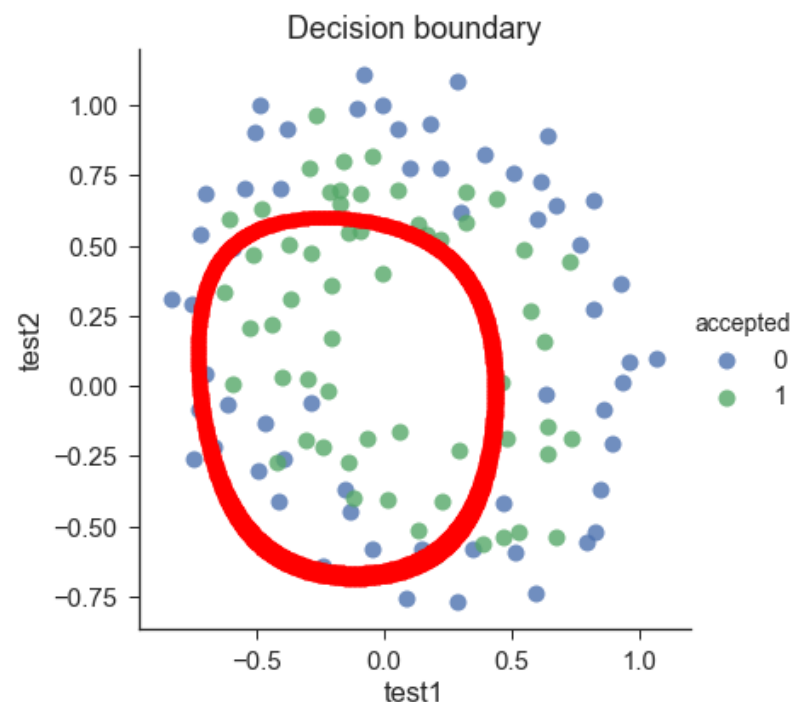
正则化项

$$\frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

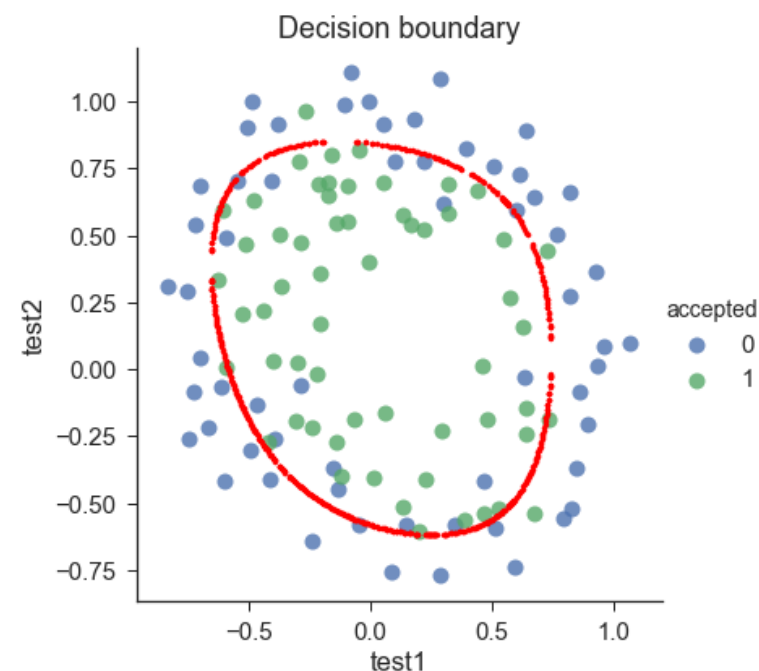
当 λ 的值开始上升时，降低了方差。



没有正则化，过拟合



正则化过度，欠拟合



适当的正则化

4.逻辑回归代码实现

18

01 分类问题

02 Sigmoid函数

03 逻辑回归求解

04 逻辑回归代码实现

4.逻辑回归代码实现

19

Sigmoid 函数

$$\sigma(z) = g(z) = \frac{1}{1+e^{-z}}$$

```
def sigmoid(z):  
    return 1 / (1 + np.exp(-z))
```

代价函数

$$J(w) = -\frac{1}{m} \sum_{i=1}^m (y^{(i)} \log(h(x^{(i)})) + (1 - y^{(i)}) \log(1 - h(x^{(i)})))$$

```
def cost(w, X, y):  
    w = np.matrix(w)  
    X = np.matrix(X)  
    y = np.matrix(y)  
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))  
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))  
    return np.sum(first - second) / (len(X))
```

4.逻辑回归代码实现

20

正则化

$$J(w) = \frac{1}{m} \sum_{i=1}^m \left[-y^{(i)} \log(h(x^{(i)})) - (1 - y^{(i)}) \log(1 - h(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

```
def costReg(w, X, y, LearningRate):  
    w = np.matrix(w)  
    X = np.matrix(X)  
    y = np.matrix(y)  
    first = np.multiply(-y, np.log(sigmoid(X * w.T)))  
    second = np.multiply((1 - y), np.log(1 - sigmoid(X * w.T)))  
    reg = (learningRate /  
           (2 * len(X))) * np.sum(np.power(w[:, 1:w.shape[1]], 2))  
    return np.sum(first - second) / len(X) + reg
```

1. Prof. Andrew Ng. Machine Learning. Stanford University
2. 《统计学习方法》，清华大学出版社，李航著，2019年出版
3. 《机器学习》，清华大学出版社，周志华著，2016年出版
4. Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006



谢谢观赏 下节课见

