

# Logistic回归

[ idaidai 的笔记本    机器学习    逻辑回归    最优化算法

## 简介

**Logistic回归**——我们经常用一条直线对一些点进行拟合（该线成为最佳拟合直线），这个拟合过程就称作回归。利用Logistic回归进行分类的主要思想是：根据现有数据对分类边界线建立回归公式，以此进行分类。

Logistic回归的一般过程为：

- **收集数据**：采用任意方法收集数据。
- **准备数据**：由于需要进行距离计算，因此要求数据类型采用数值型。另外，结构化数据格式则最佳。
- **分析数据**：采用任意方法对数据进行训练。
- **训练算法**：大部分时间将用于训练，训练的目的是为了找到最佳的分类回归函数。
- **测试算法**：一旦训练步骤完成，分类将会很快。
- **使用算法**：首先，我们需要输入一些数据，并将其转换成对应的结构化数值；接着，基于训练好的回归系数就可以对这些数值进行简单的回归计算，判定他们属于哪个类别；在这之后，我们就可以在输出的类别上做一些其他分析工作。

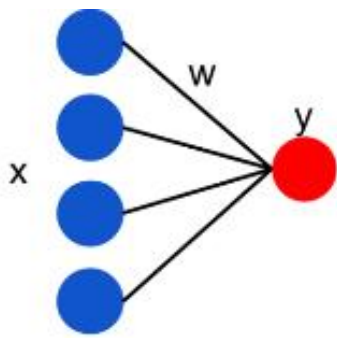
——好了，以上摘自机器学习实战这本书。写的有点抽象.....看了这些，**maybe**一些童鞋对逻辑回归有那么一丢丢的理解，但一定还是不知其所以然，那么下面我们从回归、线性回归和逻辑方程来逐步深入逻辑回归:)

## 神马是回归

回归按照我的理解就是有一堆数据，而且我们又事先知道这些数据之间的关系，举个例子说吧：比如我们知道弹簧的弹力与弹簧的形变量之间存在线性关系： $F=kx$ ，但是却不知道系数 $k$ 的值，那么我们可以通过大量的弹力和与之对应的形变量来计算出弹性系数，这个过程就叫做回归。也就是说，在进行回归的时候我们首先要知道变量与因变量之间的关系，而仅仅是去计算参数集而已。可是现实中，我们往往对于数据之间的关系不得而知，这就需要去观察数据然后预测数据之间可能的关系，然后再去回归参数集。

当然，根据数据间的关系，我们可以将回归分为线性回归（一元一次方程、二元一次方程.....）和非线性回归（ $N$ 元 $M$ 次方程、Log方程、指数方程.....）

## 神马是线性回归



这就是线性回归的图示： $y = w \times x$

**线性回归**我们举个最简单的例子，还是上面弹簧弹力与形变量之间的线性关系问题，它们之间是一元一次方程的关系，根据我们初中OR小学的知识我们可以知道：解这个方程仅仅需要一个F值和一个与之对应的x值就OK了，可是现实中由于这些数据都是测量出来的，不可避免的会出现误差甚至的错误，那么我们就需要大量数据来预测一个较为准确的弹性系数。通常采用的方式是，定义一个代价函数，通过最小化代价函数来估计参数集；或者定义代价函数为最大似然函数，通过最大化似然函数来估计参数集。

## 神马是Logistic方程

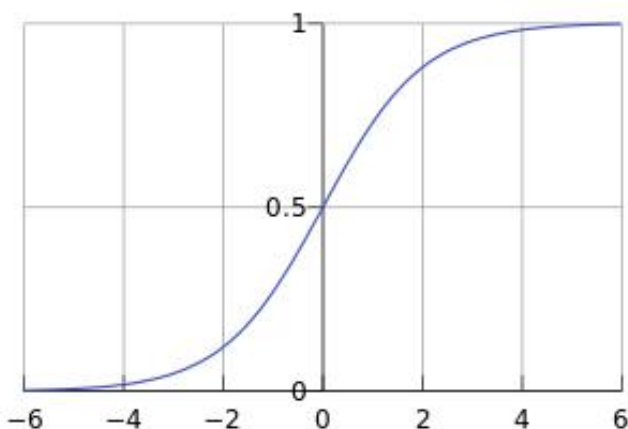
**Logistic方程**——直觉上，一个线性模型的输出值  $y$  越大，这个事件  $P(Y=1|x)$  发生的概率就越大。另一方面，我们可以用事件的几率（odds）来表示事件发生与不发生的比值，假设发生的概率是  $p$ ，那么发生的几率（odds）是  $p/(1-p)$ ，odds 的值域是 0 到正无穷，几率越大，发生的可能性越大。将我们的直觉与几率联系起来的就是下面这个（log odds）或者是 logit 函数（有点牵强 - -!）：

$$\ln \left( \frac{P}{1-P} \right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$$

其中， $P$  和  $Q$  就为行向量。进而可以求出  $l$  关于  $P$  和  $Q$  的表示：

$$l = - \sum_{i=1}^n P_i \ln P_i - \sum_{i=1}^n Q_i \ln Q_i$$

看到这里，恭喜你！终于见到传说中的Logistic函数了，其实就是人为的将线性回归的结果与对数概率结合在了一起！以  $P$  和  $Q$  为自变量的函数图像如下所示：



我们可以看得出，当横坐标拉伸时，该函数图像就像是一个阶跃函数一样，因此可以用它来做分类！

# 参数估计方法

上面也简单提到了一下，对于线性回归大致我了解到的有两种参数估计方法：

- **最大似然法：**逻辑回归输出的是分到每一类的概率，参数估计的方法自然可以用最大似然估计 (MLE) 咯。对于训练样本来说，假设每个样本是独立的，输出（标签）为  $R$ ； $\cdot$ \*/，样本的似然函数就是将所有训练样本  $label$  对应的输出节点上的概率相乘，令  $I$ ； $/ \&8$ ； $/z_Q'$ ，如果  $R$ ； $/$ ，概率就是  $I$ ，如果  $R$ ； $\cdot$ ，概率就是  $1 - I$ ，（好吧，我是个罗嗦的家伙），将这两种情况合二为一，得到似然函数：

$$L = \prod_{B=1}^N \left( \frac{Y_B}{\&8} ; / z_{Q_B}' \right)^{Y_B} \left( \frac{\&8} ; / z_{Q_B}' } \right)^{1 - Y_B}$$

有连乘，这个好办，取对数就OK：

$$\begin{aligned} L &= \prod_{B=1}^N \left( \frac{Y_B}{\&8} ; / z_{Q_B}' \right)^{Y_B} \left( \frac{\&8} ; / z_{Q_B}' } \right)^{1 - Y_B} \\ \ln L &= \sum_{B=1}^N \left( Y_B \ln \left( \frac{Y_B}{\&8} ; / z_{Q_B}' \right) + (1 - Y_B) \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) \right) \\ &= \sum_{B=1}^N \left( Y_B \ln \left( \frac{Y_B}{\&8} ; / z_{Q_B}' \right) - Y_B \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) + (1 - Y_B) \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) \right) \end{aligned}$$

根据上面的对数概率公式，我们可以得到：

$$\ln L = \sum_{B=1}^N \left( Y_B \ln \left( \frac{Y_B}{\&8} ; / z_{Q_B}' \right) - Y_B \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) + (1 - Y_B) \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) \right)$$

对  $P$  求导可以得到：

$$\frac{\partial \ln L}{\partial P} = \sum_{B=1}^N \left( Y_B \ln \left( \frac{Y_B}{\&8} ; / z_{Q_B}' \right) - Y_B \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) + (1 - Y_B) \ln \left( \frac{\&8} ; / z_{Q_B}' } \right) \right)$$

- **平方损失函数法：**

## 代码块

```
@requires_authorization
def somefunc(param1='', param2=0):
    '''A docstring'''
    if param1 > param2: # interesting
        print 'Greater'
    return (param2 - param1 + 1) or None
class SomeClass:
    pass
>>> message = '''interpreter
... prompt'''
```

## LaTex 公式

$$Q ; \frac{I ; ^4 ; ^0 I 2 : <}{0 :}$$

## 表格

Item	Value	Qty
Computer	1600 USD	5
Phone	12 USD	12
Pipe	1 USD	234

## 反馈与建议

- 微博: [@左手流星雨](#)
- 邮箱: [utopiar@gs.zzu.edu.cn](mailto:utopiar@gs.zzu.edu.cn)

---

感谢阅读本人笔记。