

# 逻辑回归

Mia Feng

2018 年 4 月 8 日

## 1 概述

逻辑回归是广义线性模型的一种，适用的数据为可分的。主要用于解决二分类问题，也可以扩展至多分类问题求解目标：分类超平面，以二维空间为例，求解一条直线

$$y = b + \theta_1 x_1 + \theta_2 x_2 = \theta^T x + b \quad (1)$$

求解思路：求解  $\theta$  的控制方程选择概率描述。使样本点被分类至真实标签的概率最大。推导证明 logistic 分布的鲁棒性最好，（见 LogisticRegression-MaxEnt.pdf）简单来说，因为  $p(y|x) \sim \text{Bernoulli}$ ，根据最大熵原则求解分布（也就是说这个分布应该在满足假设的前提下越均匀越好），推导得到线性模型  $f(\theta x)$ ，结果为 sigmoid 函数，即 Bernoulli 的指数家族形式

求解方法：最大似然估计

迭代方法：gradient ascent

### 1.1 推导

**Sigmoid 函数**

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \quad (2)$$

基于二分类的假定，每个样本点的概率为

$$p(y|x, \theta) = h_\theta(x)^y (1 - h_\theta(x))^{1-y} \quad (3)$$

这里只是一个技巧性的推导，代入标签值计算一下即可得到。

又基于 i.i.d. 假设（独立同分布）， $m$  个样本点的后验概率为：

$$L(\theta) = \prod_{i=1}^m p(y_i | x_i, \theta) = \prod_{i=1}^m h_{\theta}(x_i)^{y_i} (1 - h_{\theta}(x_i))^{1-y_i} \quad (4)$$

为便于计算，取对数

$$L(\theta) = \ln L(\theta) = \sum_{i=1}^m y_i \ln h_{\theta}(x_i) + (1 - y_i) \ln (1 - h_{\theta}(x_i)) \quad (5)$$

接下来计算梯度，设置梯度更新，迭代求解  $\theta$  参考的 blog，打字太多不写了，把他? 的推导粘过来了，推倒还是很明了的。梯度：

The image shows a handwritten derivation of the partial gradient of the log-likelihood function  $J(\theta)$  with respect to  $\theta_j$ . The derivation proceeds as follows:

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) &= -\frac{1}{m} \sum_{i=1}^m \left( y_i \frac{1}{h_{\theta}(x_i)} \frac{\partial}{\partial \theta_j} h_{\theta}(x_i) - (1-y_i) \frac{1}{1-h_{\theta}(x_i)} \frac{\partial}{\partial \theta_j} h_{\theta}(x_i) \right) \\ &= -\frac{1}{m} \sum_{i=1}^m \left( y_i \frac{1}{g(\theta^T x_i)} - (1-y_i) \frac{1}{1-g(\theta^T x_i)} \right) \frac{\partial}{\partial \theta_j} g(\theta^T x_i) \\ &= -\frac{1}{m} \sum_{i=1}^m \left( y_i \frac{1}{g(\theta^T x_i)} - (1-y_i) \frac{1}{1-g(\theta^T x_i)} \right) \frac{\partial}{\partial \theta_j} \left( \frac{1}{1+e^{-\theta^T x_i}} \right) \\ &= -\frac{1}{m} \sum_{i=1}^m \left( y_i \frac{1}{g(\theta^T x_i)} - (1-y_i) \frac{1}{1-g(\theta^T x_i)} \right) \left( -\frac{e^{-\theta^T x_i}}{(1+e^{-\theta^T x_i})^2} \cdot x_{ij} \right) \\ &\quad \text{ } x_{ij} \text{ 表示第 } i \text{ 个样本的第 } j \text{ 个特征} \\ &= -\frac{1}{m} \sum_{i=1}^m \left( y_i \frac{1}{g(\theta^T x_i)} - (1-y_i) \frac{1}{1-g(\theta^T x_i)} \right) g(\theta^T x_i) (1-g(\theta^T x_i)) x_{ij} \\ &= -\frac{1}{m} \sum_{i=1}^m (y_i (1-g(\theta^T x_i)) - (1-y_i) g(\theta^T x_i)) x_{ij} \\ &= \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i) x_{ij} \end{aligned}$$

图 1: Partial gradient derivation

梯度更新公式：

$$\theta_j^{t+1} = \theta_j^t + \alpha \frac{\partial}{\partial \theta_j} L(\theta_j) = \theta_j^t + \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i) x_{ij} \quad (6)$$

## 2 Pesudo Code

1. 初始化回归系数  $\theta$
2. 重复下面步骤直至收敛
  - {
  - 计算整个数据集的梯度
  - 使用公式 (6) 更新梯度
  - }
3. 返回回归系数  $\theta$

Note: 程序实现中 gradient ascent 和 gradient descent 都是可以的, 全看损失函数怎么写。如果计算的是  $h(x) - y$ , 那么用 gradient descent; 反之如果计算的是  $y - h(x)$ , 那么用 gradient ascent。