Logistic Loss函数

原创 buracag_mc 2019-04-26 17:51:20 5638 收藏 5 版权

分类专栏： 统计学运用 技术备忘

同步于音尘杂记

前面在浏览sklearn中关于Logistic Regression部分，看到关于带正则项的LR目标损失函数的定义形式的时候，对具体表达式有点困惑，后查阅资料，将思路整理如下。

文章目录

1. sklearn文档中的LR损失函数
2. LR损失函数
    2.1 logistic基础知识
    2.2 旧思路
    2.3 新思路
3. 思考

1. sklearn文档中的LR损失函数

先看sklearn对于LR目标损失函数(带L2)的定义：

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1)$$

看到这个表达形式，其实是有两个疑问：

- logistic loss的表达形式
- 正则项的惩罚系数

对于第二个问题，其实比较容易解释。通常我们在最小化结构风险时，会给我们的惩罚项乘上一个惩罚系数λ(通常1 < λ < 0)，

$$\min_{w,\lambda} \sum_{i=1}^{n} loss(y, y_i) + \lambda w^T w$$

一般，为方便处理，做一个技巧性地处理，对多项式乘上一个正数 1/2λ, 得到：

$$\min_{w,\lambda} \frac{1}{2\lambda} \sum_{i=1}^{n} loss(y, y_i) + \frac{1}{2} w^T w$$

令C = 1/2λ即可。

但是对于第一个形式，当时比较困惑；特意翻看了一下我以前记录的关于LR以及LR损失函数的一些笔记。

2. LR损失函数

为了方便说明笔者当时的疑惑所在，便将当时脑海里存在的logistic loss函数形式 和 sklearn中LR损失函数的推导方法分别记为旧思路和新思路吧。

2.1 logistic基础知识

如指数分布、高斯分布等分布一样，logistic是一种变量的分布，它也有自己的概率分布函数和概率密度函数，其中概率分布函数如下：

$$F(x) = P(X \le x) = \frac{1}{1 + e^{-(x-\mu)/\gamma}}$$

对概率分布函数求导，记得到对应的概率密度函数：

$$f(x) = \frac{e^{-(x-\mu)/\gamma}}{\gamma(1 + e^{-(x-\mu)/\gamma})^2}$$

其中，μ就是分布对应的均值，γ是对应的形状参数。

下文，为简介方便起见，将$-(x-\mu)/\gamma$替换为$-x$,故记为：

$$F(x) = \frac{1}{1 + exp(-x)}$$

对应示例图如下：

logistic有一个很重要的性质是：

$$F(-x) = \frac{1}{1 + exp(x)} = \frac{1}{1 + \frac{1}{exp(-x)}} = \frac{exp(-x)}{1 + exp(-x)} = 1 - \frac{1}{1 + exp(-x)} = 1 - F(x)$$

通常，应用到LR中，有如下形式：

> **(1)**
>
> $$P(Y = 1|\beta, x) = \frac{1}{1 + exp(-\beta x)} = \frac{e^{\beta x}}{1 + e^{\beta x}}$$
>
> $$P(Y = 0|\beta, x) = 1 - \frac{1}{1 + exp(-\beta x)} = \frac{1}{1 + e^{\beta x}}$$

> 一个事件的几率(odds)，定义为该事件发生与不发生的概率比值，若事件发生概率为p：
>
> $$odds = \frac{p}{1 - p}$$

那么该事件的对数几率（log odds或者logit）如下：

$$logit(p) = log\frac{p}{1-p}$$

那么，对于上述二项，Y=1的对数几率就是：

$$log\frac{P(Y = 1|\beta, x)}{1-P(Y = 1|\beta, x)} = log\frac{P(Y = 1|\beta, x)}{P(Y = 0|\beta, x)} = \beta x$$

也就是说，输出Y=1的对数几率是由输入x的线性函数表示的模型，这就是逻辑回归模型。易知，当 βx的值越大，$P(Y = 1|\beta, x)$越接近1；βx越小,$P(Y = 1|\beta, x)$ 越接近0。

其实，LR就是一个线性分类的模型。与线性回归不同的是：LR将线性方程输出的很大范围的数压缩到了[0,1]区间上；更优雅地说：**LR就是一个被logistic方程归一化后的线性回归。**

2.2 旧思路

旧思路要从LR的参数求解过程说起。

我们知道统计学中一种很常用的方法是根据最大化似然函数的值来估计总体参数。在机器学习领域，我们听到的更多是损失函数的概念，常通过构建损失函数，然后最小化损失函数估计目标参数。在这里，**最大化对数似然函数与最小化对数似然损失函数其实是等价的**，下面我们可以看到。

- 假设我们有n个独立的训练样本$\{(x_1,y_1),(x_2,y_2),(x_3,y_3),...,(x_n,y_n)\}, y=0,1$,那么每一个观察到的样本$(x_i,y_i)$出现的概率是：

$$P(y_i,x_i)=P(y_i=1|x_i)^{y_i}(1-P(y_i=1|x_i))^{1-y_i}$$

显然，y_i为1时，保留前半部分；y_i为0时，保留后半部分。

- 构建似然函数：

$$L(\beta)=\prod P(y_i=1|x_i)^{y_i}(1-P(y_i=1|x_i))^{1-y_i}$$

- OK,对似然函数取对数，得到对数似然函数：

$$LL(\beta)=log(L(\beta))=log(\prod P(y_i=1|x_i)^{y_i}(1-P(y_i=1|x_i))^{1-y_i})$$

$=\sum_{i=1}^{n}(y_i logP(y_i=1|x_i)+(1-y_i)log(1-P(y_i=1|x_i)))$

$=\sum_{i=1}^{n}y_i log\frac{P(y_i=1|x_i)}{1-P(y_i=1|x_i)}+\sum_{i=1}^{n}log(1-P(y_i=1|x_i))$

$=\sum_{i=1}^{n}y_i(\beta x)+\sum_{i=1}^{n}logP(y_i=0|x_i)$

$=\sum_{i=1}^{n}y_i(\beta x)-\sum_{i=1}^{n}log(1+e^{\beta x})$

- 用 $LL(\beta)$ 对 β 求偏导，得：
$\frac{\partial LL(\beta)}{\partial \beta}=\sum_{i=1}^{n}y_i x_i-\sum_{i=1}^{n}\frac{e^{\beta x_i}}{1+e^{\beta x_i}}.x_i$

$=\sum_{i=1}^{n}(y_i-P(y_i=1|x_i))x_i$

该式是无法解析求解，故会用到一些优化算法进行求解(梯度下降、牛顿法等)，这不是本文重点，便不再赘述。

咋一看的确与sklearn中的形式差别有点大，所以请看新思路。

2.3 新思路

在式(1)中，x表示特征向量，β表示相应的超参数，此时$y\in(0,1)$表示样本对应的标签(label)。

这里，特别要讲的是另一种表达形式，将标签与预测函数在形式上统一了：

> **(2)**
>
> $$P(g=\pm 1|\beta,x)=\frac{1}{1+exp(-g\beta x)}$$

此时的样本标签$g\in(1,-1)$。

虽然式(1)与式(2)看起来似乎不同，但是我们可以有如下证明：

$$P(Y=1|\beta,x)=\frac{e^{\beta x}}{1+e^{\beta x}}=\frac{1}{1+exp(-\beta x)}=P(g=1|\beta,x)$$

同理，我们可以证明$P(Y=0|\beta,x)$ 和 $P(g=-1|\beta,x)$是等价的。

既然两种形式是等价的，为了适应更加广泛的分类loss最小化的框架，故采用第二种形式来表示LR.毕竟**Simple is better than complex.**

首先定义x_i为特征向量，y_i为样本标签,则目标损失函数可以表示为：

$$arg\min_{\beta}\sum_{i=1}L(y_i,f(x_i))$$

其中，f是我们的回归方程，L是目标损失函数。

对应到LR中，我们有

$$f(x)=\beta x$$

$$L(y,f(x))=log(1+exp(-yf(x)))$$

如果将LR的第二种表达形式带入到损失函数L中，可得：

$$L(y,f(x))=log(1+exp(-yf(x)))=log(\frac{1}{P(y|\beta,x)})$$

再进一步：

$$arg\min_{\beta}\sum_{i=1}L(y_i,f(x_i))=arg\min_{\beta}\sum_{i=1}log(\frac{1}{P(y_i|\beta,x_i)})$$

$$=arg\max_{\beta}\sum_{i=1}log(P(y_i|\beta,x_i))=arg\max_{\beta}\prod_{i=1}P(y_i|\beta,x_i)$$

等式最后即为极大似然估计的表达形式。

3. 思考

其实到这儿，我们不难发现在旧思路中，推导极大化对数似然函数中的第二步：
$=\sum_{i=1}^{n}(y_ilogP(y_i=1|x_i)+(1-y_i)log(1-P(y_i=1|x_i)))$

与新思路中的：

$$=arg\max_{\beta}\sum_{i=1}log(P(y_i|\beta,x_i))$$

本质是统一的。

最后

> **"Simple is better than complex." – The Zen of Python, by Tim Peters**