

[\[关闭\]](#)

@frank-shaw 2015-07-31 20:52 字数 2457 阅读 15444

Logistic Loss函数、Logistics回归与极大似然估计

机器学习理论

一直对Loss函数的类型的具体由来是怎样的弄不清楚。现在学到了经验风险最小化方面的知识，感觉可以尝试去探索一番。

Logistic函数与Logistic回归

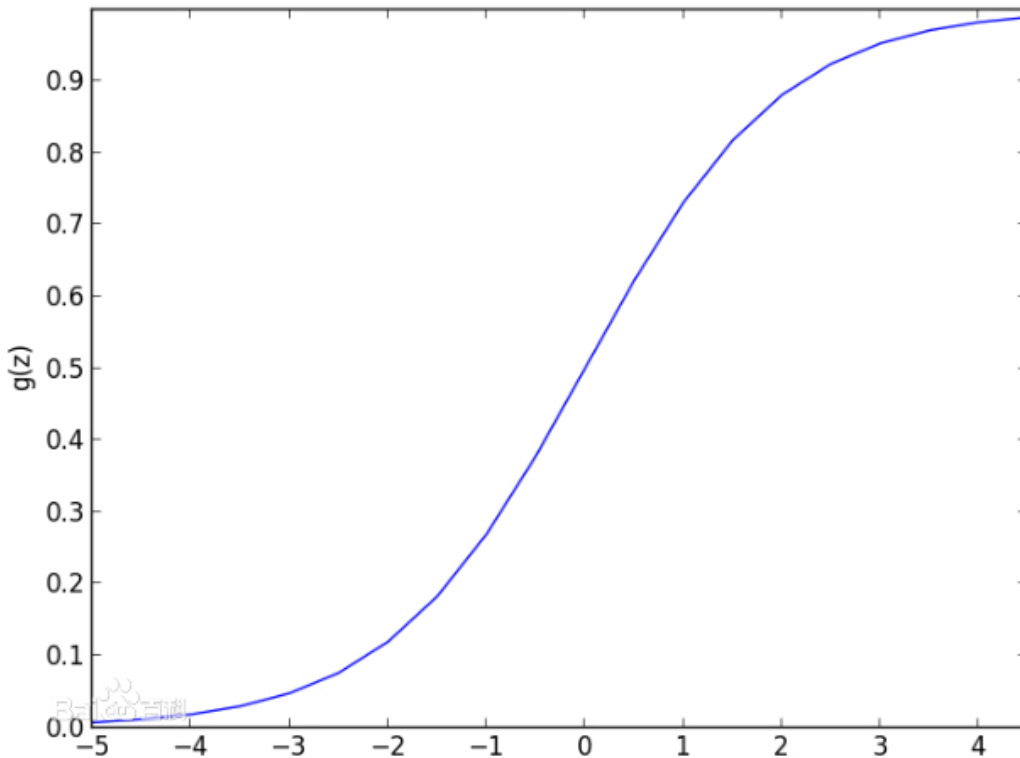
通常，Logistic函数的定义如下：

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

其中 $x \in \mathbb{R}$, $P(x) \in [0, 1]$ 。其中一个重要性质为：

$$\begin{aligned} P(-x) &= \frac{1}{1 + \exp(x)} = \frac{1}{1 + \frac{1}{\exp(-x)}} \\ &= \frac{\exp(-x)}{1 + \exp(-x)} = 1 - \frac{1}{1 + \exp(-x)} = 1 - P(x) \end{aligned}$$

logistic函数图像：



公式 (1) 则是被应用到了Logistic回归中,常见形式如下：

$$\begin{aligned}
 P(y = 1|\beta, x) &= \frac{1}{1 + \exp(-\beta^T x)} = \frac{\exp(\beta^T x)}{1 + \exp(\beta^T x)} \\
 P(y = 0|\beta, x) &= 1 - \frac{1}{1 + \exp(-\beta^T x)} = \frac{1}{1 + \exp(\beta^T x)}
 \end{aligned} \tag{2}$$


其中 β 为相应参数， x 表示特征向量，此时 $y \in \{0, 1\}$ 表示的是样本标签。

另一种表示形式将标签与预测函数放在了一起：


$$P(g = \pm 1|\beta, x) = \frac{1}{1 + \exp(-g\beta^T x)} \tag{3}$$

此时的样本标签 $g \in \{\pm 1\}$ 。很容易证明 $P(g = 1|\beta, x) = 1 - P(g = -1|\beta, x)$ 。显然，这种形式和第一种logistic回归形式本质上并没有区别。

第一种形式的分类法则：

logistics回归1

相似的，第二中形式的分类法则：

logistics回归2

Logistic Loss

既然两种形式是等价的，为了适应更加广泛的分类Loss最小化框架，我们使用第二种形式来表示Logistic回归。

首先定义 y 为样本标签， x 为特征向量。该分类Loss最小化框架可以表示为：

$$\operatorname{argmin}_{\beta} \sum_i L(y_i, f(x_i))$$

其中 f 为假设函数， L 表示的是loss函数。

对于logistic回归，对应于该分类框架，我们有：

$$\begin{aligned}
 f(x) &= \beta^T x \\
 L(y, f(x)) &= \log(1 + \exp(-yf(x)))
 \end{aligned}$$

这里使用的Loss函数即为Logistic Loss函数。实际上，我们可以通过该Loss最小化框架得到极大似然法则。如果将Logistic回归**第二种表示形式**代入到此时的 $L(y, f(x))$ ，可得：

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

由此，Loss最小化可以表示为：

$$\begin{aligned}
 \operatorname{argmin}_{\beta} \sum_i L(y_i, f(x_i)) &= \operatorname{argmin}_{\beta} \sum_i \log\left(\frac{1}{P(y_i|\beta, x_i)}\right) \\
 &= \operatorname{argmax}_{\beta} \sum_i \log(P(y_i|\beta, x_i)) = \operatorname{argmax}_{\beta} \prod_i P(y_i|\beta, x_i)
 \end{aligned}$$

上式等式最后即为极大似然估计的表示形式。也就是说，Logistic回归模型使用的Loss函数为Logistic Loss函数，使用极大似然估计法的目的是为了使得该Loss函数最小。

感想

这个时候，我似乎想明白了很多事情。将之前零散的知识点串联起来了。网易《机器学习》第二课中讲到线性回归的时候，将 $\frac{1}{2} \sum_i (f(x_i) - y_i)^2$ 作为Loss函数，最终通过极大似然估计解释了使用这个Loss函数的原因。接着就直接使用极大似然估计来求解Logistic回归问题，至于为什么以及最小化的是哪一个Loss函数，并没有提及。直到现在才弄懂。

当然，如果在Loss函数后面加上一个关于变量的L2范数，这个时候可以推导出贝叶斯学派的极大后验概率估计法则（MAP），在此不展开。似乎，很多算法之间的差异性都可以用Loss函数来解释。

参考文献：

《Regularized Regression under Quadratic Loss, Logistic Loss, Sigmoidal Loss, and Hinge Loss》
《Notes on Logistic Loss Function》

• 内容目录

-
- [Logistic Loss函数、Logistics回归与极大似然估计](#)
 - [Logistic函数与Logistic回归](#)
 - [Logistic Loss](#)
 - [感想](#)
 -
 - - 0目录 1
 - [frank-shaw的博客目录](#)
 - - DataAtlas 1
 - [分区简单使用](#)
 - - HTML 3
 - [HTML笔记目录](#)
 - [HTML标签含义总结](#)
 - [input标签各属性分析](#)
 - - Hadoop 6
 - [linux下安装Hadoopeclipse插件以及编写第一个简单的MapReduce程序](#)
 - [JDK与Hadoop的安装配置过程（Hadoop单机版）](#)
 - [hadoop伪分布下三个配置文件的参数含义说明core-site.xml、hdfs-site.xml mapred-site.xml](#)
 - [设置虚拟机静态IP](#)
 - [SSH配置Hadoop能够不需要密码就能够运行](#)
 - [MapReduce编程模型及其在Hadoop上的实现](#)
 - - VNode 1
 - [vue源码阅读（五）：虚拟DOM的引入](#)
 - - angular.js 7
 - [自定义指令中的transclude选项理解](#)
 - [angularJS的compile过程与link过程](#)
 - [angularJs中自定义指令的scope参数以及绑定策略](#)
 - [AngularJS目录](#)
 - [angular中的依赖注入](#)
 - [angularJS自定义依赖service](#)
 - [angularJS中关于controller的实现方式](#)
 - - java 6
 - [java.内存](#)
 - [java.集合](#)
 - [java.类加载](#)
 - [java.异常](#)
 - [java.多线程](#)
 - [java.基础知识](#)
 - - java.内存 6
 - [JVM内存相关参数设置总结](#)
 - [Minor GC、Major GC和Full GC之间的区别](#)