机器学习算法系列（3）：逻辑斯谛回归

一、逻辑斯谛分布

介绍逻辑斯谛回归模型之前，首先看一个并不常见的概率分布，即逻辑斯谛分布。设X是连续随机变量，X服从逻辑斯谛分布是指X具有下列的分布函数和密度函数：

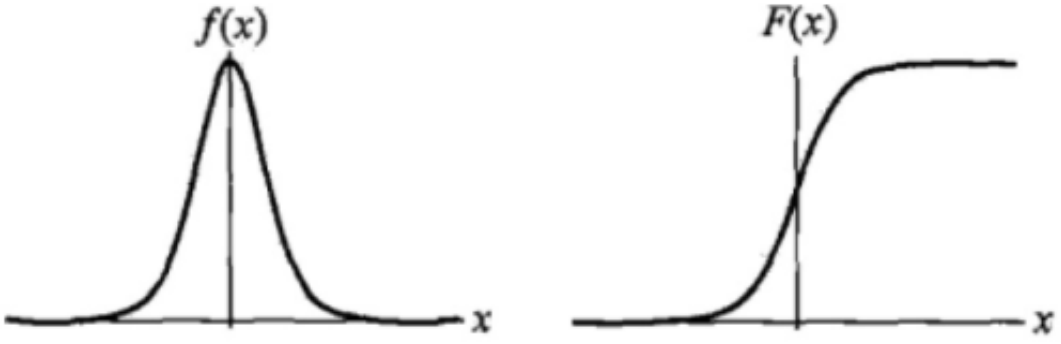
$$F(x) = P(X \le x) = \frac{1}{1 + e^{-(x-\mu)/\gamma}}$$

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

式中，μ为位置参数，$\gamma > 0$为形状参数。逻辑斯谛的分布的密度函数$f(x)$和分布函数$F(x)$的图形如下图所示。其中分布函数属于逻辑斯谛函数，其图形为一条S形曲线。该曲线以点$(\mu, \frac{1}{2})$为中心对称，即满足

$$F(-x + \mu) - \frac{1}{2} = -F(x + \mu) + \frac{1}{2}$$

曲线在中心附近增长较快，在两端增长速度较慢。形状参数γ的值越小，曲线在中心附近增长得越快。



二、逻辑斯谛回归模型

线性回归的应用场合大多是回归分析，一般不用在分类问题上。原因可以概括为以下两个：

- 1）回归模型是连续型模型，即预测出的值都是连续值（实数值），非离散值；

- 2）预测结果受样本噪声的影响比较大。

2.1 LR模型表达式

LR模型表达式为参数化的逻辑斯谛函数（默认参数$\mu = 0, \gamma = 1$）,即

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

其中$h_\theta(x)$作为事件结果$y = 1$的概率取值。这里,$x \in R^{n+1}, y \in \{1, 0\}, \theta \in R^{n+1}$是权值向量。其中权值向量$w$中包含偏置项，即$w = (w_0, w_1, \cdots, w_n)$ ， $x = (1, x_1, x_2, \cdots, x_n)$

2.2 理解LR模型

2.2.1 对数几率

一个事件发生的几率（odds）是指该事件发生的概率与该事件不发生的概率的比值。如果事件发生的概率是p，那么该事件的几率为$\frac{p}{1-p}$，该事件的对数几率（log odds）或logit函数是：

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

对LR而言，根据模型表达式可以得到：

$$\log \frac{h_\theta(x)}{1 - h_\theta(x)} = \theta^T x$$

即在LR模型中，输出$y = 1$的对数几率是输入x的线性函数。或者说输出$y = 1$的对数几率是由输入x的线性函数表示的模型，即LR模型

2.2.2 函数映射

除了从对数几率的角度理解LR外，从函数映射也可以理解LR模型。

考虑对输入实例x进行分类的线性表达式θ^T，其值域为实数域。通过LR模型表达式可以将线性函数$\theta^T x$的结果映射到(0,1)区间，取值表示为结果为1的概率（在二分类场景中）。

线性函数的值越接近于正无穷大，概率值就越接近1；反之，其值越接近于负无穷，概率值就越接近0。这样的模型就是LR模型。

LR本质上还是线性回归，知识特征到结果的映射过程中加了一层函数映射（即sigmoid函数），即先把特征线性求和，然后使用sigmoid函数将线性和约束至（0，1）之间，结果值用于二分或

回归预测。

2.2.3 概率解释

LR模型多用于解决二分类问题，如广告是否被点击（是/否）、商品是否被购买（是/否）等互联网领域中常见的应用场景。但是实际场景中，我们又不把它处理成“绝对的”分类问题，而是用其预测值作为事件发生的概率。

这里从事件、变量以及结果的角度给予解释。

我们所能拿到的训练数据统称为观测样本。问题：样本是如何生成的?

一个样本可以理解为发生的一次事件，样本生成的过程即事件发生的过程。对于0/1分类问题来讲，产生的结果有两种可能，符合伯努利试验的概率假设。因此，我们可以说样本的生成过程即为伯努利试验过程，产生的结果（0/1）服从伯努利分布。这里我们假设结果为1的概率为$h_\theta(x)$，结果为0的概率为$1-h_\theta(x)$。

那么对于第i个样本，概率公式表示如下：

$$P(y^{(i)}=1\,|\,x^{(i)};\theta)=h_\theta(x^{(i)})$$

$$P(y^{(i)}=0\,|\,x^{(i)};\theta)=1-h_\theta(x^{(i)})$$

将上面两个公式合并在一起，可得到第i个样本正确预测的概率：

$$P(y^{(i)}\,|\,x^{(i)};\theta)=(h_\theta(x^{(i)})^{y(i)})\cdot\quad(\,1-h_\theta(x^{(i)}))^{1-y(i)}$$

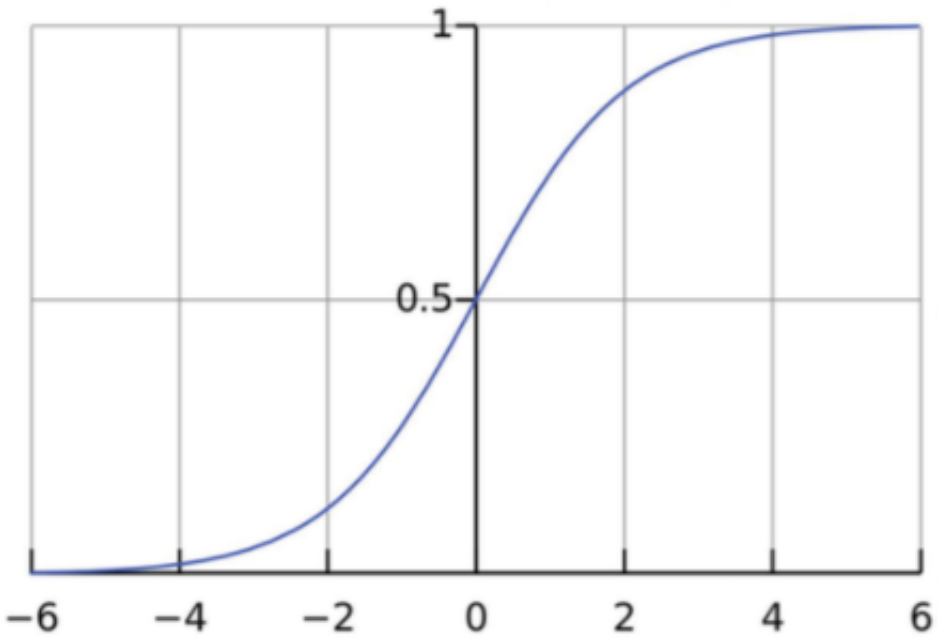
上式是对一个样本进行建模的数据表达。对于所有的样本，假设每条样本生成过程独立，在整个样本空间中（N个样本）的概率分布（即似然函数）为：

$$P(Y|X;\theta)=\prod_{i=1}^{N}\left(h_\theta\left(x^{(i)}\right)^{y^{(i)}}\left(1-h_\theta\left(x^{(i)}\right)^{1-y^{(i)}}\right)\right)$$

通过极大似然估计（Maximum Likelihood Evaluation，简称MLE）方法求概率参数。具体地，第三节给出了通过随机梯度下降法（SGD）求参数。

三、模型参数估计

3.1 Sigmoid函数



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

上图所示即为sigmoid函数，它的输入范围为$-\infty \rightarrow +\infty$，而值域刚好为$(0, 1)$，正好满足概率分布为$(0, 1)$的要求。用概率去描述分类器，自然要比阈值要来的方便。而且它是一个单调上升的函数，具有良好的连续性，不存在不连续点。

此外非常重要的，sigmoid函数求导后为：

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

以下的推导中会用到，带来了很大的便利。

3.2 参数估计推导

上一节的公式不仅可以理解为在已观测的样本空间中的概率分布表达式。如果从统计学的角度可

以理解为参数θ似然性的函数表达式（即似然函数表达式）。就是利用已知的样本分布，找到最有可能（即最大概率）导致这种分布的参数值；或者说什么样的参数才能使我们观测到目前这组数据的概率最大。参数在整个样本空间的似然函数可表示为：

$$L(\theta) = P\left(\vec{Y} \mid X; \theta\right)$$

$$= \prod_{i=1}^{N} P\left(y^{(i)} \parallel x^{(i)}; \theta\right)$$

$$= \prod_{i=1}^{N} \left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - h_\theta\left(x^{(i)}\right)\right)^{1-y^{(i)}}$$

为了方便参数求解，对这个公式取对数，可得对数似然函数：

$$l(\theta) = \sum_{i=1}^{N} \log l(\theta)$$

$$= \sum_{i=1}^{N} y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right)$$

最大化对数似然函数其实就是最小化交叉熵误差（Cross Entropy Error）。
先不考虑累加和，我们针对每一个参数w_j求偏导：

$$\frac{\partial}{\partial \theta_j} l(\theta) = \left(y \frac{1}{h_\theta(x)} - (1-y) \frac{1}{1 - h_\theta(x)}\right) \frac{\partial}{\partial \theta_j} h_\theta(x)$$

$$= \left(\frac{y\left(1 - h_\theta(x)\right) - (1-y) h_\theta(x)}{h_\theta(x)\left(1 - h_\theta(x)\right)}\right) h_\theta(x)\left(1 - h_\theta(x)\right) \frac{\partial}{\partial \theta_j} \theta^T x$$

$$= \left(y - h_\theta(x)\right) x_j$$

最后，通过扫描样本，迭代下述公式可求得参数：

$$\theta_j := \theta_j + a\left(y^{(i)} - h_\theta\left(x^{(i)}\right)\right) x_j^{(i)}$$

其中a表示学习率，又称学习步长。此外还有Batch GD，共轭梯度，拟牛顿法（LBFGS），ADMM分布学习算法等都可以用来求解参数。另作优化算法一章进行补充。

以上的推导是LR模型的核心部分，在机器学习相关面试中，LR模型公式推导可能是考察频次最

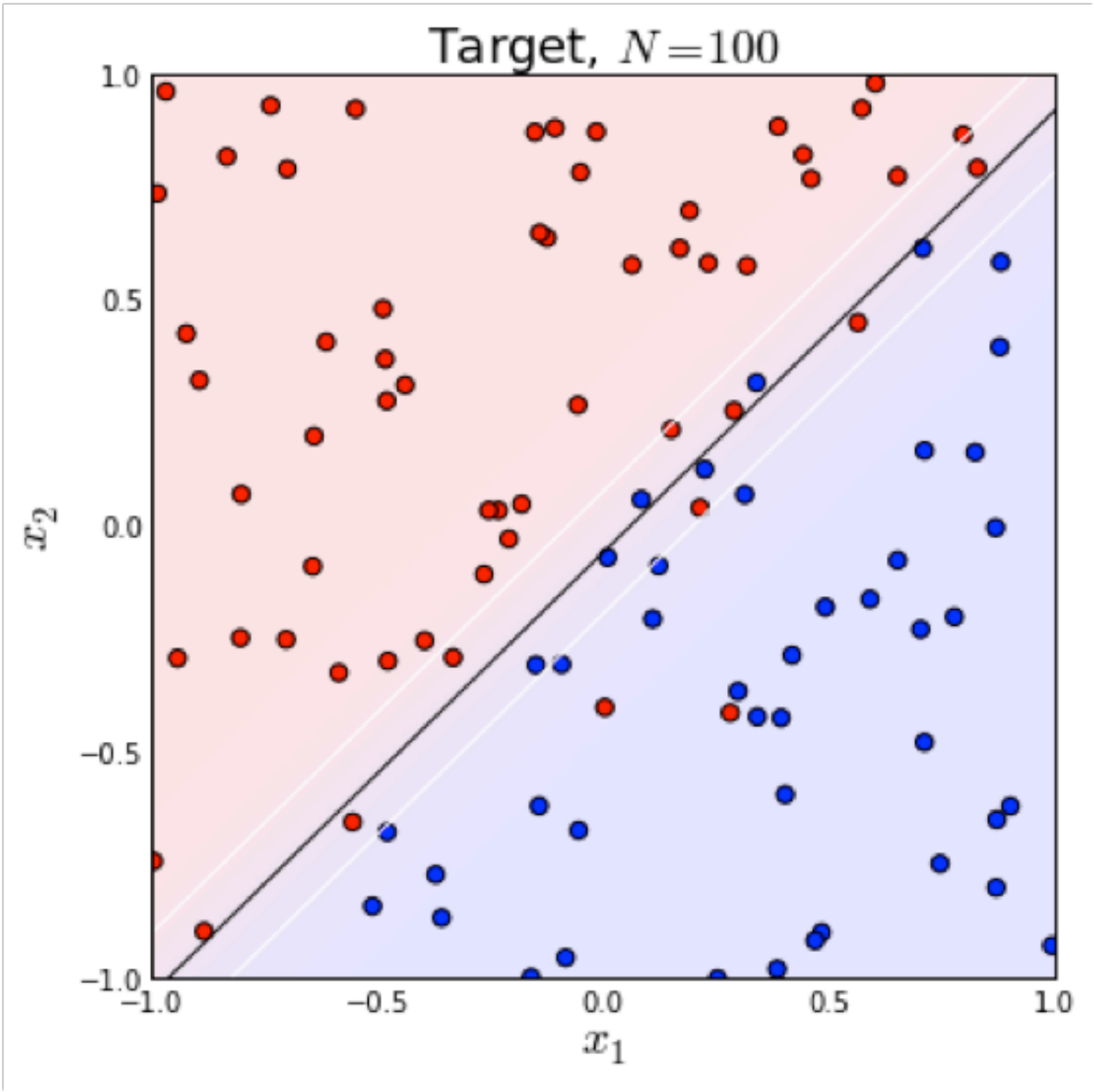
高的一个点。要将其熟练推导。

3.3 分类边界

知道如何求解参数后，我们看一下模型得到的最后结果是什么样的。
假设我们的决策函数为：

$$y^* = 1, \ \ if \ P(y = 1 \,|\, x) > 0.5$$

选择0.5作为阈值是一个一般的做法，实际应用时特定的情况可以选择不同阈值，如果对正例的判别准确性要求高，可以选择阈值大一些，对正例的召回要求高，则可以选择阈值小一些。
很容易看出，当$\theta^T x > 0$时，$y = 1$，否则$y = 0$。$\theta^T x = 0$是模型隐含的分类平面（在高维空间中，我们说是超平面）。所以说逻辑回归本质上是一个线性模型，但是这不意味着只有线性可分的数据能通过LR求解，实际上，我们可以通过特征变换的方式把低维空间转换到高维空间（kernel trick），而在低维空间不可分的数据，到高维空间中线性可分的几率会高一些。下面两个图的对比说明了线性分类曲线和非线性曲线（通过特征映射）。



左图是一个线性可分的数据集，右图在原始空间中线性不可分，但是在特征转换

$[x_1, x_2] => [x_1, x_2, x_1^2, x_2^2, x_1x_2]$后的空间是线性可分的，对应的原始空间中分类边界为一条椭圆曲线。

不过，通常使用的kernel都是隐式的，也就是找不到显式地把数据从低维映射到高维的函数，而只能计算高维空间中的数据点的内积。在这种情况下，logistic regression模型就不能再表示成$w^Tx + b$的形式（原始形式primal form），而只能表示成$\sum_i a_i < x_i, x > + b$的形式（对偶形式dual form）。忽略b，则原始形式的模型蚕食只有w，只需要一个数据点那么多的存储量；而对偶形式的模型不仅需要存储各个a_i，还要存储训练数据x_i本身，这个存储量就大了。

SVM也具有原始形式和对偶形式，相比之下，SVM的对偶形式是稀疏的，即只有支持向量的a_i才非零，才需要存储相应的x_i，所以，在非线性可分的情况下，SVM用的更多。

四、延伸

4. 1 生成模型与判别模型

逻辑回归是一种判别模型，表现为直接对条件概率$P(y|x)$建模，而不关心背后的数据分布$P(x, y)$。而高斯贝叶斯（Gaussian Naive Bayes）是一种生成模型，先对数据的联合分布建模，再通过贝叶斯公式来计算属于各个类别的后验概率，即：

$$p(y|x) = \frac{P(x|y)P(y)}{\sum P(x|y)P(y)}$$

通常假设$P(x|y)$是高斯分布，$P(y)$是多项式分布，相应的参数可以通过最大似然估计得到。如果我们考虑二分类问题，通过简单的变化可以得到：

$$log\frac{P(y=1|x)}{P(y=0|x)} = log\frac{P(x|y=1)}{P(x|y=0)} + log\frac{P(y=1)}{P(y=0)} = -\frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{(x-\mu_0)^2}{2\sigma_0^2} + \theta_0$$

如果$\sigma_1 = \sigma_0$，二次项会抵消，我们得到一个简单的线性关系：

$$log\frac{P(y=1|x)}{P(y=0|x)} = \theta^Tx$$

上式进一步可以得到：

$$P(y=1|x) = \frac{e^{\theta^Tx}}{1+e^{e^Tx}} = \frac{1}{1+e^{-\theta^Tx}}$$

可以看到，这个概率和逻辑回归中的形式是一样的，这种情况下高斯贝叶斯和LR会学习到同一个模型。实际上，在更一般的假设（P(x|y)的分布属于指数分布族）下，我们都可以得到类似的结

论。

4.2 多分类

如果y不是在$[0, 1]$中取值，而是在K个类别中取值，这时问题就变为一个多分类问题。有两种方式可以出处理该类问题：一种是我们对每个类别训练一个二元分类器（One-vs-all），当K个类别不是互斥的时候，比如用户会购买哪种品类，这种方法是合适的。如果K个类别是互斥的，即 $y = i$ 的时候意味着 y 不能取其他的值，比如用户的年龄段，这种情况下 Softmax 回归更合适一些。Softmax 回归是直接对逻辑回归在多分类的推广，相应的模型也可以叫做多元逻辑回归（Multinomial Logistic Regression）。模型通过 softmax 函数来对概率建模，具体形式如下：

$$P(y = i \mid x, \theta) = \frac{e^{\theta_i^T x}}{\sum_j^K e^{\theta_j^T x}}$$

而决策函数为：

$$y^* = argmax_i P(y = i \mid x, \theta)$$

对于的损失函数为

$$J(\theta) = -\frac{1}{N}\sum_i^N \sum_j^K P(y_i = j) log \frac{e^{\theta_i^T x}}{\sum e^{\theta_k^T x}}$$

类似的，我们也可以通过梯度下降或其他高阶方法来求解该问题，这里不再赘述。

4.3 应用

这里以预测用户对品类的购买偏好为例，介绍一下美团是如何用逻辑回归解决工作中问题的。该问题可以转换为预测用户在未来某个时间段是否会购买某个品类，如果把会购买标记为1，不会购买标记为0，就转换为一个二分类问题。我们用到的特征包括用户在美团的浏览，购买等历史信息，见下表：

类别	特征
用户	购买频次，浏览频次，时间，地理位置 ...
品类	销量，购买用户，浏览用户 ...
交叉	购买频次，浏览频次，购买间隔 ...

其中提取的特征的时间跨度为30天，标签为2天。生成的训练数据大约在7000万量级（美团一个月有过行为的用户），我们人工把相似的小品类聚合起来，最后有18个较为典型的品类集合。如果用户在给定的时间内购买某一品类集合，就作为正例。有了训练数据后，使用Spark版的LR算法对每个品类训练一个二分类模型，迭代次数设为100次的话模型训练需要40分钟左右，平均每个模型2分钟，测试集上的AUC也大多在0.8以上。训练好的模型会保存下来，用于预测在各个品类上的购买概率。预测的结果则会用于推荐等场景。

由于不同品类之间正负例分布不同，有些品类正负例分布很不均衡，我们还尝试了不同的采样方法，最终目标是提高下单率等线上指标。经过一些参数调优，品类偏好特征为推荐和排序带来了超过1%的下单率提升。

此外，由于LR模型的简单高效，易于实现，可以为后续模型优化提供一个不错的baseline，我们在排序等服务中也使用了LR模型。

逻辑回归的数学模型和求解都相对比较简洁，实现相对简单。通过对特征做离散化和其他映射，逻辑回归也可以处理非线性问题，是一个非常强大的分类器。因此在实际应用中，当我们能够拿到许多低层次的特征时，可以考虑使用逻辑回归来解决我们的问题。

4.4 LR与SVM

两种方法都是常见的分类算法，从目标函数来看，区别在于逻辑回归采用的是logistical loss，svm采用的是hinge loss。这两个损失函数的目的都是增加对分类影响较大的数据点的权重，减少与分类关系较小的数据点的权重。SVM的处理方法是只考虑support vectors，也就是和分类最相关的少数点，去学习分类器。而逻辑回归通过非线性映射，大大减小了离分类平面较远的点的权重，相对提升了与分类最相关的数据点的权重。两者的根本目的都是一样的。此外，根据需要，两个方法都可以增加不同的正则化项，如l1,l2等等。所以在很多实验中，两种算法的结果是很接近的。但是逻辑回归相对来说模型更简单，好理解，实现起来，特别是大规模线性分类时比较方便。而SVM的理解和优化相对来说复杂一些。但是SVM的理论基础更加牢固，有一套结构化风险最小化的理论基础，虽然一般使用的人不太会去关注。还有很重要的一点，SVM转化为对偶问题后，分类只需要计算与少数几个支持向量的距离，这个在进行复杂核函数计算时优势很明显，能够大大简化模型和计算量。

两者对异常的敏感度也不一样。同样的线性分类情况下，如果异常点较多的话，无法剔除，首先LR，LR中每个样本都是有贡献的，最大似然后会自动压制异常的贡献，SVM+软间隔对异常还是比较敏感，因为其训练只需要支持向量，有效样本本来就不高，一旦被干扰，预测结果难以预料。

参考资料

对线性回归，logistic回归和一般回归的认识

Logistic Regression 模型简介