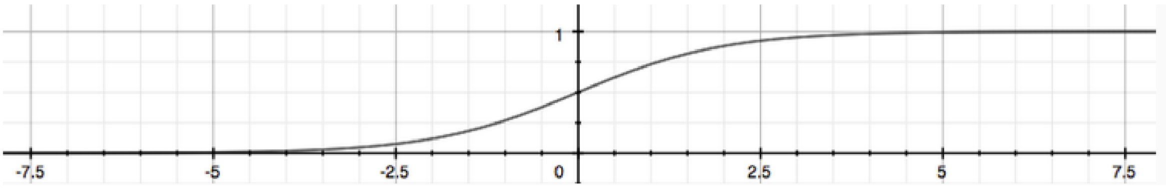
# 逻辑斯蒂回归

1. 回归问题把数据拟合到一条直线（多项式回归是曲线），逻辑斯蒂回归则是在原有回归的假设函数的基础上，利用Sigmoid函数，将数据映射为两部分，逼近于0和逼近于1。

$$h_\theta(x) = g(\theta^T x)$$

$$z = \theta^T x$$

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




g函数的自变量大于0时，分类器结果逼近于1，反之亦然。而g函数的自变量的参数确定后就确定了一个超平面（多项式是曲面），将输入数据切割为两部分，这个超平面叫做决策边界（Decision Boundary）

2. 损失函数

首先想到的损失函数是

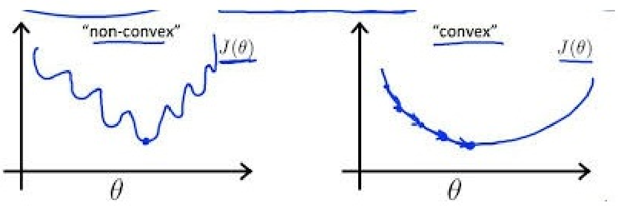
$$J = 1/m * \sum_{i=1}^{m} (h(x_i) - y_i)^2$$

即：

y=0：$J = 1/m * \sum_{1}^{\cdots} (h(x) - 0)^2$

y=1: $J = 1/m * \sum_{1}^{m} (h(x) - 1)^2$

但是这样的两个曲线不都是凸函数，无法求全局最优解



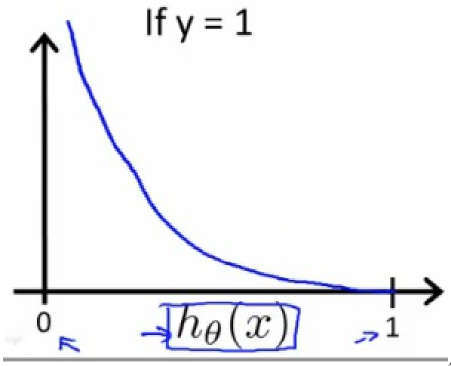

于是想到区找一个函数将h(x)和y进行映射

、

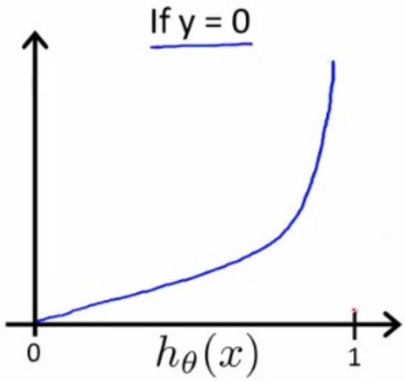
$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x)) \qquad \text{if } y = 1$$

$$\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x)) \qquad \text{if } y = 0$$




然后利用梯度下降方法求解即可




3.最大似然估计

确定模型以后，不寻找损失函数，而是直接使用最大似然方法进行参数估计，这和上面的方法是一个道理，同样的数学原理，两种解法而已

设：

$$P(Y=1|x)=\pi(x), P(Y=0|x)=1-\pi(x)$$

似然函数：

$$L(w)=\prod[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i}$$

概率函数P(Y = 1 | x)为：

$$P(Y=1|x)==\frac{e^{w\cdot x}}{1+e^{w\cdot x}}$$

$$P(Y=0|x)==\frac{1}{1+e^{w\cdot x}}$$

设Y=1时的概率为p，则这个时间的几率为p/(1-p),对数几率为log(p/1-p),即：log(p/1-p) = wx

这就是说逻辑斯谛回归是用线性函数拟合一个对数几率。

模型参数估计：

给定数据集的情况下，似然函数为：

$$\prod_{i=1}^{N}(\pi(x_i))^{y_i}(1-\pi(x_i))^{1-y_i}$$

对数似然函数为：

$$L(w)=\sum_{i=1}^{N}(y_i log\pi\,(x_i)+(1-y_i)log\pi\,(1-x_i)$$

$$=\sum_{i=1}^{N}(y_i(log\pi(x_i)-log\pi(1-x_i))+log(1-\pi(x_i)))$$

$$=\sum_{i=1}^{N}(y_i(w\cdot x)-log(1+exp(w\cdot x)))$$

这里要注意的是，根据我们似然函数关心的是y的取值，也就是y取不同值时候的概率，而我们的模型中，y出现的概率是由P(y|x)产生的。

对其利用梯度下降求最大值即可得到w的估计值。

关于逻辑斯蒂回归的另一个角度的解释参见博客http://www.hankcs.com/ml/the-logistic-regression-and-the-maximum-entropy-model.html

3. 核逻辑回归

首先介绍一个定理：

## Kernel Logistic Regression

我们首先介绍表示定理（Representer Theorem）：

claim: for any L2-regularized linear model

$$\min_{\mathbf{w}} \quad \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + \frac{1}{N}\sum_{n=1}^{N} \operatorname{err}(y_n, \mathbf{w}^T\mathbf{z}_n)$$

optimal $\mathbf{w}_* = \sum_{n=1}^{N} \beta_n \mathbf{z}_n$.

即解任意一个L2 Regularization的问题，其最佳$w_*$都可以用$\beta_n$与$Z_n$线性组合得到。

也就是说，只要是L2正则的问题，它的最优解都可以表示成输入的线性组合的形式，这样就可以在LR中引入核函数。

原始的LR的损失函数：

solving L2-regularized logistic regression

$$\min_{\mathbf{w}} \quad \frac{\lambda}{N}\mathbf{w}^T\mathbf{w} + \frac{1}{N}\sum_{n=1}^{N} \log\left(1 + \exp\left(-y_n\mathbf{w}^T\mathbf{z}_n\right)\right)$$

yields optimal solution $\mathbf{w}_* = \sum_{n=1}^{N} \beta_n \mathbf{z}_n$

将所有的w替换成输入z的线性组合：

$$\min_{\beta} \frac{\lambda}{N}\sum_{n=1}^{N}\sum_{m=1}^{N} \beta_n\beta_m K(\mathbf{x}_n, \mathbf{x}_m) + \frac{1}{N}\sum_{n=1}^{N} \log\left(1 + \exp\left(-y_n\sum_{m=1}^{N} \beta_m K(\mathbf{x}_m, \mathbf{x}_n)\right)\right)$$

这样，对β求解救可以得到最优解w，求解β可以使用梯度下降。

# 七月在线部分笔记

1. 优点：

概率输出，可做rank，可解释性强，速度快，特征工程后效果好，容易添加特征

样本量大怎么办：

特征离散化，转one-hot，只有0 1 的时候处理特别快

连续值做scaling

采样

样本不均衡：

上采样，下采样

修改loss function，修改不同类别的权重

采样后，做预测时要调回来

离散化好处：

把连续特征分成了好几个区间，带来一定的非线性

稀疏化，0 1 做矩阵乘法速度更快

SVM与LR的区别

1. 损失函数不同
2. SVM仅与支持向量有关
3. SVM是有约束的
4. LR的可解释性更强
5. LR可以给出概率结果，可以用做rank
6. SVM自带正则，泛化能力强一些
7. 使用核函数的情况下，SVM更好更快