逻辑回归

机器学习笔记 create by siwanghu v1.0

逻辑回归，机器学习常用二分类算法，它将数据拟合到一个 logit 函数(或者叫做 logistic 函数)中，从而能够完成对事件发生的概率进行预测。

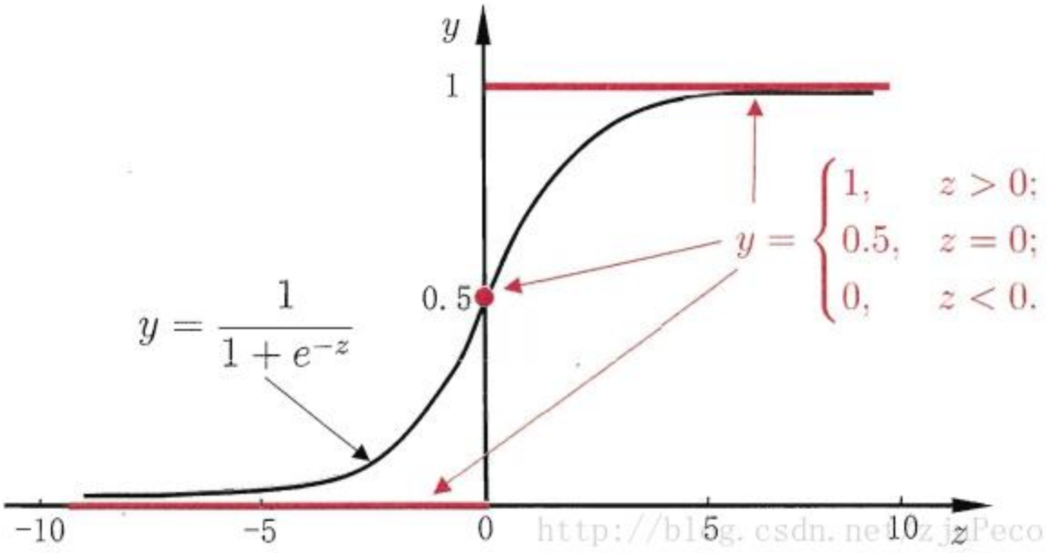
常用逻辑函数(sigmoid 函数)：

$$y(\mathrm{x}) = \frac{1}{1+\mathrm{e}^{-\mathrm{x}}}$$

性质：

$$y(x)' = y(x)(1-y(x))$$

图像：



假设样本为：$(x^1, x^2, x^i \ldots, x^n)$，对应的标签分别为：$(y^1, y^2, y^i \ldots, y^n)$

则：

$$z = w * x^i + b$$

w 为权重，b 为偏置，带入 sigmoid 函数，将样本映射到 sigmoid 函数：

$$y_*^i = \frac{1}{1+\mathrm{e}^{-\mathrm{z}}}$$

定义二次损失函数：

$$L(w) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(y_*^i - y^i\right)^2$$

优化目标是确定w的值，使得$L(w)$最小，m 为训练批次数据量大小，下面假设 m 为 1，则损失函数为：

$$L(w)=\frac{1}{2}\left(y_*^i-y^i\right)^2$$

所以：

$$\begin{aligned}\frac{\partial L(w)}{\partial w}&=\left(y_*^i-y^i\right)\left(y_*^i\right)'x^i\\&=\left(y_*^i-y^i\right)\left(\frac{1}{1+\mathrm{e}^{-\mathrm{z}}}\right)'x^i\end{aligned}$$

所以：

$$\begin{aligned}\frac{\partial L(w)}{\partial b}&=\left(y_*^i-y^i\right)\left(y_*^i\right)'\\&=\left(y_*^i-y^i\right)\left(\frac{1}{1+\mathrm{e}^{-\mathrm{z}}}\right)'\end{aligned}$$

使用梯度下降法更新w和b：

$$w=w-\alpha\frac{\partial L(w)}{\partial w}$$

$$b=b-\alpha\frac{\partial L(w)}{\partial b}$$

采用二次函数作为损失函数来优化模型时，由于$\frac{\partial L(w)}{\partial w}$和$\frac{\partial L(w)}{\partial b}$的取值与 sigmoid 的导数相关，sigmoid 函数的导数在函数两边是非常的小，也就是说函数的梯度会接近于 0，容易产生梯度消失，使得学习异常艰难，下面采用交叉熵函数作为损失函数。

交叉熵：

$$h(p,q)=-\sum_{i=1}^{n}p(i)log(q(i))$$

交叉熵函数定义分布律$p(i)$与分布律$q(i)$相似的程度

交叉熵损失函数：

$$L(w)=-\frac{1}{m}\left[\sum_{i=1}^{m}y^i log(y_*^i)+\left(1-y^i\right)\log(1-y_*^i)\right]$$

m 为训练批次数据量大小，下面假设 m 为 1，则损失函数为：

$$L(w)=-[y^i log(y_*^i)+\left(1-y^i\right)\log(1-y_*^i)]$$

所以：

$$\frac{\partial L(w)}{\partial w}=-\frac{y^i\left(y_*^i\right)'x^i}{y_*^i}+\frac{(1-y^i)\left(y_*^i\right)'x^i}{1-y_*^i}$$

$$\frac{\partial L(w)}{\partial w}=\frac{\left(1-y^{i}\right)\left(y_{*}^{i}\right)' x^{i} y_{*}^{i}-(1-y_{*}^{i}) y^{i}\left(y_{*}^{i}\right)' x^{i}}{y_{*}^{i}(1-y_{*}^{i})}$$

又因为 sigmoid 的性质：

$$y(x)'=y(x)(1-y(x))$$

所以：

$$\frac{\partial L(w)}{\partial w}=x^{i}\text{（}y_{*}^{i}-y^{i}\text{）}$$

同理：

$$\frac{\partial L(w)}{\partial b}=y_{*}^{i}-y^{i}$$

使用梯度下降法更新w和b：

$$w=w-\alpha\frac{\partial L(w)}{\partial w}$$

$$b=b-\alpha\frac{\partial L(w)}{\partial b}$$

采用交叉熵损失函数，梯度与 sigmoid 函数导数无关，不会出现梯度消失的问题，而且因为梯度与输出值与实际值的差成正比，也就是说输出值与实际值偏差越大，梯度越大，参数调整的越快，输出值与实际值偏差越小，参数调整的越慢，符合预期。可见使用交叉熵损失函数比使用二次损失函数效果更好。