逻辑回归 (Logistic regression)

Guangyao Zhao

2022-01-25

Contents

二分类问题

逻辑回归的原理是用逻辑函数把线性回归的结果 $(-\infty, +\infty)$ 映射到 $(0, 1)$。逻辑函数 (sigmoid function):

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

对 sigmoid 函数求导可得 (求解的时候用):

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

对预测结果定义如下:

- 正例: $p(y=1|x) = g(w^{\mathrm{T}}x) = \frac{1}{1+e^{-w^{\mathrm{T}}x}}$

- 反例: $p(y=0|x)=1-g(w^{\mathrm{T}}x)=\frac{e^{-w^{\mathrm{T}}x}}{1+e^{-w^{\mathrm{T}}x}}$

将两个式子合并成一个, 即:

$$p(y|x)=p_1^y p_0^{1-y}$$

利用最大似然函数:

$$\begin{aligned}
w^* &= \underset{w}{\operatorname{argmax}} \ln p(Y|X) \\
&= \underset{w}{\operatorname{argmax}} \ln \prod_{i=1}^{n} p(y_i|x_i) \\
&= \underset{w}{\operatorname{argmax}} \sum_{i=1}^{n} \left(y_i \ln p_1 + (1-y_i) \ln p_0\right) \\
&= \underset{w}{\operatorname{argmax}} \sum_{i=1}^{n} \left(y_i \ln g(w^{\mathrm{T}}x_i) + (1-y_i) \ln\left(1-g(w^{\mathrm{T}}x_i)\right)\right)
\end{aligned}$$

所以, 逻辑回归的损失函数可定义为:

$$\mathrm{J}(w) = -\frac{1}{n}\sum_{i=1}^{n} \left(y_i \ln g(w^{\mathrm{T}}x_i) + (1-y_i) \ln\left(1-g(w^{\mathrm{T}}x_i)\right)\right)$$

其中 $g(z)=\frac{1}{1+e^{-z}}$。从上式可看出, 其实逻辑回归的表达式是交叉熵 (cross entropy)。

对损失函数进行求导:

$$\begin{aligned}
\frac{\partial \mathrm{J}(w)}{\partial w_j} &= -\frac{1}{n}\sum_{i=1}^{n} \left[y_i \frac{1}{g(z)} - (1-y_i)\frac{1}{1-g(z)}\right] \frac{\partial g(z)}{\partial w_j} \\
&= -\frac{1}{n}\sum_{i=1}^{n} \left[\frac{y_i - g(z)}{g(z)\left(1-g(z)\right)}\right] \frac{\partial g(z)}{\partial w_j} \\
&= -\frac{1}{n}\sum_{i=1}^{n} \left(y_i - g(z)\right) x_{ij} \\
&= \frac{1}{n}\sum_{i=1}^{n} \left(g(z) - y_i\right) x_{ij}
\end{aligned}$$

由上式子和线性回归 w 更新相比, 仅仅是多了一个 sigmoid 变换。

非矩阵形式权值更新如下:

$$w_j := w_j - \alpha \frac{1}{n}\sum_{i=1}^{n} \left(g(w^{\mathrm{T}}x_i) - y_i\right) x_{ij}$$

矩阵形式权值更新如下：

$$w := w - \alpha \frac{1}{n} \mathbf{X}^{\mathrm{T}} (g(\mathbf{X} w) - y)$$

多分类问题

非矩阵化

多分类问题就要用到 softmax 函数，在人工神经网络一节中已经推导得出结论：

$$\begin{aligned} \delta_q &= \frac{\partial L}{\partial z_q} \\ &= \begin{cases} y_q a_p, \, p \neq q \\ y_q (a_p - 1), \, p = q \end{cases} \end{aligned}$$

特别注意，在 softmax 中系数 w 为一个矩阵，形状为（分类个数 `k`，特征 `m`）。w 更新如下：

$$\begin{aligned} \frac{\partial L}{w_{ij}} &= \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} \\ &= \delta_j x_j \end{aligned}$$

其中 i 为 w 的行数，j 为输出层的神经元序数。

矩阵化

$$\begin{aligned} \frac{\partial L}{w_{ij}} &= \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ij}} \\ &= \delta_j x_j \end{aligned}$$

矩阵化表示：

1. `x.shape=(n,m)`
2. `w.shape=(k,m)`，$z = w x^{\mathrm{T}}$ 的形状为 `(k,n)`
3. `z.shape=(k,n)`，损失求平均 `z.shape=(k,1)`

更新公式：

$$w := w - \delta x^{\mathrm{T}}$$