人工智能之机器学习

Logistic-Softmax

主讲人：李老师

课程内容

- odds（几率）
- Logistic回归算法
- Softmax回归算法

Logistic回归

- Logistic/sigmoid函数 $p = h_\theta(x) = g\left(\theta^T x\right) = \dfrac{1}{1+e^{-\theta^T x}}$

$$y = \begin{cases} 1 \\ 0 \end{cases}$$

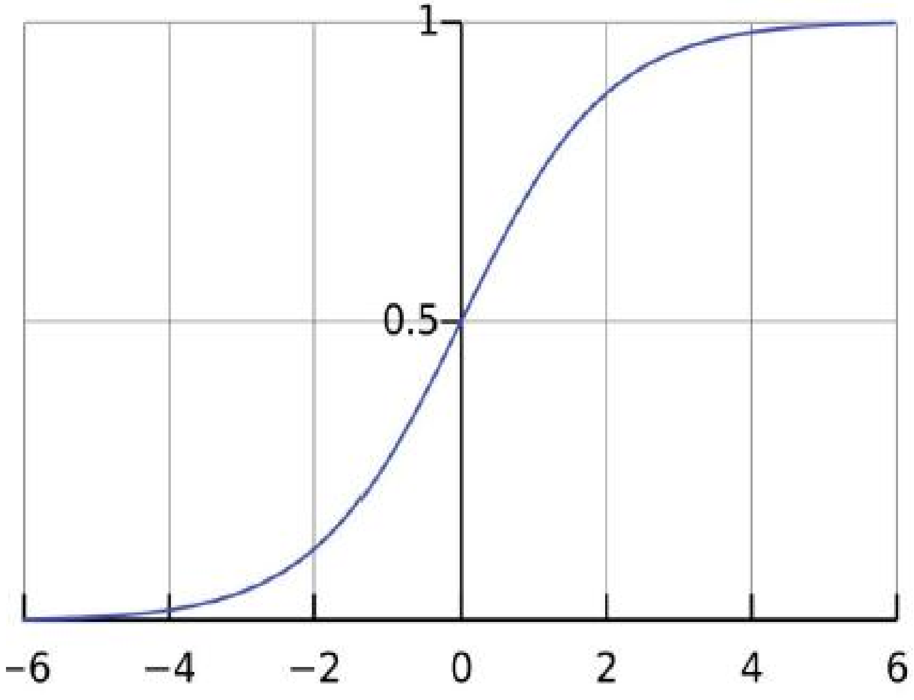
$$\hat{y} = \begin{cases} 1, p > threshold \\ 0, p \le threshold \end{cases}$$

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

$$g'(z) = \left(\frac{1}{1+e^{-z}}\right)' = \frac{e^{-z}}{\left(1+e^{-z}\right)^2}$$

$$= \frac{1}{1+e^{-z}} \cdot \frac{e^{-z}}{1+e^{-z}} = \frac{1}{1+e^{-z}} \cdot \left(1 - \frac{1}{1+e^{-z}}\right)$$

$$= g(z) \cdot \left(1 - g(z)\right)$$



Logistic回归及似然函数

- 假设: $$P(y=1\mid x;\theta)=h_\theta(x)$$ $$P(y=0\mid x;\theta)=1-h_\theta(x)$$

	y=1	y=0
p(y|x)	p	1-p

$$P(y\mid x;\theta)=\left(h_\theta(x)\right)^{y}\left(1-h_\theta(x)\right)^{(1-y)}$$

- 似然函数: $$L(\theta)=p(\vec{y}\mid X;\theta)=\prod_{i=1}^{m}p\left(y^{(i)}\mid x^{(i)};\theta\right)$$

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

- 对数似然函数:

$$\ell(\theta)=\ln L(\theta)=\sum_{i=1}^{m}\left(y^{(i)}\ln h_\theta\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right)$$

最大似然/极大似然函数的随机梯度

- 对数似然函数：$\ell(\theta)=\ln L(\theta)=\sum_{i=1}^{m}\left(y^{(i)}\ln h_\theta\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right)$

$$\frac{\partial \ell(\theta)}{\partial \theta_j}=\sum_{i=1}^{m}\left(\frac{y^{(i)}}{h_\theta\left(x^{(i)}\right)}-\frac{1-y^{(i)}}{1-h_\theta\left(x^{(i)}\right)}\right)\cdot\frac{\partial h_\theta\left(x^{(i)}\right)}{\partial \theta_j}$$

$$=\sum_{i=1}^{m}\left(\frac{y^{(i)}}{g\left(\theta^T x^{(i)}\right)}-\frac{1-y^{(i)}}{1-g\left(\theta^T x^{(i)}\right)}\right)\cdot\frac{\partial g\left(\theta^T x^{(i)}\right)}{\partial \theta_j}$$

$$=\sum_{i=1}^{m}\left(\frac{y^{(i)}}{g\left(\theta^T x^{(i)}\right)}-\frac{1-y^{(i)}}{1-g\left(\theta^T x^{(i)}\right)}\right)\cdot g\left(\theta^T x^{(i)}\right)\left(1-g\left(\theta^T x^{(i)}\right)\right)\cdot\frac{\partial \theta^T x^{(i)}}{\partial \theta_j}$$

$$=\sum_{i=1}^{m}\left(y^{(i)}\left(1-g\left(\theta^T x^{(i)}\right)\right)-\left(1-y^{(i)}\right)g\left(\theta^T x^{(i)}\right)\right)\cdot x_j^{(i)}=\sum_{i=1}^{m}\left(y^{(i)}-g\left(\theta^T x^{(i)}\right)\right)\cdot x_j^{(i)}$$

极大似然估计与Logistic回归目标函数

- 由于在极大似然估计中，当似然函数最大的时候模型最优；而在机器学习领域中，目标函数最小的时候，模型最优；故可以使用似然函数乘以-1的结果作为目标函数。

$$\ell(\theta)=\ln L(\theta)=\sum_{i=1}^{m}\left(y^{(i)}\ln h_\theta\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right)$$

$$loss=-\ell(\theta)=-\sum_{i=1}^{m}\left(y^{(i)}\ln h_\theta\left(x^{(i)}\right)+\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right)$$

$$=\sum_{i=1}^{m}\left[-y^{(i)}\ln\left(h_\theta\left(x^{(i)}\right)\right)-\left(1-y^{(i)}\right)\ln\left(1-h_\theta\left(x^{(i)}\right)\right)\right]$$

θ参数求解

- Logistic回归θ参数的求解过程为(类似梯度下降方法):

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta\left(x^{(i)}\right) \right) x_j^{(i)}$$

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

Softmax回归

- softmax回归是logistic回归的一般化，适用于K分类的问题，针对于每个类别都有一个参数向量θ，第k类的参数为向量θ_k，组成的二维矩阵为θ_{k*n}；
- softmax函数的本质就是将一个K维的任意实数向量压缩（映射）成另一个K维的实数向量，其中向量中的每个元素取值都介于（0，1）之间。
- softmax回归概率函数为：

$$p(y=k \mid x;\theta)=\frac{e^{\theta_k^T x}}{\sum_{l=1}^{K} e^{\theta_l^T x}}, k=1,2\cdots,K$$

Softmax算法原理

$$p(y=k\mid x;\theta)=\frac{e^{\theta_k^T x}}{\sum_{l=1}^{K} e^{\theta_l^T x}},k=1,2\cdots,K$$

$$h_\theta(x)=\begin{bmatrix} p\left(y^{(i)}=1\mid x^{(i)};\theta\right) \\ p\left(y^{(i)}=2\mid x^{(i)};\theta\right) \\ \cdots \\ p\left(y^{(i)}=k\mid x^{(i)};\theta\right) \end{bmatrix}=\frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}}\begin{bmatrix} e^{\theta_1^T x} \\ e^{\theta_2^T x} \\ \cdots \\ e^{\theta_k^T x} \end{bmatrix} \Rightarrow \theta=\begin{bmatrix} \theta_{11} & \theta_{12} & \dots & \theta_{1n} \\ \theta_{21} & \theta_{22} & \dots & \theta_{2n} \\ \dots & \dots & \dots & \dots \\ \theta_{k1} & \theta_{k2} & \dots & \theta_{kn} \end{bmatrix}$$

Softmax算法损失函数

$$p\left(y=k \mid x;\theta\right)=\frac{e^{\theta_k^T x}}{\sum_{l=1}^{K} e^{\theta_l^T x}}, k=1,2\cdots,K$$

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} I\left(y^{(i)}=j\right)\ln\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)$$

$$I\left(y^{(i)}=j\right)=\begin{cases}1, & y^{(i)}=j\\ 0, & y^{(i)}\neq j\end{cases}$$

$$p\left(y=k \mid x;\theta\right)=\frac{e^{\theta_k^T x}}{\sum_{l=1}^{K} e^{\theta_l^T x}}, k=1,2\cdots,K$$

Softmax算法梯度下降法求解

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k} I\left(y^{(i)} = j\right)\ln\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)$$

$$I\left(y^{(i)} = j\right) = \begin{cases} 1, & y^{(i)} = j \\ 0, & y^{(i)} \neq j \end{cases}$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} - I\left(y^{(i)} = j\right)\ln\left(\frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)$$

$$= \frac{\partial}{\partial \theta_j} - I\left(y^{(i)} = j\right)\left(\theta_j^T x^{(i)} - \ln\left(\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}\right)\right)$$

$$= -I\left(y^{(i)} = j\right)\left(1 - \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)x^{(i)}$$

Softmax算法梯度下降法求解

$$\frac{\partial}{\partial \theta_j} J(\theta) = -I\left(y^{(i)} = j\right)\left(1 - \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right)x^{(i)}$$

$$\theta_j = \theta_j + \alpha \sum_{i=1}^{m} I\left(y^{(i)} = j\right)\left(1 - p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right)\right)x^{(i)}$$

$$\theta_j = \theta_j + \alpha I\left(y^{(i)} = j\right)\left(1 - p\left(y^{(i)} = j \middle| x^{(i)}; \theta\right)\right)x^{(i)}$$

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

总结

- 线性模型一般用于回归问题，Logistic和Softmax模型一般用于分类问题
- 求θ的主要方式是梯度下降算法，梯度下降算法是参数优化的重要手段，主要是SGD，适用于在线学习以及跳出局部极小值
- Logistic/Softmax回归是实践中解决分类问题的最重要的方法
- 广义线性模型对样本要求不必要服从正态分布、只需要服从指数分布簇(二项分布、泊松分布、伯努利分布、指数分布等)即可；广义线性模型的自变量可以是连续的也可以是离散的。