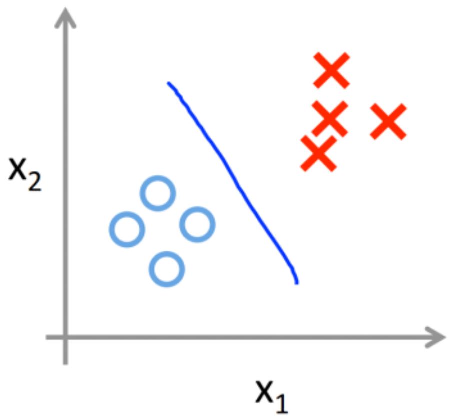
Class 2 逻辑回归

Sunday, October 14, 2018   10:01 AM

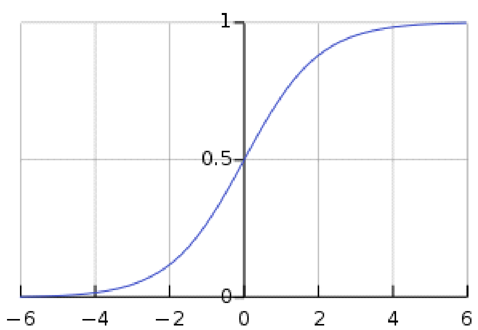
• 简介:

- 逻辑回归是一种二分类模型;
- 逻辑回归是一种线性分类模型，它有一个线性决策边界（超平面），但用一个非线性激活函数（Sigmoid）来模拟后验概率。



• 模型假设:

- **Sigmoid函数**



$$\delta(z) = \frac{1}{1+e^{-z}} \qquad \frac{d\delta(z)}{dz} = \delta(z)\big(1-\delta(z)\big)$$

- ❖ 当 z = 0 时， $\delta(z) = 0.5$;
- ❖ 当 z > 0 时，$\delta(z) > 0.5$, 当 z 越来越大时，$\delta(z)$无限接近于 1；
- ❖ 当 z < 0 时，$\delta(z) < 0.5$, 当 z 越来越小时，$\delta(z)$无限接近于 0 。

- **假设**

$$p(y=1|x;\theta) = h_\theta(x) = \delta(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

$$p(y=0\,|x;\theta) = 1 - h_\theta(x)$$

上述公示表示:在参数 θ 条件下，样本 x 是 y=1(y=0) 类别的概率。

将以上两类情况合并可得:

$$p(y\,|x;\theta) = (h_\theta(x))^y(1-h_\theta(x))^{(1-y)} = (\frac{1}{1+e^{-\theta^T x}}\ )^y(1-\frac{1}{1+e^{-\theta^T x}}\ )^{(1-y)}$$

- ## 学习算法:

  - **(条件)似然函数**

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{N} p\left(y^{(i)} \middle| x^{(i)};\theta\right) \\ &= \prod_{i=1}^{N} \left(h_\theta\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1-h_\theta\left(x^{(i)}\right)\right)^{(1-y^{(i)})} \\ &= \prod_{i=1}^{N} \left(\frac{1}{1+e^{-\theta^T x^{(i)}}}\right)^{y^{(i)}} \left(1-\frac{1}{1+e^{-\theta^T x^{(i)}}}\right)^{(1-y^{(i)})} \end{aligned}$$

  - **最大似然估计**

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

- ## 优化问题及方法:

  - **最优化问题**

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

  - **成本函数**

最优化问题为求解最大似然估计，可转化为求其相反数的最小值，即成本函数为

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

  - **优化方法**

    ❖ 梯度下降(GD)

- ❖ 随机梯度下降(SGD)
- ❖ 牛顿法
- ❖ ...

• 优化:

• 梯度下降(GD)

❖ 梯度计算:

$$\begin{aligned}\frac{dl(\theta)}{d\theta} &= \sum_{i=1}^{N}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})}-(1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right)\frac{\partial}{\partial\theta}h_\theta(x^{(i)}) \\ &= \sum_{i=1}^{N}\left(y^{(i)}\frac{1}{h_\theta(x^{(i)})}-(1-y^{(i)})\frac{1}{1-h_\theta(x^{(i)})}\right)h_\theta(x^{(i)})\left(1-h_\theta(x^{(i)})\right)\frac{\partial}{\partial\theta}\theta^T x^{(i)} \\ &= \sum_{i=1}^{N}\left(y^{(i)}\left(1-h_\theta(x^{(i)})\right)-(1-y^{(i)})h_\theta(x^{(i)})\right)x^{(i)} \\ &= \sum_{i=1}^{N}\left(y^{(i)}-h_\theta(x^{(i)})\right)x^{(i)}\end{aligned}$$

❖ 梯度优化:

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

其中

$$\boldsymbol{h}(\boldsymbol{\theta}) = \frac{\mathbf{1}}{\mathbf{1}+\boldsymbol{e}^{-\boldsymbol{\theta}^T\boldsymbol{x}}} \quad \text{(向量形式)}$$

或

$$\boldsymbol{h}(\boldsymbol{\theta}) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1\mathbf{x}_1 + \boldsymbol{\theta}_1\mathbf{x}_1 + \boldsymbol{\theta}_2\mathbf{x}_2 + \cdots \quad \text{(标量形式)}$$

• 随机梯度下降(SGD)

- **随机梯度计算:**
  - 随机选择一个训练样本

$$(x, y)$$

  - 计算梯度

$$(y - h_\theta(x))x$$

  - 更新参数

$$\theta := \theta + \alpha(y - h_\theta(x))x$$

  - 重复...

- **梯度优化:**

$$\theta := \theta + \alpha(y - h_\theta(x))x$$

其中

$$\boldsymbol{h}(\boldsymbol{\theta}) = \frac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-\boldsymbol{\theta}^T \boldsymbol{x}}} \quad (\text{向量形式})$$

或

$$\boldsymbol{h}(\boldsymbol{\theta}) = \boldsymbol{\theta}_0 + \boldsymbol{\theta}_1 \mathbf{x}_1 + \boldsymbol{\theta}_1 \mathbf{x}_1 + \boldsymbol{\theta}_2 \mathbf{x}_2 + \cdots \ (\text{标量形式})$$

- **两种梯度算法的比较:**

Gradient Descent	批处理 梯度下降	每次更新 需计算所有样本	迭代速度慢 迭代次数多
Stochastic Gradient Descent	随机 梯度下降	每次更新 只计算某一个样本	迭代速度慢 迭代次数少

- ## 牛顿法

- 简介:

  - 牛顿法解决的问题

$$\arg\min f(\theta) \Leftrightarrow solve:\ \nabla f(\theta) = 0$$

  - 将f(θ)用泰勒公式二阶展开

$$f(\theta) = f(\theta^{(k)}) + \nabla f(\theta^{(k)})(\theta - \theta^{(k)}) + \frac{1}{2}\nabla^2 f(\theta^{(k)})(\theta - \theta^{(k)})^2$$

  - 求导并移项

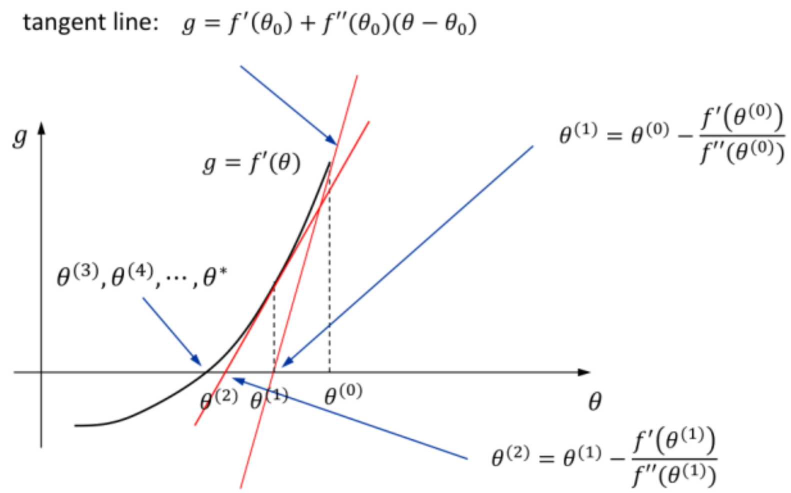
$$\nabla f(\theta) = 0$$

$$\theta = \theta^{(k)} - \nabla^2 f(\theta^{(k)})^{-1} \nabla f(\theta^{(k)})$$

  - 迭代过程

$$\theta^{(k+1)} = \theta^{(k)} - \underbrace{\boxed{\nabla^2 f(\theta^{(k)})}}_{\text{Hessian Matrix}}{}^{-1} \nabla f(\theta^{(k)})$$

  - 图示



- **逻辑回归的牛顿法优化：**

  - 优化问题

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

  - 梯度及海森矩阵

$$\Delta J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

$$H = \frac{1}{N}\sum_{i=1}^{N} h_\theta\left(x^{(i)}\right)^{\mathrm{T}}\left(1 - h_\theta\left(x^{(i)}\right)\right)x^{(i)}\left(x^{(i)}\right)^{\mathrm{T}}$$

  - 参数更新

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla J(\theta^{(t)})$$

- **牛顿法与梯度下降法的比较：**
  - 梯度下降法的迭代公式

$$\theta^{(t+1)} = \theta^{(t)} - \alpha\nabla J\left(\theta^{(t)}\right)$$

    其中α学习率
  - 牛顿法的迭代公式

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla J(\theta^{(t)})$$

与梯度下降法相比，牛顿法只是将学习率 α 替换成了 H^{-1}；

这可以理解为：

牛顿法根据代价函数的二阶导数信息，自动计算出了合适的学习率；因此有更快的迭代速度，而作为交换，牛顿法需要计算庞大的Hessian矩阵，矩阵的大小为 m x m，计算速度慢，消耗资源大。

- **牛顿法的矩阵实现：**
  - 对于每一个样本点，我们有一个特征向量 $\boldsymbol{x_i}$，这个向量的维度就是特征的个数。同时还有一个观测类别 $\boldsymbol{y_i}$，当 $\boldsymbol{y_i = 1}$ 的时候，该类的概率为 p，否则为 1-p 。因此，似然函数为

$$L(\beta_0,\beta)=\prod_{i=1}^{n}p(x_i)^{y_i}(1-p(x_i))^{1-y_i}$$

对数似然函数为

$$\begin{aligned}\ell(\beta_0,\beta)&=\sum_{i=1}^{n}y_i ln(p(x_i))+(1-y_i)ln(1-p(x_i))\\&=\sum_{i=1}^{n}y_i(ln\frac{p(x_i)}{1-p(x_i)})+ln(1-p(x_i))\\&=\sum_{i=1}^{n}y_i(\beta_0+x_i\beta)-ln(1+e^{\beta_0+x_i\beta})\end{aligned}$$

为了表示方便，统一将 β0, β 表示成 β,则 ℓ 对 β 的一阶导数为

$$\begin{aligned}\frac{\partial\ell}{\partial\beta}&=\sum_{i=1}^{n}[y_i-\frac{e^{\beta^T x_i}}{1+\beta^T x_i}]x_i\\&=\sum_{i=1}^{n}(y_i-p_i)x_i\end{aligned}$$

为了求出待估参数 β，对对数似然函数求二阶偏导

$$\frac{\partial^2\ell(\beta)}{\partial\beta\partial\beta^T}=-\sum_{i=1}^{n}x_i x_i^T p_i(1-p_i)$$

上面的 $\boldsymbol{x_i}$ 是向量，也就是上面所说的特征向量，维度为特征个数加1。即假设原始数据为 $\boldsymbol{n\times m}$ 矩阵，其中$\boldsymbol{n}$表示观测数，$\boldsymbol{m}$表示特征数。则$\boldsymbol{x_i}$的长度为$\boldsymbol{m+1}$。根据上述说明，上面的二阶偏导实际上是一个$(\boldsymbol{m+1})\times(\boldsymbol{m+1})$的矩阵。

将上述式子表示成矩阵的形式是

$$\frac{\partial\ell(\beta)}{\partial\beta}=X^T(y-p)$$

$$\frac{\partial^2\ell(\beta)}{\partial\beta\partial\beta^T}=-X^TWX$$

其中，$\boldsymbol{X}$为原始样本矩阵，$\boldsymbol{y}$为类别向量，**p**为预测概率向量，$\boldsymbol{W}$是一个$\boldsymbol{n\times n}$对角矩阵，第**i**个元素取值为

$$p(x_i,\hat{\beta}^{old})(1-p(x_i,\hat{\beta}^{old})).$$

那么，迭代公式可以写为

$$\begin{aligned}\hat{\beta}^{new} &= \hat{\beta}^{old} - H(\beta)^{-1}\frac{\partial \ell(\beta)}{\beta} \\ &= \hat{\beta}^{old} - H(\beta)^{-1}X^T(y-p)\end{aligned}$$

➢ 假设J(β)是我们的目标函数，则

$$\begin{aligned}J(\beta) &= -\frac{1}{n}\ell(\beta) \\ &= -\frac{1}{n}\sum_{i=1}^{n} y_i ln(p(x_i)) + (1-y_i)ln(1-p(x_i))\end{aligned}$$

此时梯度公式就变成了

$$\frac{\partial J(\beta)}{\partial \beta} = -\frac{1}{n}\sum_{i=1}^{n}(y_i - p_i)x_i = \frac{1}{n}\sum_{i=1}^{n}(p_i - y_i)x_i$$

二阶偏导数就变成了

$$\frac{\partial^2 J(\beta)}{\partial \beta \partial \beta^T} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T p_i(1-p_i)$$

那么，此时为了求得回归**参数**，即求使得J(β)最小的**参数**。牛顿迭代公式就变成了：

$$\begin{aligned}\hat{\beta}^{new} &= \hat{\beta}^{old} - H^{-1}\nabla \\ &= \hat{\beta}^{old} - \frac{1}{n}(X^T \cdot diag(p) \cdot diag(1-p) \cdot X)^{-1} \cdot \frac{1}{n}X^T(p-y)\end{aligned}$$

- **（拓展）海森矩阵的计算公式：**
  - ➢ $\Delta J(\theta)$和 H 分别表示似然函数的相对于参数 θ 的一阶导数和二阶导数

$$\Delta J(\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x^{(i)}$$

$$H = \frac{1}{N}\sum_{i=1}^{N} h_\theta\left(x^{(i)}\right)^{\mathrm{T}}\left(1 - h_\theta\left(x^{(i)}\right)\right)x^{(i)}\left(x^{(i)}\right)^{\mathrm{T}}$$

  - ➢ 海森矩阵 H 的计算公式为（一节导数再对 θ 求导）

$$
\begin{aligned}
H_{ij} =& \frac{\partial^2 l(\theta)}{\partial\theta_i \partial\theta_j} \\
&= \frac{1}{m}\frac{\partial}{\theta_j}\sum_{t=1}^{m}(y^{(t)} - h_\theta(x^{(t)}))x_i^{(t)} \\
&= \frac{1}{m}\sum_{t=1}^{m}\frac{\partial}{\theta_j}(y^{(t)} - h_\theta(x^{(t)}))x_i^{(t)} \\
&= \frac{1}{m}\sum_{t=1}^{m} -x_i^{(t)}\frac{\partial}{\partial\theta_j}h_\theta(x^{(t)}) \\
&= \frac{1}{m}\sum_{t=1}^{m} -x_i^{(t)}h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))\frac{\partial}{\theta_j}(\theta^T x^{(t)}) \\
&= \frac{1}{m}\sum_{t=1}^{m} h_\theta(x^{(t)})(h_\theta(x^{(t)}) - 1)x_i^{(t)}x_j^{(t)}
\end{aligned}
$$