回归

cloud

2016.12.13

1 线性回归

1.1 求目标函数

在样本空间中有 M 个样本和 N 维特征，它们组成矩阵 X。普通的线性回归如下所示，这里的 x_i 表示某个样本 x 的第 i 个特征：

$$\begin{aligned} h_\theta(x) &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta_n x_n \\ &= \sum_{i=0}^{n} \theta_i x_i = \theta^T x \end{aligned}$$

目标函数是使得拟合的曲线误差最小化，我们用 $\varepsilon^{(i)}$ 表示第 i 个样本的预测值和真实值的误差，误差是独立同分布的，服从均值为 0，方差为某定值 σ^2 的高斯分布。

我们采用极大似然估计来解释最小二乘，误差分布的似然函数如下所示，这里的 $x^{(i)}$ 表示第 i 个样本：

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

其中 $\varepsilon^{(i)}$ 服从的概率密度函数为

$$p(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(\varepsilon^{(i)})^2}{2\sigma^2})$$

就有

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2})$$

所以误差的似然函数为：

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) \end{aligned}$$

采用对数似然得到：

$$\begin{aligned} l(\theta) &= logL(\theta) \\ &= \sum_{i=1}^{m} log\frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}) \\ &= mlog\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2 \end{aligned}$$

因为是极大似然估计，即要使得 $l(\theta)$ 极大，就要使减号后面极小。因为第一项是定值，那么参数的极大似然估计就转化为求目标函数 $J(\theta)$ 极小，目标函数如下：

$$\begin{aligned} J(\theta) &= \frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2 \\ &= \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \end{aligned}$$

1.2 参数求解（最小二乘法）

目标函数可以写成：

$$\begin{aligned} J(\theta) &= \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \\ &= \frac{1}{2}(X\theta - y)^T(X\theta - y) \end{aligned}$$

求目标函数的梯度：

$$\begin{aligned} \nabla_\theta J(\theta) &= \nabla_\theta(\frac{1}{2}(X\theta - y)^T(X\theta - y)) \\ &= \nabla_\theta(\frac{1}{2}(\theta^T X^T - y^T)(X\theta - y)) \\ &= \nabla_\theta(\frac{1}{2}(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y)) \\ &= \frac{1}{2}(2X^T X\theta - X^T y - (y^T X)^T) \\ &= X^T X\theta - X^T y \end{aligned}$$

求其驻点就是使得梯度为 0 的点，也为极值点，即解得 θ：

$$X^T X\theta - X^T y = 0$$

$$\theta = (X^T X)^{-1} X^T y$$

如果 $X^T X$ 不可逆或防止过拟合，增加 λ 扰动，其中 $\lambda > 0$：

$$\theta = (X^T X + \lambda I)^{-1} X^T y$$

为了防止过拟合，也可以给目标函数增加平方和损失：

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2+\lambda\sum_{i=1}^{n}\theta_j^2$$

1.3 参数求解（梯度下降法）

目标函数 $J(\theta)$ 同上，梯度下降算法如下所示：

(1) 随机初始化 θ。

(2) 沿着负梯度方向迭代，更新后的 θ 使得 $J(\theta)$ 更小，对于 θ 中每一个分量 θ_j 有如下的变化，其中 $j=0...n$。

$$\theta_j:=\theta_j-\alpha\frac{\partial J(\theta)}{\partial\theta_j}$$

求梯度方向如下，这里为了简便将 $y^{(i)}$ 写成 y，$x^{(i)}$ 写成 x：

$$\begin{aligned}\frac{\partial J(\theta)}{\partial\theta_j}&=\frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x)-y)^2\\&=(h_\theta(x)-y)\frac{\partial}{\partial\theta_j}(h_\theta(x)-y)\\&=(h_\theta(x)-y)\frac{\partial}{\partial\theta_j}(\sum_{i=0}^{n}\theta_i x_i-y)\\&=(h_\theta(x)-y)x_j\end{aligned}$$

梯度是有方向的，对于一个向量 θ，每一维分量 θ_j 都可以求出一个梯度的方向，我们就可以找到一个整体的方向，在变化的时候，我们就朝着下降最多的方向进行变化就可以达到一个最小点，不管它是局部的还是全局的。所以 θ 每加入一个样本进行迭代可以理解成下面的变化，其中 α 表示学习率，可以理解为每次迭代的步长：

$$\theta=\theta-\alpha\begin{pmatrix}\frac{\partial J(\theta)}{\partial\theta_0}\\...\\\frac{\partial J(\theta)}{\partial\theta_n}\end{pmatrix}$$

对于梯度下降方式，有随机梯度下降和批量梯度下降，前者是每加入一个样本进行一次迭代，后者是加入所有样本之后进行一次迭代。

(1) 随机梯度下降：

循环迭代 {

$for\quad i=1...m${

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

}

}

(2) 批量梯度下降：

循环迭代直到收敛 {

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

}

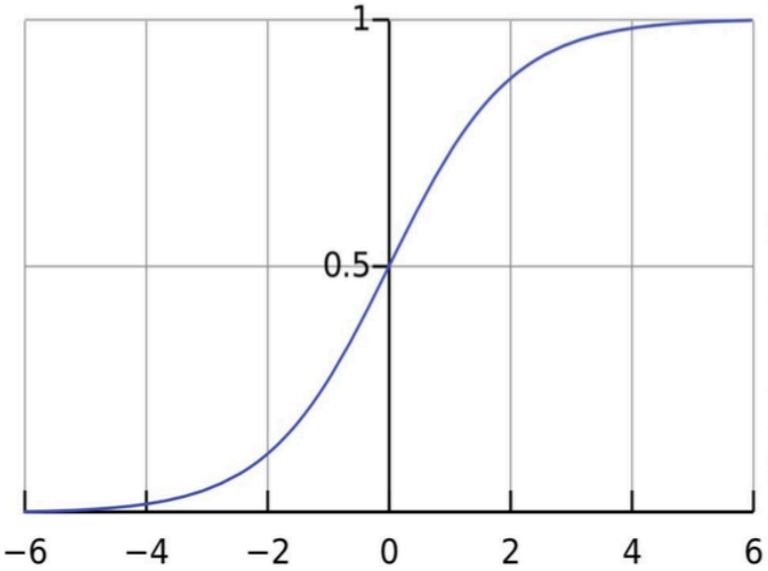
1.4 总结

线性回归的目标函数本质上就是假定误差服从正态分布，得到目标函数之后通过最小二乘法或者梯度下降法求解参数。

2 逻辑回归

2.1 *sigmod* 函数

逻辑回归用于分类模型，将 $\theta^T x$ 代入 *sigmod* 函数即可得到分类的结果 y, 当 $h_\theta(x) > 0.5$ 分类的结果为 1，当 $h_\theta(x) < 0.5$ 分类结果为 0. 图示如下:



sigmod 函数如下：

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

2.2 参数求解（梯度下降法）

我们假定对于每一个样本满足概率：

$$P(y=1|x;\theta)=h_\theta(x)$$

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

可以写成：

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

对应的似然函数为：

$$\begin{aligned}L(\theta)&=\prod_{i=1}^{m}P(y^{(i)}|x^{(i)};\theta)\\&=\prod_{i=1}^{m}(h_\theta(x^{(i)}))^{y^{(i)}}(1-h_\theta(x^{(i)}))^{1-y^{(i)}}\end{aligned}$$

取对数可得：

$$\begin{aligned}l(\theta)&=logL(\theta)\\&=\sum_{i=1}^{m}y^{(i)}log(h_\theta(x^{(i)}))+(1-y^{(i)})log(1-h_\theta(x^{(i)}))\end{aligned}$$

采用梯度下降法进行参数估计，这里为了简便将 $y^{(i)}$ 写成 y，$x^{(i)}$ 写成 x：

$$\begin{aligned}\frac{\partial}{\partial\theta_j}l(\theta)&=(\frac{y}{g(\theta^Tx)}-\frac{1-y}{1-g(\theta^Tx)})\frac{\partial}{\partial\theta_j}g(\theta^Tx)\\&=(\frac{y}{g(\theta^Tx)}-\frac{1-y}{1-g(\theta^Tx)})g(\theta^Tx)(1-g(\theta^Tx))\frac{\partial}{\partial\theta_j}\theta^Tx\\&=(y-h_\theta(x))x_j\end{aligned}$$

参数的学习规则如下，它和线性回归有着同样的形式。

(1) 随机梯度下降：

循环迭代 {

$for\quad i=1...m${

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

}

}

(2) 批量梯度下降：

循环迭代直到收敛 {

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

}

2.3 损失函数

损失函数有很多种定义，在这里我们使用逻辑回归对数损失函数，这里表示单个样本的损失函数：

$$cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & if \quad y = 1 \\ -log(1 - h_\theta(x)) & if \quad y = 0 \end{cases}$$

之所以这样定义，是因为当 $y = 1$ 时，表示这个样本为正类。如果此时 $h_\theta(x) = 1$, 则单对这个样本而言的 $cost = 0$, 表示这个样本的预测完全准确。但是如果 $h_\theta(x) = 0$，那么表示预测完全错误，那么 $cost = +\infty$。当 $y = 0$ 时，推理过程跟上述完全一致。

那么总的损失函数可以写成：

$$cost = \sum_{i=1}^{m} -y^{(i)} log(h_\theta(x^{(i)})) - (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))$$

由此我们发现损失函数 $cost = -l(\theta)$，$cost$ 要极小，那么 $l(\theta)$ 就要极大，等价于 2.2 中通过极大似然估计来得到的目标函数。极大似然估计法的原理就是通过求得相关参数使得所有样本同时出现的概率最大。

2.4 总结

逻辑回归的目标函数是通过极大似然方法或者极小化损失函数的方法得到的，参数求解常采用梯度下降法。

3 参考文献

1. 斯坦福大学吴恩达课件

2. 小象学院邹博课件

3.http://blog.csdn.net/suibianshen2012/article/details/51532348

4.http://blog.csdn.net/bitcarmanlee/article/details/51165444