深入理解线性回归算法（一）

原创 石头 机器学习算法那些事 2018-10-25

前言

线性回归算法是公众号介绍的第一个机器学习算法，原理比较简单，相信大部分人对线性回归算法的理解多于其他算法。本文介绍的线性回归算法包括最小二乘法和最大似然法，进而讨论这两种算法蕴含的一些小知识，然后分析算法的偏差和方差问题，最后总结全文。

目录

最小二乘法和最大似然函数

最小二乘法

训练数据D共有N个观测数据，数据的输入$\vec{x}=(x_1,x_2,\cdots x_N)^T$，
输出$\vec{y}=(y_1,y_2,\cdots y_N)^T$，假设模型有$M$个参数个数
最小二乘法求数据集D的线性回归模型线步骤如下：
(1)、线性回归的表达式：

$$y(x,w)=w_0+\sum_{j=1}^{M-1}w_j\phi_j(x)$$

$$y(x,w)=\sum_{j=0}^{M-1}w_j\phi_j(x)=\vec{w}^T\,\overline{\phi(x)}$$

其中，$\vec{w}=(w_0,w_1,\cdots w_{M-1})^T$，$\overline{\phi(x)}=(\phi_0(x),\phi_1(x),\cdots,\phi_{M-1}(x))^T$，$\phi_0(x)=1$

(2)、最小二乘法求最优参数$\vec{w}$

设误差平方和$E_D(\mathrm{w})$

$$E_D(\overline{\mathrm{w}}) = \sum_{n=1}^{N}(t_n - \overline{w}^T \overline{\phi(x)})^2$$

由$\frac{\partial E_D(\overline{w})}{\partial \overline{w}} = 0$,可得最优化参数$\overline{\mathrm{w}}$

最大似然函数

假设训练数据的目标变量t是由确定性方程**y(x,w)**和高斯噪声叠加产生的，即：

$$\mathrm{t} = y(x,w) + \varepsilon$$

其中 ε 是期望为0，精度为β（方差的倒数）的高斯噪声的随机抽样。
目标变量t的分布推导如下：

$\mathrm{t} = y(x,w) + \varepsilon$
等式两边取期望，得：
$E(\mathrm{t}) = E(y(x,w)) + E(\varepsilon)$
$\because$ y(x,w)为确定性方程（可理解成常数）
变量ε的期望为0，精度为β
$\therefore E(\mathrm{t}) = y(x,w)$
同理，$D(\mathrm{t}) = \beta^{-1}$

因此，目标变量t的分布：

$$\mathrm{p}(t \mid x, w, \beta) = N(\mathrm{t} \mid y(x,w), \beta^{-1})$$

即观测数据集的似然函数：

$$\mathrm{p}(t \mid \overline{x}, \overline{w}, \beta) = \prod_{n=1}^{N} N(\mathrm{t} \mid \overline{w}^T \overline{\phi(x_n)}, \beta^{-1})$$

为了书写方便，求最大似然函数对应的参数$\overline{w}$：
似然函数取对数并不影响结果：

$$\ln p(t|\vec{w},\vec{\beta}) = \sum_{n=1}^{N} \ln N(t_n|\vec{w}^T\overrightarrow{\phi(x_n)}, \beta^{-1})$$

$$= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\vec{w})$$

由上式可知：

最大化似然函数$\ln p(t|\vec{w},\vec{\beta})$等价于最小化方差平方和$E_D(\vec{w})$

$$E_D(\vec{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \vec{w}^T\overrightarrow{\phi(x_n)}\}^2$$

似然函数对变量$\vec{w}$求梯度，得

$$\nabla \ln p(t|\vec{w},\vec{\beta}) = \sum_{n=1}^{N}\{t_n - \vec{w}^T\overrightarrow{\phi(x_n)}\}\overrightarrow{\phi(x_n)}^T$$

令$\nabla \ln p(t|\vec{w},\vec{\beta}) = 0$，得

$$0 = \sum_{n=1}^{N} t_n\overrightarrow{\phi(xn)}^T - \vec{w}^T\left(\sum_{n=1}^{N}\overrightarrow{\phi(x_n)}\overrightarrow{\phi(x_n)}^T\right)$$

解得最优参数$\overrightarrow{w_{ML}}$

$$\overrightarrow{w_{ML}} = (\vec{\Phi}^T\vec{\Phi})^{-1}\vec{\Phi}^T\vec{t}$$

$$\text{其中},\vec{\Phi} = \begin{pmatrix} \phi_0(x_1),\phi_1(x_1),\cdots,\phi_{M-1}(x_1) \\ \phi_0(x_2),\phi_1(x_2),\cdots,\phi_{M-1}(x_2) \\ \vdots \\ \phi_0(x_2),\phi_1(x_2),\cdots,\phi_{M-1}(x_2) \end{pmatrix}$$

因此，对于输入变量**x**，即可求得输出变量**t**的期望。

$$E(t) = \overrightarrow{w_{ML}}^T\overrightarrow{\phi(x)}$$

期望值就是模型的预测输出变量，与最小二乘法的预测结果相同。

算法若干细节的分析

偏置参数w0

线性回归表达式的偏置参数w0有什么意义，我们最小化$E_D(w)$来求解w0，根据w0结果来说明其意义。

$$E_D(w)=\frac{1}{2}\sum_{n=1}^{N}\{t_n-\sum_{j=0}^{M-1}w_j\phi_j(x_n)\}^2$$

$$E_D(w)=\frac{1}{2}\sum_{n=1}^{N}\{t_n-w_0-\sum_{j=1}^{M-1}w_j\phi_j(x_n)\}^2$$

$$\frac{\partial E_D(w)}{\partial w_0}=\sum_{n=1}^{N}\{t_n-w_0-\sum_{j=1}^{M-1}w_j\phi_j(x_n)\}$$

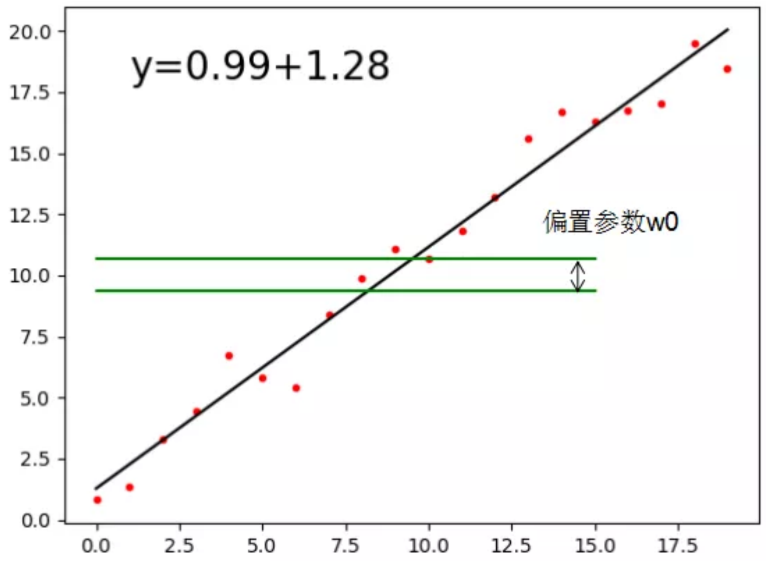
$$令\frac{\partial E_D(w)}{\partial w_0}=0$$

$$\mathrm{w}_0=\bar{t}-\sum_{j=1}^{M-1}w_j\overline{\phi_j}$$

$$其中,\bar{\mathrm{t}}=\frac{1}{N}\sum_{n=1}^{N}t_n,\overline{\phi_j}=\frac{1}{N}\sum_{n=1}^{N}\phi_j(x_n)$$

由w0结果可知，偏置参数w0补偿了目标值的平均值（在训练集）与基函数的值的加权求和之间的差。

图形表示为：



最小二乘法的几何意义

根据最小二乘法的结果可以作如下推导：

令$\overrightarrow{\varphi_j} = (\phi_j(x_1), \phi_j(x_2), \cdots, \phi_j(x_N))^T, \vec{t} = (t_1, t_2, \cdots t_N)^T$

$$\because \quad \vec{\Phi} = \begin{pmatrix} \phi_0(x_1), \phi_1(x_1), \cdots, \phi_{M-1}(x_1) \\ \phi_0(x_2), \phi_1(x_2), \cdots, \phi_{M-1}(x_2) \\ \vdots \\ \phi_0(x_2), \phi_1(x_2), \cdots, \phi_{M-1}(x_2) \end{pmatrix}$$

$$\therefore \quad \vec{\Phi} = (\overrightarrow{\varphi_0}, \overrightarrow{\varphi_1}, \cdots, \overrightarrow{\varphi_{M-1}})$$

由最小二乘法求得最优参数$\overrightarrow{\mathrm{w}_{ML}}$，令$\overrightarrow{\mathrm{w}_{ML}} = (w_0, w_1, \cdots, w_{M-1})^T$

由线性回归表达式可得：

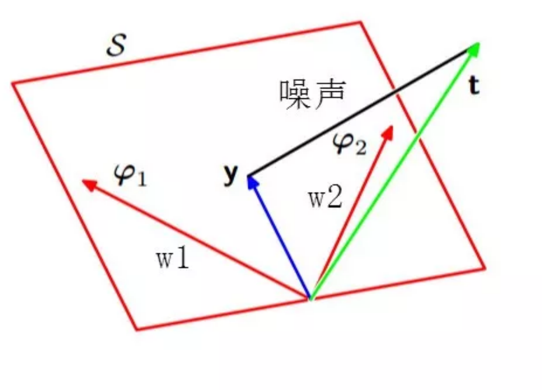
$$\vec{\Phi} * \overrightarrow{\mathrm{w}_{ML}} = \vec{t}$$

$$w_0 * \overrightarrow{\varphi_0} + w_1 * \overrightarrow{\varphi_1} + w_2 * \overrightarrow{\varphi_2} + \cdots w_{M-1} * \overrightarrow{\varphi_{M-1}} = \vec{t}$$

由上式可知：

训练数据集的目标变量$\vec{t}$可以分解成$M-1$个基函数，系数w_j为目标变量$\vec{t}$在基函数的投影。

图形表示如下：



黑色线表示噪声。

备注： 推导公式是假设 $\Phi^T\Phi$ 是非奇异矩阵（ $\Phi^T\Phi$ 的行列式不等于0），若 $\Phi^T\Phi$ 是奇异矩阵，则需要通过奇异值分解（SVD）成新的基向量，后续文章会讲到。

噪声模型分析

线性回归模型叠加的噪声是假设均值为0方差为ε的高斯分布，下面是笔者分析这一假设的原因。

假设噪声是高斯分布的原因： 高斯分布是实际生活中最常见的高斯分布，采用高斯分布的模型更贴近实际情况。

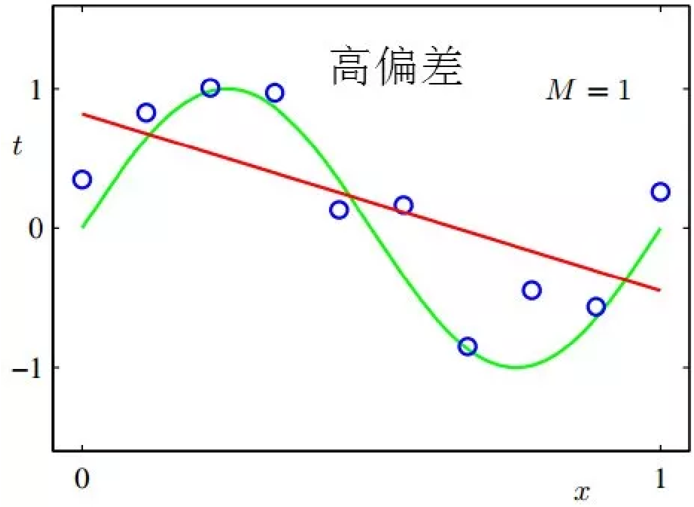
假设噪声是均值为0的原因： 这个比较好理解，就是为了方便计算，偏置参数w0包含了噪声均值。

偏差和方差

最小二乘法和最大似然法构建的模型是一样的，本文的线性回归表达式的复杂度用模型参数的个数来表示，模型参数个数越多，则模型复杂度越大；反之模型复杂度越小（**只针对无正则化的线性回归方程**）。本节讨论模型复杂度与偏差和方差的关系。

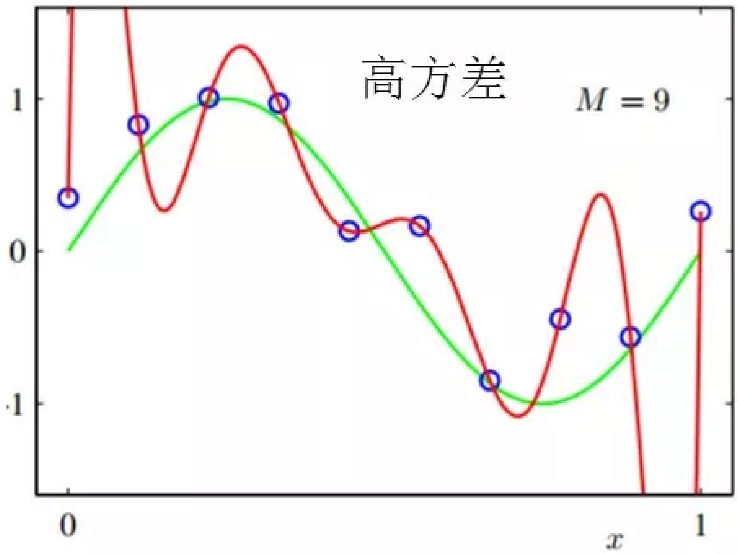
高偏差

若模型参数个数比较少，即模型复杂度很低，模型处于高偏差状态。
如下图用直线去拟合正弦曲线。



高方差

若模型参数个数较大，即复杂度较高，则模型处于高方差（过拟合）状态。
如下图M=9拟合正弦曲线，模型训练误差为0。



总结

本文介绍了最小二乘法和最大似然法来求线性回归的最优参数，分析了算法中容易忽视的某些细节，由于本文的线性回归表达式没有正则化项，因此模型的复杂度等同于模型参数的个数，参数个数过多模型容易产生高方差（过拟合），参数个数过低模型容易产生高偏差，下节将要介绍贝叶斯线性回归算法，该算法很好的解决了复杂度的问题。

参考：

Christopher M.Bishop <<Pattern Reconition and Machine Learning>>

推荐阅读文章

线性回归：不能忽视的三个问题

浅谈频率学派和贝叶斯学派

浅谈先验分布和后验分布

模型优化的风向标：偏差与方差

机器学习模型性能评估（三）：代价曲线

机器学习模型性能评估（二）：P-R曲线与ROC曲线

-END-