MLAPP 读书笔记 - 07 线性回归(Linear regression)

> A Chinese Notes of MLAPP,MLAPP 中文笔记项目
> https://zhuanlan.zhihu.com/python-kivy

记笔记的人：cycleuser

2018年06月15日13:41:04

7.1 概论

线性回归是统计学和(监督)机器学习里面的基本主力.使用核函数或者其他形式基函数来扩展之后,还可以用来对非线性关系进行建模.把高斯输出换成伯努利或者多元伯努利分部,就还可以用到分类上面,这些后文都会讲到.所以这个模型很值得详细学习一下.

7.2 模型选择

在本书1.4.5已经看到过线性回归了,其形式为:

$$p(y|x\theta) = N(y|w^T x\sigma^2) \text{(7.1)}$$

线性回归也可以通过将x替换成为输入特征的非线性函数比如$\phi(x)$来对非线性关系进行建模.也就是将形式变成了:

$$p(y|x\theta) = N(y|w^T \phi(x)\sigma^2) \text{(7.2)}$$

这就叫基函数扩展(basis function expansion).(要注意这时候模型依然是以w为参数,依然还是线性模型;这一点后面会有很大用处.)简单的例子就是多项式基函数,模型中函数形式为:

$$\phi(x) = [1, x, x^2, \ldots, x^d] \text{(7.3)}$$

图1.18展示了改变d的效果,增加d就可以建立更复杂的函数.

对于多输入的模型,也可以使用线性回归.比如将温度作为地理位置的函数来建模.图7.1(a)所示为:
$\mathrm{E}[y|x] = w_0 + w_1 x_1 + w_2 x_2$,图7.1(b)所示为:
$\mathrm{E}[y|x] = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2$.

7.3 最大似然估计(最小二乘法)

最大似然估计(MLE)是估计统计模型参数的常用方法了,定义如下:

$$\widehat{\theta} \triangleq \arg\max_{\theta} \log p(D|\theta) \text{(7.4)}$$

此处参考原书图7.1

通常假设训练样本都是独立同分布的(independent and identically distributed,缩写为iid).这就意味着可以写出下面的对数似然函数(log likelihood):

$$l(\theta) \triangleq \sum_{i=1}^{N} \log p(y_i|x_i,\theta) \text{(7.5)}$$

我们可以去最大化对数似然函数,或者也可以等价的最小化负数对数似然函数(the negative log likelihood,缩写为NLL):

$$NLL(\theta) \triangleq -\sum_{i=1}^{N} \log p(y_i|x_i,\theta) \text{(7.6)}$$

负对数似然函数(NLL)有时候更方便,因为很多软件都有专门设计找最小值的函数,所以比最大化容易.

接下来设我们对这个线性回归模型使用最大似然估计(MLE)方法.在上面的公式中加入高斯分布的定义,就得到了下面形式的对数似然函数:

$$l(\theta) = \sum_{i=1}^{N} \log[(\frac{1}{2\pi\sigma^2})^{\frac{1}{2}} \exp(-\frac{1}{2\sigma^2}(y_i - w^T x_i)^2)] \qquad (7.7)$$

$$= \frac{-1}{2\sigma^2} RSS(w) - \frac{N}{2}\log(2\pi\sigma^2) \qquad (7.8)$$

上式中的RSS是residual sum of squares的缩写,意思是残差平方定义为:

$$RSS(w) \triangleq \sum_{i=1}^{N}(y_i - w^T x_i)^2 \text{ (7.9)}$$

此处参考原书图7.2

RSS也叫做平方误差总和(sum of squared errors),这样也可以缩写成SSE,这样就有SSE/N,表示的是均方误差MSE(mean squared erro).也可以写成残差(residual errors)向量的二阶范数(l2 norm)的平方和:

$$RSS(w) = ||\epsilon||_2^2 = \sum_{i=1}^{N} \epsilon_i^2 \text{ (7.10)}$$

上式中的$\epsilon_i = (y_i - w^T x_i)^2$.

这样就能发现w的最大似然估计(MLE)就是能让残差平方和(RSS)最小的w,所以这个方法也叫作小

二乘法(least squares).这个方法如图7.2所示.图中红色圆点是训练数据x_i, y_i,蓝色的十字点是估计数据$x_i, \hat{y}_i$,竖直的蓝色线段标识的就是残差$\epsilon_i = y_i - \hat{y}_i$.目标就是要寻找能够使平方残差总和(图中蓝色线段长度)最小的图中所示红色直线的参数(斜率w_1和截距w_0).

在图7.2(b)中是线性回归样例的负对数似然函数(NLL)曲面.可见其形态类似于一个单底最小值的二次型碗,接下来就要进行以下推导.(即便使用了基函数扩展,比如多项式之类的,这也是成立的,因为虽然输入特征可以不是线性的,单负对数似然函数依然还是以w为参数的线性函数.)

7.3.1 最大似然估计(MLE)的推导

首先以更好区分的形式重写目标函数(负对数似然函数):

$$NLL(w) = \frac{1}{2}(y - Xw)^T(y - Xw) = \frac{1}{2}w^T(X^TX)w - w^T(X^Ty) \text{ (7.11)}$$

上式中

$$X^TX = \sum_{i=1}^N x_i x_u^T = \sum_{i=1}^N \begin{pmatrix} x_{i,1}^2 & \dots x_{i,1}x_{i,D} \\ & \dots \\ x_{i,D}x_{i,1} & \dots & x_{i,D}^2 \end{pmatrix} \text{ (7.12)}$$

是矩阵平方和（sum of squares matrix）,另外的一项为：

$$X^Ty = \sum_{i=1}^N x_i y_i \text{ (7.13)}$$

使用等式4.10中的结论,就得到了梯度函数（gradient）,如下所示：

$$g(w) = [X^TXw - X^Ty] = \sum_{i=1}^N x_i(w^Tx_i - y_i) \text{ (7.14)}$$

使梯度为零,则得到了：

$$X^TXw = X^Ty \text{ (7.15)}$$

这就是正规方程(normal equation).这个线性方程组对应的解$\hat{w}$就叫做常规最小二乘解（ordinary least squares solution,缩写为 OLS solution）：

$$\hat{w}_{OLS} = (X^TX)^{-1}X^Ty \text{ (7.16)重要公式}$$

7.3.2 几何解释

这个方程有很优雅的几何解释.假设N>D,也就意味样本比特征数目多.X列向量（columns）定义的是在N维度内的一个D维度的子空间.设第j列为$\tilde{x}_j$,是在R^N上的一个向量.(应该不难理解,$x_i \in R^D$表示的就是数据情况中的第i个.)类似的y也是一个R^N中的向量.例如,如果社N=3个样本,二D=2的子空间:

$$X = \begin{pmatrix} 1 & 2 \\ 1 & -2 \\ 1 & 2 \end{pmatrix}, y = \begin{pmatrix} 8.8957 \\ 0.6130 \\ 1.7761 \end{pmatrix} \text{ (7.17)}$$

这两个向量如图7.3所示.

然后我们就要在这个线性子空间中找一个尽可能靠近y的向量$\hat{y} \in R^N$,也就是要找到:

$\arg\min_{\hat{\in} span(\{\tilde{x}_1,\ldots,\tilde{x}_D\})} ||y - \hat{y}||_2$ (7.18)

由于$\hat{y} \in span(X)$,所以就会存在某个权重向量(weight vector)w使得:

$\hat{y} = w_1\tilde{x}_1 + \ldots + w_D\tilde{x}_D = Xw$ (7.19)

此处参考原书图7.3

要最小化残差的范数(norm of the residual)$y - \hat{y}$,就需要让残差向量(residual vector)和X的每一列相正交(orthogonal),也就是对于$j = 1 : D$有$\tilde{x}_j^T(y - \hat{y}) = 0$.因此有:

$$\tilde{x}_j^T(y - \hat{y}) = 0 \implies X^T(y - Xw) = 0 \implies w = (X^TX)^{-1}X^Ty \text{ (7.20)}$$

这样y的投影值就是:

$$\hat{y} = X\hat{w} = X(X^TX)^{-1}X^Ty \text{(7.21)}$$

这对应着在X的列空间(column space)中的y的正交投影(orthogonal projection).投影矩阵$P \triangleq X(X^TX)^{-1}X^T$就叫做帽子矩阵(hat matrix),因为在y上面盖了个帽子成了$\hat{y}$.

7.3.3 凸性质

在讲到最小二乘法的时候,我们注意到负对数似然函数(NLL)形状像是一个碗,有单一的最小值.这样的函数用专业术语来说是凸(convex)的函数.凸函数在机器学习里面非常重要.

然后咱们对这个概念进行一下更确切定义.设一个集合S,如果对于任意的$\theta, \theta' \in S$,如果有下面的性质,则S是凸的集合:

$$\lambda\theta + (1 - \lambda)\theta' \in S, \forall\lambda \in [0, 1] \text{(7.22)}$$

此处参考原书图7.4
此处参考原书图7.5

也就是说在θ和θ'之间连一条线,线上所有的点都处在这个集合之内.如图7.4(a)所示就是一个凸集合,而图7.4(b)当中的就是一个非凸集合.

一个函数的上图(epigraph,也就是一个函数上方的全部点的集合)定义了一个凸集合,则称这个函数$f(\theta)$就是凸函数.反过来说,如果定义在一个凸集合上的函数$f(\theta)$满足对任意的$\theta, \theta' \in S$,以及任意的$0 \leq \lambda \leq 1$,都有下面的性质,也说明这个函数是凸函数:

$$f(\lambda\theta + (1-\lambda)\theta') \leq \lambda f(\theta) + (1-\lambda)f(\theta') \text{ (7.23)}$$

图7.5(b)是一个一维样本.如果不等式严格成立,就说这个函数是严格凸函数(strictly convex).如果其反函数$-f(\theta)$是凸函数,则这个函数$f(\theta$是凹函数(concave).标量凸函数(scalar convex function)包括$\theta^2, e^\theta, \theta\log\theta(\theta > 0)$.标量凹函数(scalar concave function)包括$\log(\theta), \sqrt{\theta}$.

直观来看,(严格)凸函数就像是个碗的形状,所以对应在碗底位置有全局的唯一最小值θ^*.因此其二阶导数必须是全局为正,即$\frac{d}{d\theta}f(\theta) > 0$.当且仅当一个二阶连续可微(twice-continuously differentiable)多元函数f的海森矩阵(Hessian)对于所有的θ都是正定的(positive definite),这个函数才是凸函数.在机器学习语境中,这个函数f通常对应的都是负对数似然函数(NLL).

此处参考原书图7.6

负对数似然函数(NLL)是凸函数的模型是比较理想的.因为这就意味着能够找到全局最优的最大似然估计(MLE).本书后面还会看到很多这类例子.不过很多模型还并不一定就能有凹的似然函数.这时候就要推一些方法来求局部最优参数估计了.

7.4 健壮线性回归*

通常我们用均值为零且方差固定的正态分布$\epsilon_i \sim N(0, \sigma^2)$来对回归模型的噪音建模,$\epsilon_i = y_i - w^T x_i$.这时候对似然函数最大化就等价于使平方残差总和(sum of squared residuals)最小,这部分之前刚才已经讲过了.不过如果在数据里面有异常值/离群点(outliers),就容易影响你和质量,如图7.6(a)所示,图底部的几个点就是异常值.这是因为平方差(squared error)以二次形式惩罚偏差,所以远离拟合曲线的点会比临近曲线的点对拟合有更大的影响.

有一种方法可以实现对异常值的健壮性,就是把对响应变量的正态分布替换成更重尾(heavy tails)的分布.这样的分布赋予异常值更高似然性,而不用改变直线去特地解释这些异常值.

拉普拉斯分布(Laplace distribution)就是一个选择,如本书2.4.3所示.如果用拉普拉斯分布来对回归的观测模型建模,就得到了下面的似然函数:

$$p(y|x, w, b) = Lap(y|w^T x, b) \propto \exp(-\frac{1}{b}|y - w^T x|) \text{(7.24)}$$

为了健壮性,将平方项替换成绝对值,即用$|y - w^T x|$替换$(y - w^T x)^2$.为了简单起见,这里就设b是一个固定值.然后设第i个残差为$r_i \triangleq y_i - w^T x_i$.这样负对数似然函数(NLL)就成了:

$$l(w) = \sum_i |r_i(w)| \text{(7.25)}$$

似然函数	先验	名称	章节
正态分布	均匀先验	最小二乘法	7.3
正态分布	正态分布	岭回归	7.5
正态分布	拉普拉斯	套索回归(Lasso)	13.3
拉普拉斯	均匀先验	健壮回归	7.4
学生分布	均匀先验	健壮回归	练习11.12

表 7.1 对线性回归中各种似然函数和先验的总结.似然函数指的是$p(y|x,w,\sigma^2)$的分布,先验指的是$p(w)$的分布.均匀分布的最大后验估计(MAP)对应的就是最大似然估计(MLE).

然而很不幸,等式7.25是一个非线性的目标函数,很难去优化.还好我没可以将负对数似然函数(NLL)转化成一个线性目标函数,受线性约束的限制,这需要用到分割变量(split variable)的技巧.首先定义:

$$r_i \triangleq r_i^+ - r_i^- \text{(7.26)}$$

然后将线性不等式约束带入,即$r_i^+ \geq 0, r_o^- \leq 0$.这样受约束的目标函数就成了:

$$\min_{w,r^+,r^-} \sum_i (r^+ - r_i^-) s.t. r_i^+ \geq 0, r_i^- \geq 0, w^T x_i + r_i^+ + r_i^- = y_i \text{(7.27)}$$

这就是一个线性规划(linear program),有D+2N未知和3N约束.(This is an example of a linear program with D + 2N unknowns and 3N constraints.)

因为这是一个凸优化问题,所以有唯一解.要解线性规划(LP),必须首先将其写成标准型:

$$\min_\theta f^T \theta s.t. A\theta \leq b, A_{ep}\theta = b_{eq}, 1 \leq \theta \leq u \text{(7.28)}$$

目前在咱们的例子中,
$\theta = (w, r^+, r^-), f = [0,1,1], A = [], b = [], A_{eq} = [X, I, -I], b_{eq} = y, I = [-\infty 1, 0, 0], u$
.这可以通过线性规划求解器(LP solver)(参考Boyd and Vandenberghe 2004).图7.6(b)所示为具体方法的一个例子.

除了拉普拉斯似然函数下使用负对数似然函数,另一种方法是最小化胡贝尔损失函数(Huber loss function,出自 Huber 1964),定义如下:

$$L_H(r,\delta) = \begin{cases} r^2/2 & \text{if } |r| \leq \delta \\ \delta|r| - \delta^2/2 & \text{if } |r| > \delta \end{cases} \text{(7.29)}$$

这个定义等价于对小于δ的误差使用二次损失函数l_2,对大于δ的误差使用一次损失函数l_1.如图7.6(b)所示.这个损失函数的优势就是全局可微(everywhere differentiable),这是利用了当$r \neq 0$时候$\frac{d}{dr}|r| = sign(r)$.还可以验证这个函数是$C_1$连续的,因为两部分函数的梯度都符合$r = \pm\delta$,名

义上就是$\frac{d}{dr}L_H(r,\delta)|_{r=\delta}=\delta$.结果就是胡贝尔损失函数油画起来比拉普拉斯似然函数容易多了,因为可以使用标准光滑优化方法(比如拟牛顿法,quasi-Newton),而不用使用线性规划了.

图7.6(a)所示的就是胡贝尔损失函数.结果和概率方法的结果很相似.实际上胡贝尔方法确实也有一种概率角度的解释,不过就是不太自然 (Pontil et al).

此处参考原书图7.7

7.5 岭回归

最大似然估计中一大问题就是过拟合.在本节要讲的就是使用高斯先验的最大后验估计来改善这个问题.为了简单起见,本节用高斯似然函数,而不用健壮似然函数了.

7.5.1 基本思想

最大似然估计过拟合的原因就是它选的参数都是最适合对训练数据建模的;但如果训练数据有噪音,这些参数就经常会形成非常复杂的函数.举个简单的例子,对一个N=21个点的数据使用最小二乘法拟合一个14次多项式曲线.得到的函数就可能是非常七扭八歪,如图7.7(a)所示.对应的除了w_0之外的最小二乘系数为:

6.560, -36.934, -109.255, 543.452, 1022.561, -3046.224, -3768.013,
8524.540, 6607.897, -12640.058, -5530.188, 9479.730, 1774.639, -2821.526

可见其中有很多特别大的正数和负数.这些系数让曲线七扭八歪,完美地插入所有数据.但这就很不稳定:如果我们稍微改变一下数据,系数就会发生巨大变化.

可以让系数小一点,这样得到的就是更光滑的曲线,使用一个零均值高斯先验就可以了:

$$p(w)=\prod_j N(w_j|0,\tau^2 \quad (7.30)$$

其中的$1/\tau^2$控制了先验强度.对应的最大后验估计(MAP)问题就是:

$$\arg\max_w \sum_{i=1}^N \logN(y_i|w_0+w^Tx_i,\sigma^2)+\sum_{j=1}^D \log(w_j|0,\tau^2) \quad (7.31)$$

此处参考原书图7.8

很容易证明上面这个问题等价于对下面这个项目求最小值:

$$J(w)=\frac{1}{N}\sum^N_{i=1}(y_i-(w_0+w^Tx_i))^2+\lambda||w||^2_2 \quad (7.32)$$

其中的$\lambda \triangleq \sigma^2/\tau^2$,而$||w||_2^2=\sum_j w^Tw$是平方二范数(squared two-norm).这样第一项就依然还是均方误差比负对数似然函数(MSE/NLL),第二项$\lambda \geq 0$就是符合惩罚项.对应的解为:

$\hat{w}_{ridge} = (\lambda I_D + X^T X)^- X^T y$(7.33)重要公式

这个方法就叫做岭回归(ridge regression),也叫惩罚最小二乘法(penalized least squares).通常,将用高斯先验来使参数变小的方法叫做l_2规范化(l_2 regularization),或者叫做权重衰减(weight decay).要注意,偏移项w_0并不是规范化的,因为这只影响函数的高度,而不影响其复杂性.通过对权重烈度综合(sum of the magnitudes of the weights)进行惩罚,能确保函数尽量简单(比如w=0对应的就是一条直线,也就是最简单的函数了,对应是常数.)

图7.7所示为此方法的思想,图中表明增加λ就会导致函数曲线更加光滑.得到的系数也更小.例如使用$\lambda = 10^{-3}$,就得到了下面的系数:

2.128, 0.807, 16.457, 3.704, -24.948, -10.472, -2.625, 4.360, 13.711,
10.063, 8.716, 3.966, -9.349, -9.232

在图7.8(a)中,对训练集上的均方误差(MSE)和$\log(\lambda)$的函数关系进行头图.可见随着增长λ,也就是让模型受约束程度增加,训练集上的误差也增加了.对于测试集,就呈现了U形曲线的,也就是模型先是过拟合,然后又欠拟合.通常都用交叉验证来选择λ,如图7.8(b)所示.在本书14.8,会用更加概率论的方法来对此进行讲解.

本书中会考虑到使用不同先验的效果.每一种都对应着不同形式的规范化(regularization).这个方法广泛用于防止过拟合.

7.5.2 数值稳定计算*

有意思的是,岭回归不仅在统计学上效果更好,也更容易进行数值拟合,因为$(\lambda I_D + X^T X)$比$X^T X$有更好条件(更容易可逆),至少对于适当的大的λ.

尽管如此,出于数值计算稳定性考虑,矩阵求逆还是尽量要避免的.(比如如果你在MATLAB里面写了$w = inv(X; *X) * X' y$,都会遇到警告.)接下来咱们讲一个拟合岭回归模型的有用技巧(另外通过扩展也可以用于计算Vanilla普通最小二乘估计(Ordinary least squares,OLS)),使数值计算健壮性提高.假设先验形式为$p(w) = N(0, \Lambda^2)$,其中的Λ是精度矩阵(precision matrix).在岭回归的情况下,$\Lambda = (1/\tau^2)I$.为了避免惩罚w_0项,应该先将数据中心化,如练习7.5所讲.

首先从先验中哪来一些虚拟数据来对原始数据进行扩充:

$$\tilde{X} = \begin{pmatrix} X/\sigma \\ \sqrt{\Lambda} \end{pmatrix}, \tilde{y} = \begin{pmatrix} y/\sigma \\ 0_{D\times 1} \end{pmatrix} \text{(7.34)}$$

$$f(w) = (\tilde{y} - \tilde{X}w)^T(\tilde{y} - \tilde{X}m) \tag{7.35}$$
$$= (\begin{pmatrix} y/\sigma \\ 0 \end{pmatrix} - \begin{pmatrix} X/\sigma \\ \sqrt{\Lambda} \end{pmatrix} w)^T (\begin{pmatrix} y/\sigma \\ 0 \end{pmatrix} - \begin{pmatrix} X/\sigma \\ \sqrt{\Lambda} \end{pmatrix} w) \tag{7.36}$$
$$= (\begin{pmatrix} \frac{1}{\sigma}(y - XW) \\ \sqrt{\Lambda} w \end{pmatrix})^T (\begin{pmatrix} \frac{1}{\sigma}(y - XW) \\ \sqrt{\Lambda} w \end{pmatrix}) \tag{7.37}$$
$$= \frac{1}{\sigma^2}(y - Xw)^T(y - Xw) + (\sqrt{\Lambda})^T(\sqrt{\Lambda}) \tag{7.38}$$
$$= \frac{1}{\sigma^2}(y - Xw)^T(y - Xw) + w^T \Lambda w \tag{7.39}$$

因此最大后验估计就是:

$$\hat{w}_{ridge} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{y} \tag{7.40}$$

然后设:

$$\tilde{X} = QR \tag{7.41}$$

是X的QR分解(QR decomposition),其中的Q是正交的(即$Q^T Q = QQ^T = I$),而R是上三角矩阵(upper triangular).因此有:

$$(\tilde{X}^T \tilde{X})^{-1} = (R^T Q^T Q R)^{-1} = (R^T R)^{-1} = R^{-1} R^{-T} \tag{7.42}$$

因此有:

$$\hat{w}_{ridge} = R^{-1} R^{-T} R^T Q^T \tilde{y} = R^{-1} Q \tilde{y} \tag{7.43}$$

由于R是上三角矩阵,所以求逆很容易.这就可以避免去对$\Lambda + X^T X$求逆就可以计算岭估计了.

这样,只要简单地计算未扩展矩阵X的QR分解,再利用原始的y,就可以计算最大似然估计(MLE)了.对于解最小二乘问题来说,这是首选方法.(实际上这个特别常用,在MATLAB里面只要一行代码就可以了,使用的是反斜杠运算符(backslash operator): w=X\y.)计算一个$N \times D$规模矩阵的QR分解只需要$O(ND^2)$的时间复杂度,所以在数值计算上很稳定.

如果D远大于N,即$D \gg N$,就要先进行SVD分解.具体来说就是设$X = USV^T$为X的SVD分解,其中的$V^T V = I_N, UU^T = U^T U = I_N$,S是一个$N \times N$的对角矩阵.然后设$Z = UD$是一个$N \times N$矩阵.然后可以将岭估计写成下面的形式:

$$\hat{w}_{ridge} = V(Z^T Z + \lambda I_N)^{-1} Z^T y \tag{7.44}$$

也就是说可以把D维度向量x_i替换成N维的向量z_i,然后跟之前一样进行惩罚拟合.接下来通过乘以一个V再把得到的N维的解转换成D维的解.几何角度来说,就旋转到一个新的坐标系统中,其中除了前面的N个参数之外其他参数都是0.这不会影响解的有效性,因为球面高斯先验(spherical Gaussian prior)具有旋转不变性(rotationally invariant).这个方法总体需要$O(DN^2)$的运算时间.

7.5.3 和主成分分析(PCA)的联系*

在本节要说岭回归和主成分分析(PCA,本书12.2)之间的联系,这一联系也会让我们明白为啥岭回归性能如此好.这部分的讨论基于(Hastie et al. 2009, p66).

设$X = USV^T$是X的SVD分解.通过等式7.44,可以得到:
$\hat{w}_{ridge} = V(S^2 + \lambda I)^{-1} SU^T y$(7.45)

这样就得到岭回归对训练集的预测:

$$\hat{y} = X\hat{w}_{ridge} = USV^T V(S^2 + \lambda I)^{-1} SU^T y \qquad (7.46)$$

$$= U\tilde{S}U^T y = \sum_{j=1}^{D} u_j \tilde{S}_{jj} u_j^T y \qquad (7.47)$$

此处参考原书图7.9

其中的
$\tilde{S}_{jj} \triangleq [S(S^2 + \lambda I)^{-1} S]_{jj} = \frac{\sigma_j^2}{\sigma_j^2 + \lambda}$(7.48)

σ_j是X的奇异值(singular values).因此有:

$\hat{y} = X\hat{w}_{ridge} = \sum_{j=1}^{D} u_j \frac{\sigma_j^2}{\sigma_j^2 + \lambda} u_j^T y$(7.49)

与之对比的最小二乘法预测为:

$\hat{y} = X\hat{w}_{ls} = (USV^T))VS^{-1}U^T y) = UU^T y = \sum_{j=1}^{D} u_j u_j^T y$(7.50)

如果和λ相比σ_j^2很小,那么方向(direction)u_j就不会对预测有太大影响.从这个角度来看,可以定义一个模型自由度(degrees of freedom)的有效数字,如下所示:

$dof(\lambda) = \sum_{j=1}^{D} \frac{\sigma_j^2}{\sigma_j^2 + \lambda}$(7.51)
当$\lambda = 0, dof(\lambda) = D$,而随着$\lambda \to \infty, dof(\lambda) \to 0$.

接下来说说为啥这个性质很理想.在7.6中,我们会看到如果对w使用一个均匀先验,就有$cov[w|D] = \sigma^2 (X^T X)^{-1}$.因此那些我们不确定w的的方向(direction)是由由最小特征值的矩阵的特征向量决定的,如图4.1所示.更进一步,在本书12.2.3中,我们会发现平方奇异值(squared singular values)σ_j^2等于$X^T X$的特征值.因此小的奇异值σ_j对应的就是高后验方差(high posterior variance)的方向.这些方向也是岭回归收缩最大的方向.

这个过程如图7.9所示.横向的w_1参数没能由数据确定(有高后验方差),而竖直方向上的w_2参数相当

确定.因此w_2^{map}很接近$\hat{w}_2^{mle}$,但w_1^{map}严重朝向先验均值(这个例子中是0)偏移.(可以和图4.14(c)比对来看,图4.14(c)所示的是不同可靠性传感器的传感器融合.)

还有一个与之相关但不太一样的方法,叫做主成分回归(principal components regression).这个方法的思路是:首先使用主成分分析(PCA)来降低数据维度到K维度,然后利用低维度特征作为输入特征进行回归.不过,这个方法的预测精确性上并不如岭回归这样好(Hastie et al. 2001, p70).原因是在主成分回归中,只有前K个(推导出来的)维度还保留着,而剩下的D-K个维度的信息都全被忽略了.而相比之下,岭回归是对所有维度进行了软加权(soft weighting).

7.5.4 大规模数据的规范化效应

规范化(regularization)是避免过拟合的最常用方法.不过还有另外一种有效的方法,就是使用大规模数据,当然了,这个不一定总能实现.直观来看就是训练用的数据规模更多,进行学习的效果就能越好.所以我们期望随着数据规模N增大,测试误差就逐渐降低到某个定值.

这个如图7.10所示,图中为不同次数多项式回归和样本规模N下的均方误差(mean squared error)(误差和训练集样本规模的曲线也叫作学习曲线(learning curve)).测试集上误差的形态有两方面决定:生成过程中的内在变异性导致的对于所有模型都会出现的无法降低的部分(也叫作噪音本底);另一个依赖于生成过程(真实情况)和模型之间差异导致的部分(也叫作结构误差,structural error).

图7.10中所示,真实情况应该是一个二次多项式,而我们分别用1次/2次/25次多项式对这个数据进行拟合.得到的三种模型对应就称之为M_1, M_2, M_{25}.从图中可以发现,M_2, M_{25}的结构误差都是0,因为都能够捕获真实生成过程.不过M_1的结构误差就特别大,这就证明了其远高于误差本底.

对于任何足以捕获真实情况的模型(也就是说有最小结构误差),测试误差都会随着样本规模增大即$N \to \infty$而趋向噪音本底.不过对于简单模型来说通常会更快趋向于0,因为要估计的参数更少.具体来说就是对于有限规模的训练集来说,我们估计得参数和给定模型类别能进行估计的最佳参数之间总是会有一些差异.这就叫做近似误差(approxmation error),会随着训练集样本规模增大,即$N \to \infty$而趋向于0,但对于简单模型来说趋向于0的速度更快.这个如图7.10所示,另外也可以参考练习7.1.

在大数据领域,简单模型效果出乎意料地好(Halevy et al. 2009).不过还是有必要学习一些更复杂的学习方法的,因为总会有一些问题中咱们没办法获得特别多的数据.甚至即便在一些数据丰富的情境下,比如网络搜索中,只要我们想要根据用户进行个性化结果生成,对任意用户的可用数据规模也都会变小(相比问题复杂程度而言).

此处参考原书图7.10

在这样的情况下,就可能需要同时学习多种相关模型,也就是所谓的多任务学习(multi-task learning).这个过程可以从有大量数据的任务中"借用统计强度"给数据规模小的任务.相关方法在本书后文中还会讲到.

7.6 贝叶斯线性回归

虽然岭回归是计算点估计的有效方法,有时候还可能要对w和σ^2的全后验进行计算.为了简单起见,就假设噪音方差σ^2已知,就只要关注与计算$p(w|D,\sigma^2)$.然后在本书7.6.3考虑更通用的情况,计算$p(w,\sigma^2|D)$.假设使用整个高斯释然模型(throughout a Gaussian likelihood model).使用一个健壮似然函数进行贝叶斯推断也是可行的,不过需要更复杂技巧(参考练习24.5).

7.6.1 计算后验

在线性回归中,似然函数为:

$$p(y|X,w,\mu,\sigma^2) = N(y|\mu + Xw, \sigma^2 I_N) \tag{7.52}$$

$$\propto \exp(-\frac{1}{2\sigma^2}(y-\mu 1_N - Xw)^T(y-\mu 1_N - Xw)) \tag{7.53}$$

其中的μ是偏移项.如果输入值是中心化的,则对于每个j都有$\sum_i x_{ij} = 0$,输出均值正负概率相等.所以假设一个不适当先验(improper prior)给μ,形式为$p(\mu) \propto 1$,然后整合起来就得到了:

$$p(y|X,w,\sigma^2) \propto \exp(-\frac{1}{2\sigma^2}||y-\bar{y}1_N - Xw||_2^2) \tag{7.54}$$

其中的$\bar{y} = \frac{1}{N}\sum_{i=1}^N y_i$ 是输出的经验均值.为了表达简洁,假设输出已经中心化了,然后将$y-\bar{y}1_N$写作为y.

上面这个高斯似然函数的共轭先验也还是高斯分布,可以表示做$p(w)\N(w|w_0,V_0)$.利用高斯分布的贝叶斯规则(灯饰4.125),就得到了下面的后验:

$$p(w|X,y,\sigma^2) \propto N(w|w)0,V_0)N(y|Xw,\sigma^2 I_N) = N(w|w_N,V_N) \tag{7.55}$$

$$W_N = V_N V_0^{-1} w_0 + \frac{1}{\sigma^2} V_N X^T y \tag{7.56}$$

$$V_N^{-1} = V_0^{-1} + \frac{1}{\sigma^2} X^T X \tag{7.57}$$

$$V_N = \sigma^2(\sigma^2 V_0^{-1} + X^T X)^{-1} \tag{7.58}$$

如果$w_0 = 0, V_0 = \tai^2 I$,且定义$\lambda = \frac{\sigma^2}{\tau^2}$那么后验均值就降低到了岭估计.这是因为高斯分布的均值和众数相等.

要对后验分布获得更深入了解(而不是只知道众数),可以考虑一个1维例子:

$$y(x,w) = w_0 + w_1 x + \epsilon \tag{7.59}$$

其中的真实参数为$w_0 = -0.3, w_1 = 0.5$.如图7.11所示是先验/似然函数/后验以及一些后验预测样本.具体来说最右边一列是函数$y(x,w^{(s)})$,其中的x取值范围在区间[-1,1],而

$w^{(s)} \sim N(w|w_N, V_N)$是从参数后验重取样的一个样本.开始的时候从先验中取样(第一行),预测就是遍布整个空间,因为先验是均匀的.随着看到了数据点之后(第二行),后验就开始收到对应的似然函数的约束了,预测也更接近观测数据了.不过我们会发现后验还是有岭状形态,反映了多解性的存在,有不同的斜率/截距.这很好理解,因为我们不能从一次观测中推出两个参数来.在看到两个点之后(第三行),后验就更窄了,预测也有更相似的斜率截距了.在观测了20个数据点之后(最后一行)后验就成了一个以真实值为中心的δ函数的形状了,真实值用白色十字表示.(这个估计会收敛到真实值是因为数据是从这个模型生成的,还因为贝叶斯估计其是连续估计器,更多细节参考本书6.4.1的讲解.)

此处参考原书图7.11

7.6.2 计算后验预测

作预测总是很难,尤其是预测未来.---Yogi Berra

在机器学习我们都更关注预测,而不是对参数的解析.利用等式4.126,可以很明显发现对于测试点x的后验预测分布也是一个高斯分布:

$$
\begin{aligned}
p(y|x, D, \sigma^2) &= \int N(y|x^T w, \sigma^2) N(w|w_N, V_N) dw && (7.60)\\
&= N(y|w_N^T x, \sigma_N^2(x)) && (7.61)\\
\sigma_N^2(x) &= \sigma^2 + x^T V_N x && (7.62)
\end{aligned}
$$

上面这个预测分布中的方差$\sigma_N^2(x)$取决于两个项:观测噪音的方差σ^2,参数方差V_N.后面这一项表示为观测方差,取决于测试点x和训练数据集D之间的距离关系.这如图7.12所示,其中误差范围随着远离训练样本中的点而增大,表示着不确定性的增加.这对于主动学习等领域来说很重要,在这些领域中我们要建模的对象是我们了解程度远不如已知数据的点.对比之下,这个插值估计就有固定的误差范围,因为:

$$p(y|x, D, \sigma^2) \approx \int N(y|x^T w, \sigma^2) \delta_{\hat{w}}(w) dw = p(y|x, \hat{w}, \sigma^2) \quad (7.63)$$

如图7.12(a)所示.

7.6.3 σ^2未知的情况下用贝叶斯推断*

在这一部分利用本书4.6.3的结论,解决在线性回归模型中计算$p(w, \sigma^2, D)$的问题.这就能推出本书7.6.1当中的结论,当时是假设σ^2是已知的.如果使用无信息先验,就会发现这和频率论统计学有一些有趣的联系.

7.6.3.1 共轭先验

一如既往,先写出似然函数:
$$p(y|X,w,\sigma^2)=N(y|Xw,\sigma^2 I_N) \tag{7.64}$$

类似本书4.6.3,很明显自然共轭先验形式如下所示:

$$\begin{aligned}
p(w,\sigma^2) &= NIG(w,\sigma^2|w_0,V_0,a_0,b_0 && (7.65)\\
&\triangleq N(w|w_0,\sigma^2 V_0)IG(\sigma^2|a_0,b_0) && (7.66)\\
&= \frac{b_0^{a_0}}{(2\pi)^{D/2}|V_0|^{\frac{1}{2}}\Gamma(a_0)}(\sigma^2)^{-(a_0+(D/2)+1)} && (7.67)\\
&\times \exp[-\frac{(w-w_0)^T V_0^{-1}(w-w_0)+2b_0}{2\sigma^2}] && (7.68)
\end{aligned}$$

此处参考原书图7.12

有了先验和似然函数,就可以得到后验如下所示:

$$\begin{aligned}
p(w,\sigma^2|D) &= NIG(w,\sigma^2|w_N,V_N,a_N,b_N) && (7.69)\\
w_N &= V_N(V_0^{-1}w_0+X^T y) && (7.70)\\
V_N &= (V_0^{-1}+X^T X)^{-1} && (7.71)\\
a_N &= a_0+n/2 && (7.72)\\
b_N &= b_0+\frac{1}{2}(w_0^T V_0^{-1}w_0+y^T y-w_N^T V_N^{-1}w_N) && (7.73)
\end{aligned}$$

w_N和V_N就和σ^2已知的情况类似了.a_N的表达时也很直观很好理解,就是用来对计数进行更新的.b_N的表达式可以按照下面方式理解:先验平方和b_0加上经验平方和$y^T y$,另外加一个w的先验误差项.

$$\begin{aligned}
p(\sigma^2|D) &= IG(a_N,b_N) && (7.74)\\
p(w|D) &= T(w_N,\frac{b_N}{a_N}V_N,2a_N) && (7.75)
\end{aligned}$$

接下来给一个利用7.6.3.3当中等式的应用样例.

类似本书4.6.3.6,这个后验预测分布也是一个学生T分布.
具体来说给定了m次新测试特征$\tilde{X}$,就有:

$$p(\tilde{y}|\tilde{X},D)=T(\tilde{y}|\tilde{X}w_N,\frac{b_N}{a_N}(I_m+\tilde{X}V_N\tilde{X}^T),2a_N) \tag{7.76}$$

预测方差有两个部分:首先是由于测量噪音导致的$(b_N/a_N)I_m$,另一个是由于对w不确定性的$(b_N/a_N)\tilde{X}V_N\tilde{X}^T$.后面这一项取决于测试输入和训练集的距离.

通常都设置$a_0=b_0=0$,对应的就是对σ^2的无信息先验,然后设置$w_0=0,V_0=g(X^T X)^{-1}$,

对应的任意正值g.这也叫做Zellner's g-prior(Zellner 1986).这里的g扮演的角色类似岭回归里面的$1/\lambda$.不过,区别是先验协方差正比于$(X^T X)^{-1}$而不是岭回归里面的I.这就保证了后验和输入范围无关(Minka 2000b).这也可以参考7.10.

接下来看个例子,如果我们使用一个无信息先验,给定N次测量的后验预测分布就是$V_N^{-1} = X^T X$.单位信息先验(unit information prior)定义为单样本包含尽量多信息的先验(Kass and Wasserman 1995).要对线性回归建立单位信息先验,需要使用$V_0^{-1} = \frac{1}{N} X^T X$就等价于g=N的g先验.

7.6.3.2 无信息先验

$$p(w, \sigma^2|D) = NIG(w, \sigma^2|w_N, V_N, a_N, b_N) \quad (7.77)$$
$$w_N = \hat{w}_{mle} = (X^T X)^{-1} X^T y \quad (7.78)$$
$$V_N = (X^T X)^{-1} \quad (7.79)$$
$$a_N = \frac{N - D}{2} \quad (7.80)$$
$$b_N = \frac{2^2}{2} \quad (7.81)$$
$$s^2 = (y - X\hat{w}_{mle})^T (y - X\hat{w}_{mle}) \quad (7.82)$$

|---|---|---|---|---|
|w_j|$\mathrm{E}[w_j|D]$|$\sqrt{var[w_j|D]}$|95\%CI|sig|

w_j	期望	标准差	95\%置信区间	显著性
w0	10.998	3.06027	[4.652,17.345]	*
w1	-0.004	0.00156	[-0.008,-0.001]	*
w2	-0.054	0.02190	[-0.099,.0.008]	*
w3	0.068	0.09947	[-0.138,0.274]	
w4	-1.294	0.56381	[-2.463,-0.124]	*
w5	0.232	0.10438	[0.015,0.448]	*
w6	-0.357	1.56646	[-3.605,2.892]	
w7	-0.237	1.00601	[-2.324,1.849]	
w8	0.181	0.23672	[-0.310,0.672]	
w9	-1.285	0.86485	[-3.079,0.508]	
w10	-0.433	0.73487	[-1.957,1.091]	

表7.2 对卡特彼勒数据(caterpillar data)使用无信息先验的线性回归模型的后验均值,标准偏差,置信区间.由本书配套PMTK3的linregBayesCaterpillar生成.

权重的边缘分布为:
$p(w|D) = T(w|\hat{w}, \frac{s^2}{N-D}C, N-D)$(7.83)
上式中的$C = (X^T X)^{-1}$,$\hat{w}$是最大似然估计(MLE).这些等式的含义后面会讲.

7.6.3.3 贝叶斯和频率论相一致的样例*

(半共轭)的无信息先验很有意思,因为得到的后验等价于从频率论统计学推出来的结果(参考本书4.6.3.9).比如从等式7.83可以得到:

$$p(w_j|D) = T(w_j|\hat{w}_j, \frac{C_{jj}s^2}{N-D}, N-D)$$(7.84)

这就等价于对最大似然估计(MLE)的抽样分布,其形式如下所示(Rice 1995, p542), (Casella and Berger 2002, p554):

$$\frac{w_j - \hat{w}_j}{s_j} \sim t_{N-D}$$(7.85)

其中的:

$$s_j = \sqrt{\frac{s^2 C_{jj}}{N-D}}$$(7.86)

是估计参数的标准差.(本书6.2有讲到取样分布.)结果就是对参数的频率论的置信区间和贝叶斯边缘置信区间在这个样本中是一样的.

还拿卡特彼勒数据集为例(Marin and Robert 2007).(这个数据集的细节含义并不重要.)可以使用等式7.84来计算回归系数的后验均值/标准差/95%置信区间.结果如表7.2所示.很明显这些95%置信区间等价于使用标准频率论方法计算得到的95%置信区间(这部分代码参考本书配套PMTK3的linregBayesCaterpillar).
还可以使用边缘厚颜来计算回归参数是否显著(significantly)远离0.一个不太正规(根本不用到决策规则)的方法就是检查其95%置信区间是否排除了0.从表7.2可以得知通过这个可以衡量得到系数0,1,2,4,5都是显著的,所以加了个小星星.很容易验证这些结果和标准频率论软件得到的p值为5%的结果一样.

对于某些读者而言,可能贝叶斯方法和频率论得到结果的对应关系很让人惊讶,本书6.6当中还提到过频率论方法中的各种问题.另外还要注意到当NN的情况.)

7.6.4 线性回归的经验贝叶斯方法(证据程序)

目前为止都是假设先验为已知的.本节要说的经验贝叶斯过程是用来挑选超参数的.也就是说要挑

选能够将边缘似然函数最大化的$\eta = (\alpha, \lambda)$,其中$\lambda = 1/\sigma^2$是观测噪音的精度,而α是先验精度,先验是$p(w) = N(w|0, \alpha^{-1}I)$.这也叫做证据程序(evidence procedure)(MacKay 1995b). 算法上的细节参考本书13.7.4.

证据程序可以作为交叉验证的一个替代方法.比如在图7.13(b)中所示的不同值α对应的对数边缘似然函数,以及通过优化器找到的最大值.可见在这个例子中,得到的和5叠交叉验证中的结果(如图7.12(a)中所示)一样.(在两种方法的样例中都使用了固定的$\lambda = 1/\sigma^2$,以保证可对比性.)

证据程序相比交叉验证的主要的实践优势,在本书13.7当中才更明显,到时候回对每个特征使用不同的α_j来将先验泛化扩展.这可以用于进行特征选择,使用一种叫做自动相关性判别(automatic relevancy determination,缩写为ARD)的方法来实现.作为对比,用交叉验证是没办法调节不同的D个超参数的.

对比不同类别模型的时候,证据程序也很有用,因为提供了对证据(evidence)的一个很好的估计:

$$p(D|m) = \int\int p(D|w,m)p(w|m,\eta)p(\eta|m)dwd\eta \tag{7.87}$$

$$\approx \max_{\eta} \int p(D|w,m)p(w|m,\eta)p(\eta|m)dw \tag{7.88}$$

很重要的一点是在η上进行积分,而不是任意设置,具体原因如本书5.3.2.5所示.实际上这也是我们在图5.7和5.8当中的多项回归模型中估计边缘似然函数所用的方法.本书21.5.2讲了更贝叶斯风格的方法,其中是要对η的不确定性建模,而不是计算点估计.

练习略