Lecture 9　Linear Regression

整理者：LobbyBoy* 2020年2月21日

1.　Linear Regression Problem

仍然以“银行发放信用卡”为例。此前我们解决的问题是帮助银行决定**是否**为一位申请客户提供信用卡，这是一个二元分类问题；现在我们需要帮助银行确定为某位客户提供**多少**的信用卡额度，这则是一个“回归”(regression)问题，即输出空间$\mathcal{Y}$变成了连续的实数域R。

一个简单的模型是：$\mathbf{x}=(x_0,x_1,\cdots,x_d)$表示客户的特征资料，额度$y$表示为(没有noise则等于，有noise为约等于)：

$$y \approx \sum_{i=0}^{d} w_i x_i = \mathbf{w}^T\mathbf{x}$$

即linear regression的hypothesis为(没有$sign$的perceptron)：

$$h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

线性回归模型的错误衡量方式为平方误差：

$$err(\hat{y},y) = (\hat{y}-y)^2$$

因此E_{in}与E_{out}可以表示为：

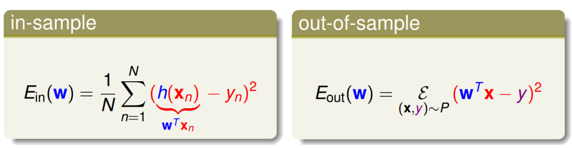

图 1: E_{in}与E_{out}

*本笔记根据台湾大学林轩田教授于线上教育平台Coursera开设的“机器学习基石”课程整理而成(课程内容见：https://www.coursera.org/learn/ntumlone-mathematicalfoundations/home/welcome)。笔记内的大多数图片来自于林老师的课程slides。感谢林老师能够将如此精彩的课程通过线上平台同所有人分享，thanks!

2. Linear Regression Algorithm

现在我们的任务是最小化$E_{in}(\mathbf{w})$。将其写成矩阵形式：

$$E_{in}(\mathbf{w}) = \frac{1}{N}||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

这是一个简单的关于$\mathbf{w}$的二次式，直接令梯度为$\mathbf{0}$可以得到：

$$(\mathbf{X}^T\mathbf{X})\mathbf{w} = \mathbf{X}^T\mathbf{y}$$

当$(\mathbf{X}^T\mathbf{X})$可逆时(可逆的状况占绝大多数，因为在实践中我们常常有$N >> d+1$→很容易会得到$d+1$个线性无关的$\mathbf{x}$→$rank(\mathbf{X}^T) = d+1$→$rank(\mathbf{X}) = d+1$→$rank(\mathbf{X}^T\mathbf{X}) = d+1$→$(\mathbf{X}^T\mathbf{X})$可逆)，便有：

$$\mathbf{w}_{LIN} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

当$(\mathbf{X}^T\mathbf{X})$不可逆时，则方程有无穷多个解，其中一个可以用“伪逆”(pseudo-inverse)$\mathbf{X}^\dagger$来表示：

$$\mathbf{w}_{LIN} = \mathbf{X}^\dagger\mathbf{y}$$

矩阵$\mathbf{X}$(不必是方阵)的pseudo-inverse记作$\mathbf{X}^\dagger$，当$(\mathbf{X}^T\mathbf{X})$可逆时等于$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$；当$(\mathbf{X}^T\mathbf{X})$不可逆时有另外一套算法。因此线性回归的解可以一般性地写成：

$$\mathbf{w}_{LIN} = \mathbf{X}^\dagger\mathbf{y}$$

综上，线性回归算法的步骤如下：

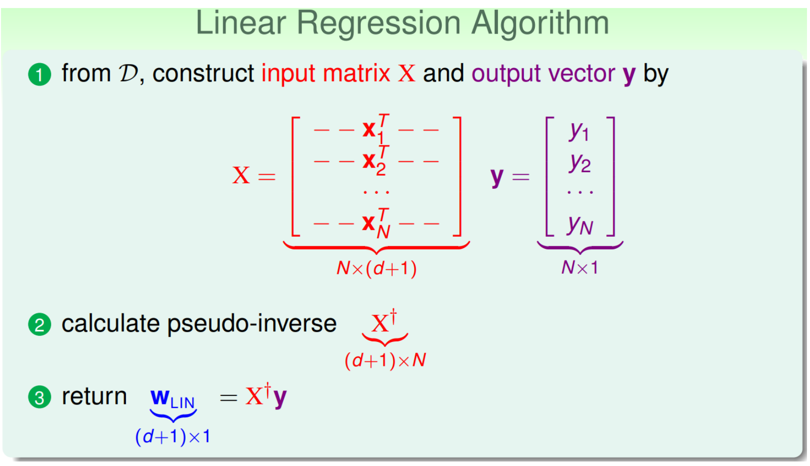

图 2: Linear Regression Algorithm

3. Generalization Issue

接下来的分析中我们都将假设$(\mathbf{X}^T\mathbf{X})$可逆。因此：

$$\mathbf{w}_{LIN} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

这种形式的解我们将其称为closed-form solution或analytic solution，因为是“一步”就得到的。我们将$\mathbf{w}_{LIN}$的预测值记作$\hat{y}$，因此有：

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}_{LIN} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \mathbf{H}\mathbf{y}$$

我们将上式中的$\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$记作$\mathbf{H}$，称为Hat matrix，因为该矩阵作用于$\mathbf{y}$使其头上多了一个“帽子”。$\mathbf{H}$有许多有趣的性质，都很容易推导。林老师的教材中提供了以下习题供大家理解$\mathbf{H}$的性质：

Exercise 3.3

Consider the hat matrix $\mathrm{H} = \mathrm{X}(\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}}$, where X is an N by $d+1$ matrix, and $\mathrm{X}^{\mathrm{T}}\mathrm{X}$ is invertible.

(a) Show that H is symmetric.

(b) Show that $\mathrm{H}^K = \mathrm{H}$ for any positive integer K.

(c) If I is the identity matrix of size N, show that $(\mathrm{I}-\mathrm{H})^K = \mathrm{I}-\mathrm{H}$ for any positive integer K.

(d) Show that $\mathrm{trace}(\mathrm{H}) = d+1$, where the trace is the sum of diagonal elements. *[Hint:* $\mathrm{trace}(\mathrm{AB}) = \mathrm{trace}(\mathrm{BA})$.*]*

图 3: $\mathbf{H}$

另外，我们也可以直观地从线性代数中space的视角来看线性回归与$\mathbf{H}$。我们希望最小化的式子如下：

$$E_{in}(\mathbf{w}) = \frac{1}{N}||\mathbf{X}\mathbf{w} - \mathbf{y}||^2$$

翻译成线性代数的语言就是：希望在矩阵$\mathbf{X}$的column space中找到一个向量(该向量可以用$\mathbf{X}$的列的线性组合$\mathbf{w}$表示出来)，该向量与$\mathbf{y}$的距离最近。因此该目标向量一定是$\mathbf{y}$在$\mathbf{X}$的column space中的projection，即下图中的$\hat{\mathbf{y}}$：

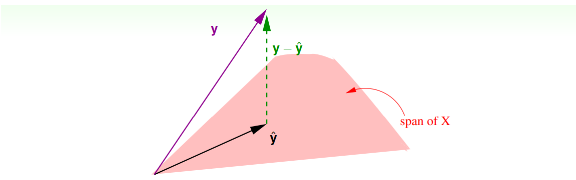

图 4: $\hat{\mathbf{y}}$ is the projection of $\mathbf{y}$ in $Col(\mathbf{X})$

结合我们代数解最优化问题的结果，上面过程中对$\mathbf{y}$进行投影动作的投影矩阵即为Hat matrix $\mathbf{H}$，因为：

$$\mathbf{H}\mathbf{y} = \hat{\mathbf{y}}$$

下面我们将证明一个重要的等式：

$$\overline{E_{\text{in}}} = \underset{\mathcal{D}\sim P^N}{\mathcal{E}}\left\{E_{\text{in}}(\mathbf{w}_{\text{LIN}} \text{ w.r.t. } \mathcal{D})\right\} \overset{\text{to be shown}}{=} \text{noise level}\cdot\left(1-\frac{d+1}{N}\right)$$

图 5: 经验误差的期望

上式左侧的$\bar{E}_{in}$表示$E_{in}(\mathbf{w}_{LIN})$的期望值。可以对其求期望是因为不同的训练数据集$\mathcal{D}$会得到不同的$\mathbf{w}_{LIN}$，也就有不同的$E_{in}(\mathbf{w}_{LIN})$。上式右侧是误差水平与一个小于1的数的乘积。在引入noise之后，我们假设y是由确定的线性部分$\mathbf{w}^{*T}\mathbf{x}$与随机的噪声部分$\epsilon$构成的，即：

$$y = \mathbf{w}^{*T}\mathbf{x} + \epsilon$$

或对于训练数据来说:

$$\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{e}$$

除了纯数学推导外，我们仍然可以借助线性代数与之前的关于space的图来直观地解决该问题：

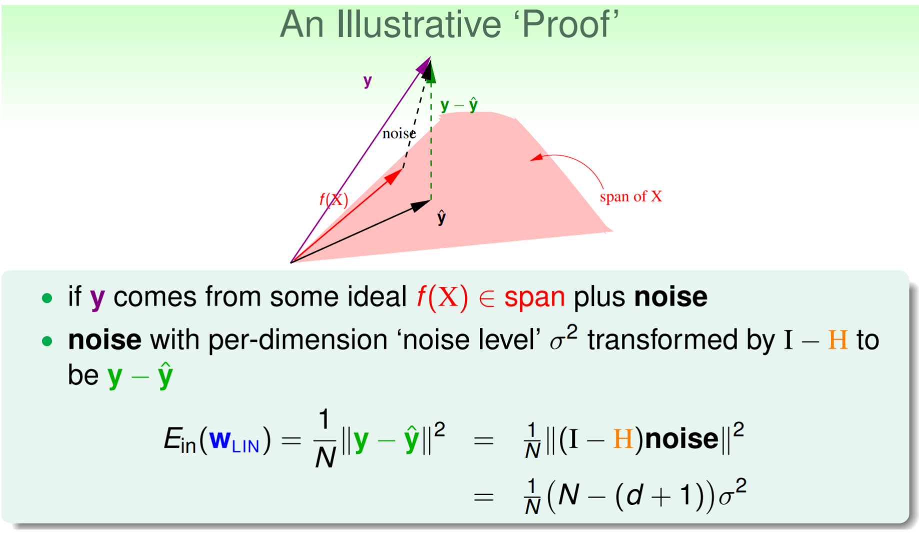

图 6: $\mathbf{y} = \mathbf{X}\mathbf{w}^* + \mathbf{e}$

从上图中我们可以看到，$\mathbf{y}-\hat{\mathbf{y}}$可以表示为$\mathbf{y}$与其正交投影之差，也可以表示为noise与其正交投影之差(同一个直角边)。因此有：

$$E_{in}(\mathbf{w}_{LIN}) = \frac{1}{N}||\mathbf{y}-\hat{\mathbf{y}}||^2 = \frac{1}{N}||(\mathbf{I}-\mathbf{H})\mathbf{e}||^2 = \frac{1}{N}\Big(\mathbf{e}^T(\mathbf{I}-\mathbf{H})\mathbf{e}\Big)$$

我们将RHS的矩阵乘法写成累加求和的形式:

$$\frac{1}{N}\Big(\mathbf{e}^T(\mathbf{I}-\mathbf{H})\mathbf{e}\Big) = \frac{1}{N}\Big(\sum_{i=1}^{N}\lambda_{ii}\epsilon_i^2 + O(\sum_{i,j,i\neq j}\lambda_{ij}\epsilon_i\epsilon_j)\Big)$$

其中λ_{ij}表示矩阵$\mathbf{I}-\mathbf{H}$第i行第j列的元素。假设noise的均值为0，方差为σ，且noise间独立，即：

$$E(\epsilon_i)=0,\quad E(\epsilon_i^2)=\sigma^2,\quad E(\epsilon_i\epsilon_j)=0\ for\ i\neq j$$

因此我们有：

$$\bar{E}_{in} = E\Big[\frac{1}{N}\Big(\sum_{i=1}^{N}\lambda_{ii}\epsilon_i^2 + O(\sum_{i,j,i\neq j}\lambda_{ij}\epsilon_i\epsilon_j)\Big)\Big] = \frac{1}{N}\sum_{i=1}^{N}\lambda_{ii}E(\epsilon_i^2) = \frac{\sigma^2}{N}\sum_{i=1}^{N}\lambda_{ii} = \frac{\sigma^2}{N}trace(\mathbf{I}-\mathbf{H})$$

通过exercise 3.3我们知道，$trace(\mathbf{I}-\mathbf{H}) = N-(d+1)$。因此：

$$\bar{E}_{in} = \sigma^2(1-\frac{d+1}{N})$$

同样的我们可以计算出$\bar{E}_{out}$：

$$\bar{E}_{out} = \sigma^2(1+\frac{d+1}{N})$$

将$\bar{E}_{in}$、$\bar{E}_{out}$以样本量N为横轴，误差的期望值为纵轴作图，得到学习曲线(learning curve)：

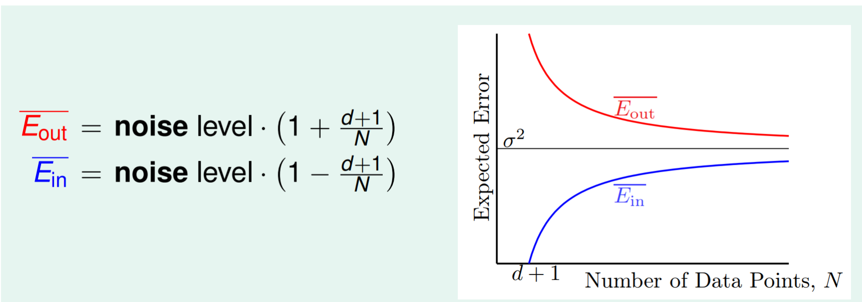

图 7: learning curve

从上图中我们可以看出两件事：第一，$E_{out}-E_{in}$的期望值为$\sigma^2(2\cdot\frac{d+1}{N})$；第二，随着$N$的增大，$E_{out}-E_{in}$的期望值逐渐趋于0。因此，我们得到了类似二元分类VC bound的

泛化保证。

4. Linear Regression for Binary Classification

通过解线性回归问题，我们得到了一个close-form solution，十分方便。那么我们自然会想，能不能将线性回归算法用于线性分类呢？毕竟二元分类的输出空间$\mathcal{Y}=\{-1,+1\}\subset R$，将回归用于分类，相当于直接去拟合-1或+1，那么我们可能也会在真实值为+1时得到某个正的预测值，在真实值为-1时候=得到某个负的预测值。因此我们心中已经有了一个算法的雏形：①在二元分类训练数据集$\mathcal{D}$上跑线性回归算法，得到$\mathbf{w}_{LIN}$；②回传hypothesis：$g(\mathbf{x})=sign\left(\mathbf{w}_{LIN}^T\mathbf{x}\right)$。但现在的问题是，这样的算法有理论保证吗？

我们在样本$(\mathbf{x},y)$上观察0/1误差$err_{0/1}$与平方误差err_{sqr}的关系：

$$err_{0/1}=I\Big[sign(\mathbf{w}^T\mathbf{x})\neq y\Big],\quad err_{sqr}=(\mathbf{w}^T\mathbf{x}-y)^2$$

作图，分真实值为+1与真实值为-1两种情况：

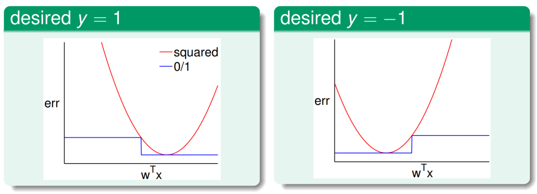

图 8: upper bound

可见，平方误差是0/1误差的上界，即：

$$err_{0/1}\leq err_{sqr}$$

在二元分类中，VC bound向我们保证了在样本量足够大时泛化误差与经验误差比较接近，如果我们将0/1误差下的$E_{in}(\mathbf{w})$做得足够小，那么我们就拿到了合适的hypothesis。现在我们知道平方误差是0/1误差的上界，那么如果我们可以把平方误差做得很小，那么必然会有相应的0/1误差很小，那么该hypothesis也是合适的，即：

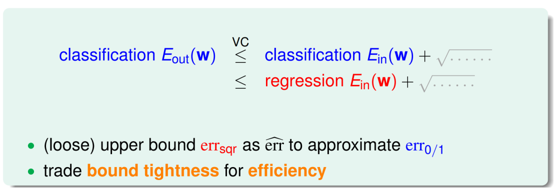

图 9: upper bound

因此，我们可以将线性回归算法用于线性分类问题。特别的，我们可以将通过线性

回归得到的$\mathbf{w}_{LIN}$作为baseline classifier，或者作为PLA/Pocket演算法的初始向量(initial vector)。