

Lesson 2 Linear and Polynomial Regression

线性回归的问题陈述

对于训练集 $D = \{\mathbf{x}_i, y_i\}_{i=1}^N$ ，学习一个模型，根据给定输入得到预测输出
模型学习公式 Hypothesis Function

$$y = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{j=0}^N w_j x_j$$

权重 \mathbf{w} 决定了特征 \mathbf{x} 在预测标签 y 中的重要程度。

实际应用当中常设 $x_0 = 1$ ，对应的 w_0 称为偏差/bias，常用 b 表示。

决定权重 \mathbf{w} ：使得均方差 mean square error/MSE 最小，物理含义是指让所有的点离最终的预测平面的平均距离最小，即使以下方程有最小值

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

要使得以上方程最小，可以令其梯度最小，即令

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = 0$$

解以上方程，可得

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

$$\text{其中 } \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \dots \\ \mathbf{x}_n^T \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$$

称为普通最小二乘解 ordinary least squares /OLS solution

线性回归的概率解释

将 y 看做随机变量，因而有

$$y = \mathbf{w}^T \mathbf{x} + \epsilon = \sum_{j=0}^D w_j x_j + \epsilon$$

其中 ϵ 代表由测量或未知特征引起的误差，假设这些误差都是独立同分布 independent and identically distributed/iid 的，则根据中心极限定律，其服从均值为 0，方差为 σ^2 的高斯分布

$$\epsilon \sim N(0, \sigma^2)$$

因而

$$(y - \mathbf{w}^T \mathbf{x}) \sim N(0, \sigma^2)$$

令模型参数 θ 包括 \mathbf{w} 以及 σ ，从而条件概率 $p(y|\mathbf{x}, \theta)$ 满足

$$p(y|\mathbf{x}, \theta) = N(y|\mu(\mathbf{x}), \sigma^2)$$

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

因而给定一个输入 \mathbf{x} 时，可以得到一个 y 的高斯分布

为了得到 y 的点估计，可以使用均值，如

$$\hat{y} = \mu(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

所以给定数据集 D 关于参数 θ 的对数似然函数是

$$\begin{aligned} l(\theta|D) &= \log p(D|\theta) \\ &= \sum_{i=1}^N \log p(\epsilon_i|\theta) \\ &= \sum_{i=1}^N \log p(y_i|\mathbf{x}_i, \theta) \\ &= \sum_{i=1}^N \log \left[\frac{1}{\sqrt{2\pi}\theta} \exp \left(-\frac{(y_i - \mathbf{w}^T \mathbf{x}_i)^2}{2\sigma^2} \right) \right] \\ &= -\frac{N}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \end{aligned}$$

因为 \log 函数是单调递增函数，因而 $p(D|\theta)$ 和 $\log p(D|\theta)$ 拥有相同的最大值点。

所以为了得到最大似然估计 MLE，可以选择最大化 $l(\theta|D)$ ，也可以选择最小化

负对数似然函数 Negative Log-likelihood/NLL，又被称为交叉熵 cross-entropy

$$NLL(\theta|D) = -l(\theta|D) = \frac{N}{2} \log(2\pi\sigma^2) + \frac{1}{2\sigma^2} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

因而对于 \mathbf{w} 来说，给定 σ ，最小化 $NLL(\theta|D)$ 等同于最小化 $\sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$ ，即等同于最小化

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

从而不论从最小化均方差 MSE 或者是最小化负对数似然函数 NLL 都会得到一样的最终结果

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

多项式回归

对于简化的线性回归

$$y = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} = \sum_{j=0}^N w_j x_j$$

可以使用一个非线性函数 $\phi(\mathbf{x})$ 来代替 \mathbf{x} ，从而使得原方程可以用来拟合非线性的模型

$$y = f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

这一过程被称为基础函数扩展 basis function expansion，函数 ϕ 被称为特征映射

如果采用以下的多项式特征映射 polynomial feature mapping

$$\phi(x) = [1, x_1, x_2, \dots, x_n, x_1^2, x_2^2, \dots, x_n^2]$$

可以得到多项式回归。例如当 $x = (x_1, x_2), d = 2$ 时，有

$$y = f(x) = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_1x_2 + w_5x_2^2$$

因为基本特征 x 是固定个数的，问题在于如何选择 d

概念：假设空间 Hypothesis Space：一个机器学习算法的解空间

容量 Capacity：假设空间的 size

对于多项式回归， d 越大，模型容量越大；模型容量越大，证明模型对训练样本的拟合程度越好。

过拟合 Overfitting 和欠拟合 Underfitting

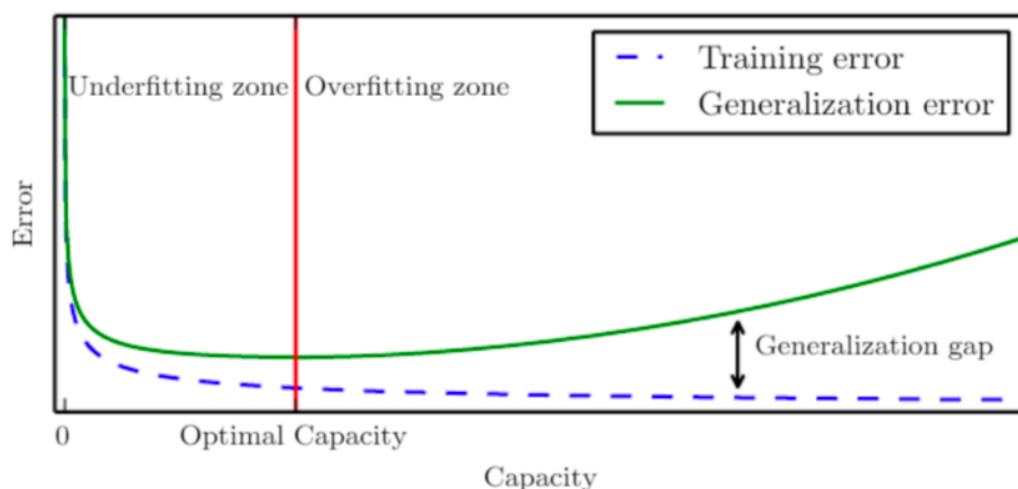
生成误差 Generative Error：模型训练目标是让其在训练样本中尽可能的表现优异，但模型最终需要在未经训练的测试样本上也能成功运作，这个过程称为 Generalization，这个过程产生的误差称之为 Test Error/Generative Error，定义为

$$J^{(test)}(w) = \frac{1}{N^{(test)}} \left\| y^{(test)} - X^{(test)}w \right\|_2^2$$

与此相对的，训练误差 Training Error 定义为

$$J^{(train)}(w) = \frac{1}{N^{(train)}} \left\| y^{(train)} - X^{(train)}w \right\|_2^2$$

Test Error 和 Training Error 是紧密相关的，因为训练集和测试集都是总数据集独立同分布的样本。一般情况下，测试误差常大于训练误差，因为训练过程是为了使训练误差最小化。因而为了解决该问题，在训练过程当中除了让训练误差尽可能小外，还要使训练误差以及测试误差的 **gap** 尽可能小。



红线代表最佳容量；

红线左边的区域 **capacity** 过小，训练和测试误差均太大，是欠拟合；
红线右边的区域 **capacity** 过大，虽然训练误差越来越小，但 **gap** 越来越大，是过拟合。

如上文所述，多项式回归的模型容量 **capacity** 常由 **d** 决定，而 **d** 是一种超参数 Hyperparameter。实际上各种模型的容量经常由超参数决定。

验证 Validation 是一种常用的确定超参数的方法，它将训练集随机分为两个不相交的子集，一个子集仍然称为训练集，另外一个子集称为验证集 Validation set。
例如在多项式回归模型当中，为了确定参数 **d**：

1. 给出一个 **d** 可能的值的集合{**d**}；
2. 对于集合当中的每个 **d**，在相同的训练集上做训练，之后测试不同的 **d** 在相同的验证集上的误差，称为 Validation error；
3. 选择具有最小验证误差的 **d**。

关于如何将原训练集分为新的训练集和一个验证集：

1. 一般而言，训练集越大，假设越好；验证集越大，误差估计越精确；
2. 一般选择 20%的原训练数据作为验证集，剩余的作为新的训练集。

交叉验证 Cross Validation：当数据有限时，若仍继续保留一部分数据作为验证集将会使训练样本更少，误差估计将会有较大的方差，因而此时要采用交叉验证的方法：

1. **N** 个可用的训练样本将被分为 **k** 个不相交的子集，每个子集大小为 **N/k**；
2. 学习过程将会 **run** 上 **k** 次，每一次采用其中一个子集作为验证集，剩下的 **k-1** 个子集作为训练集；
3. 将 **k** 次学习过程的结果作为模型的误差估计和准确率；
4. 通常情况下 **k = 10**。

除了使用 **Validation** 来从一堆可能的 **d** 中挑出一个外，还可以采用另外一种办法：使用正则化方法 Regularization，从一个较大的 **d** 开始，从其中较大的假设空间当中选取一个合适的解：

对于线性回归而言，其误差函数为

$$J(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2$$

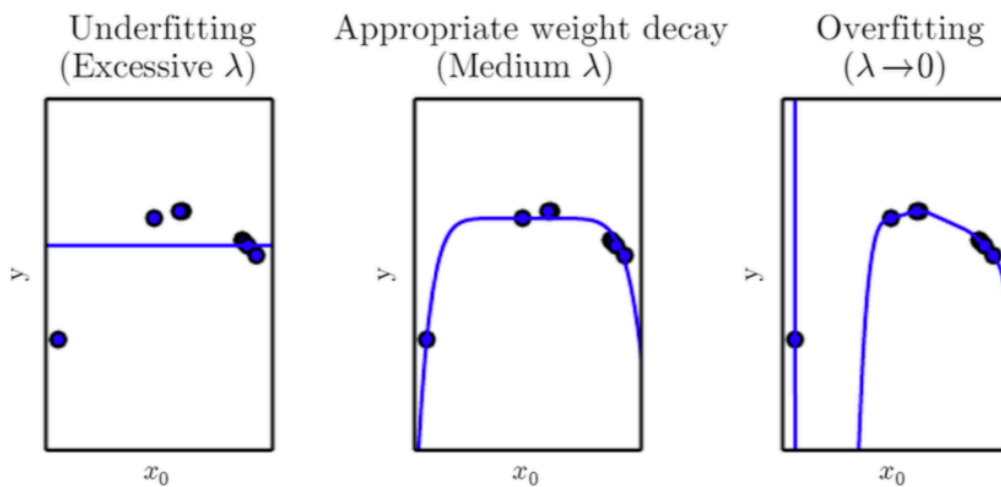
多项式回归中为了匹配 **x** 的 0 次项，加上一个 **w₀**：

$$J(\mathbf{w}, w_0) = \frac{1}{N} \sum_{i=1}^N (y_i - (w_0 + \mathbf{w}^T \phi(\mathbf{x}_i)))^2$$

加上正则化的误差函数为：

$$J(\mathbf{w}, w_0) = \frac{1}{N} \sum_{i=1}^N (y_i - (w_0 + \mathbf{w}^T \phi(\mathbf{x}_i)))^2 + \lambda \|\mathbf{w}\|_2^2$$

其中 $\lambda \geq 0$ 是一个提前选择好的值，用于控制我们对较小权重偏爱的强度。最小化以上的误差函数会使我们得到依赖较少特征的解，这个过程称之为权重衰退 weight decay。不同的 λ 影响如下：



使用以上正则化后的误差函数的模型称之为脊回归 Ridge Regression，同样使该误差函数的梯度为 0，可以得到

$$\hat{\mathbf{w}}_{ridge} = (\lambda \mathbf{I}_k + \mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$$

这个解称为惩罚最小二乘解 penalized least squares solution

其中 \mathbf{I}_k 是 k 维单位矩阵； λ 越大，权重 \mathbf{w} 越小。脊回归通过收缩回归当中较大的参数来避免过拟合。

补充：least absolute shrinkage and selection operator/LASSO 方法强迫某些参数等于 0，有效的选取一个参与特征较少、较为简单的模型。相比于脊回归使用 L2 正则化，LASSO 采用 L1 正则化，其误差函数为

$$J(\mathbf{w}, w_0) = \frac{1}{N} \sum_{i=1}^N (y_i - (w_0 + \mathbf{w}^T \phi(\mathbf{x}_i)))^2 + \lambda \|\mathbf{w}\|_1$$