LiVE Group
视觉智能与学习研究中心

机器学习（第4讲）

主讲：张 磊

E-mail: leizhang@cqu.edu.cn
Lab Website: http://www.leizhang.tk

第四章： 线性建模

线性建模

在人工智能/机器学习中，核心的本质问题是推断属性变量（自变量）与响应变量（因变量）之间的函数关系，使得给定任何一个属性集合，可以通过该函数关系，预测其响应。

高数中的自变量x与因变量y，有以下函数关系

$$y=h(x)$$

线性建模的目的：

当给定一组（x,y）集合时，如何估计函数h(.)的表达式？

线性建模

例1：建立一个能够执行疾病诊断的模型h(.)。

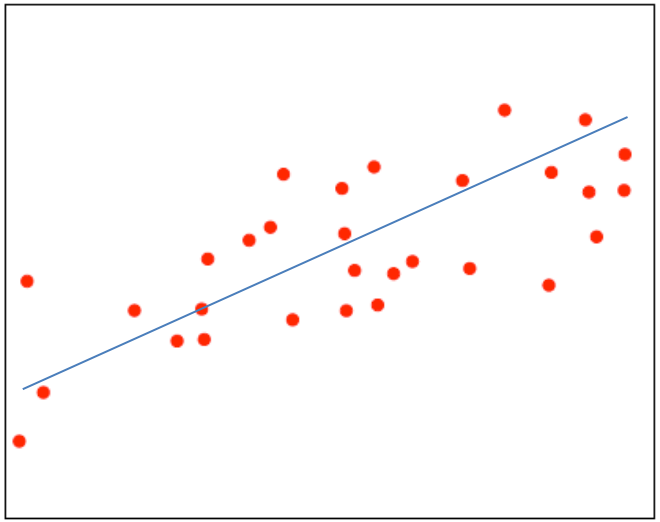
已知条件：已知疾病状态（健康、患病）的多个患者，以及对这些患者测量的属性（血压、心率、体重等）

例2：通过某用户在电影网页上的点击情况，预测该用户的喜好。

已知条件：已知被点击的电影种类（喜剧、动作、恐怖等），以及该用户对每个种类的点击次数。

模型假设

在建立回归模型时，模型假设很重要。



x	y
0.86	2.49
0.09	0.83
-0.85	-0.25
0.87	3.10
-0.44	0.87
-0.43	0.02
-1.10	-0.12
0.40	1.81
-0.96	-0.83
0.17	0.43

线性建模

线性建模是机器学习中最直接的学习问题：学习 属性与响应之间的线性关系。

例：建立苹果的重量预测模型：单属性问题（一维问题）

已知条件：已知N个苹果的重量$t_1, t_2, \cdots, t_N$，以及N个苹果的水平宽度$x_1, x_2, \cdots, x_N$。

任务：估计苹果的重量预测线性模型$t = h(x) = \omega_0 + \omega_1 x$

什么是好的准则？

为了选择某种方式下最好的ω_0和ω_1值，使得该直线能够尽可能与所有数据点接近。

度量“好”的方法是采用“平方差”，即预测的苹果重量h与实际重量t之间的平方差（为什么要平方？），定义为

$$\left(t_n - h(x_n; \omega_0, \omega_1)\right)^2$$

该式子表示第n个苹果的预测误差，该值越小，说明模型$h(\cdot)$在x_n处的值越接近t_n。

该度量有个专业称呼：“平方损失函数”(squared loss function)，因此，第n个苹果（样本）的损失可以定义为

$$L\left(t_n, h(x_n; \omega_0, \omega_1)\right) = \left(t_n - h(x_n; \omega_0, \omega_1)\right)^2$$

什么是好的模型？

对于全部的N个苹果，我们希望所有苹果的损失都小，因此，有总体平均损失：

$$L=\frac{1}{N}\sum_{n=1}^{N}L\left(t_n,h(x_n;\omega_0,\omega_1)\right)=\frac{1}{N}\sum_{n=1}^{N}\left(t_n-h(x_n;\omega_0,\omega_1)\right)^2$$

我们建模的目的是求得最佳的ω_0,ω_1,使得平均损失L达到最小。

上面的问题可以写成专业化的机器学习模型：

$$\min_{\omega_0,\omega_1}L=\frac{1}{N}\sum_{n=1}^{N}\left(t_n-h(x_n;\omega_0,\omega_1)\right)^2$$

或者$\{\omega_0,\omega_1\}=\arg\min_{\omega_0,\omega_1}L=\frac{1}{N}\sum_{n=1}^{N}\left(t_n-h(x_n;\omega_0,\omega_1)\right)^2$

求解过程

将损失函数展开

$$L = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - h(x_n;\omega_0,\omega_1)\right)^2$$

$$=$$

$$\vdots$$

$$= \frac{1}{N}\sum_{n=1}^{N}\left(\omega_1^2 x_n^2 + 2\omega_1 x_n(\omega_0 - t_n) + \omega_0^2 - 2\omega_0 t_n + t_n^2\right)$$

计算损失函数L对ω_0和ω_1的偏导数，并令导数为0，可得出损失最小值时的两个估计参数。

预测过程(测试推理阶段)

通过极值求解，可获得ω_0和ω_1的最佳估计值$\hat{\omega}_0$和 $\hat{\omega}_1$，从而可以写出函数表达式

$$t = \hat{\omega}_0 + \hat{\omega}_1 x = -0.8 + 0.19x$$

此时，对于一个新的样本（苹果），经过测量，发现其宽度为12cm，那该苹果的重量应该是多少？

$$t = -0.8 + 0.19 \times 12 = 1.48kg$$

模型的矩阵求解

继续考虑刚才的模型

$$t_n = h(x_n) = \omega_0 + \omega_1 x_n$$

可以写成

$$t_n = h(\mathbf{x}_n) = (\omega_0, \omega_1)\begin{pmatrix}1\\ x_n\end{pmatrix} = \mathbf{w}^{\mathrm{T}}\mathbf{x}_n$$

现将N个样本的属性向量$\mathbf{x}_n$合并成一个矩阵$\mathbf{X}$

$$\mathbf{X} = \begin{pmatrix}\mathbf{x}_1^{\mathrm{T}}\\ \vdots\\ \mathbf{x}_N^{\mathrm{T}}\end{pmatrix} = \begin{pmatrix}1 & x_1\\ 1 & x_2\\ & \vdots\\ 1 & x_N\end{pmatrix}$$

也将N个样本的目标t_n合并成一个向量

$$\mathbf{t} = (t_1, t_2, \cdots, t_N)^{\mathrm{T}}$$

模型的矩阵求解

原损失函数

$$L = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - h(x_n;\omega_0,\omega_1)\right)^2$$

可以完全等价地转为向量和矩阵的函数，即

$$L = \frac{1}{N}(\mathbf{t}-\mathbf{X}\mathbf{w})^{\mathrm{T}}(\mathbf{t}-\mathbf{X}\mathbf{w})$$

$$= \frac{1}{N}\left(\mathbf{t}^{\mathrm{T}} - (\mathbf{X}\mathbf{w})^{\mathrm{T}}\right)(\mathbf{t}-\mathbf{X}\mathbf{w})$$

$$= \frac{1}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathrm{T}}\mathbf{t}$$

此时，可以通过矩阵向量求导，直接获得$\mathbf{w}$,而无需对ω_0,ω_1分别求偏导。

模型的矩阵求解

$$L = \frac{1}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathrm{T}}\mathbf{t}$$

此时，可以通过矩阵向量求导，可以直接获得**w**,而无需对ω_0, ω_1分别求偏导。

因为$\frac{\partial L}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial L}{\partial \omega_0} \\ \frac{\partial L}{\partial \omega_1} \end{bmatrix} = \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} - \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{t} = 0$

可以推出

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{w} = \mathbf{X}^{\mathrm{T}}\mathbf{t}$$

那么$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$（完毕）

模型的矩阵求解

现在已经获得w的解析表达式

$$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t} = \begin{bmatrix}\omega_0 \\ \omega_1\end{bmatrix}$$

例：给出3个样本点组成的集合

$$\mathbf{X} = \begin{bmatrix}1 & 1 \\ 1 & 3 \\ 1 & 5\end{bmatrix}, t = \begin{bmatrix}4.8 \\ 11.3 \\ 17.2\end{bmatrix}$$

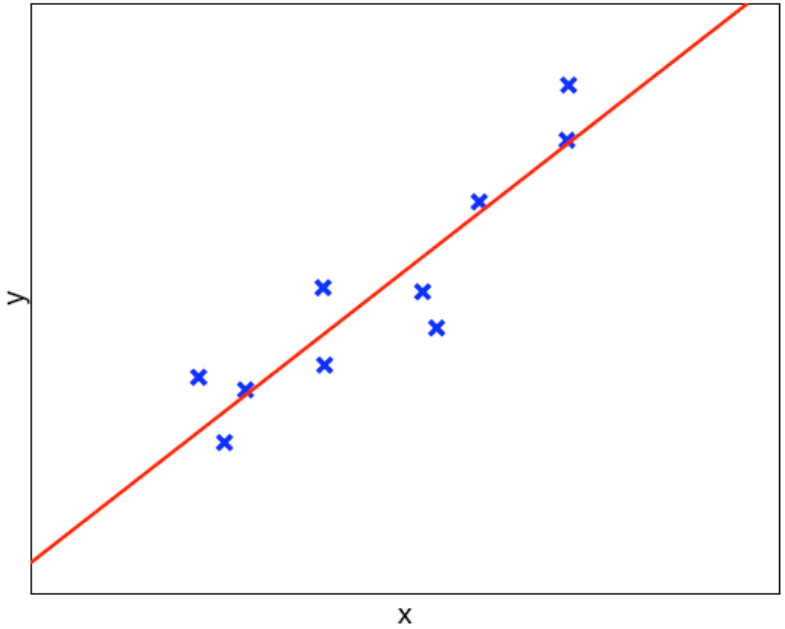
如何建立线性模型的表达式？

预测：给定一个属性$\mathbf{x}_{new} = [1\ x_{new}]^{\mathrm{T}}$的新向量，其预测公式：

$$t_{new} = \omega_0 + \omega_1 x_{new} = [\omega_0\ \ \omega_1][1\ x_{new}]^{\mathrm{T}} = \mathbf{w}^{\mathrm{T}}\mathbf{x}_{new}$$

Example: Data and best linear hypothesis

$$y = 1.60x + 1.05$$



线性建模-多维问题

多维问题是机器学习在实际应用中的普遍问题，即有多个属性的情况。

例：建立一个能够执行疾病诊断的线性模型：多属性问题（多维）

已知条件：已知疾病状态（健康、患病）的N个患者，以及每个患者的血压值x_1、心率x_2、体重x_3等，通过多个方面的因素，来预测患者的身体状况（响应）。

线性建模-多维问题

如果有d个属性，此时的属性向量$\mathbf{x}_n$，可以表示为

$$\mathbf{x}_n = \begin{pmatrix} 1 \\ x_{1,n} \\ \vdots \\ x_{d,n} \end{pmatrix}$$

注意：ω_0也叫作“偏置项”。样本$\mathbf{x}_n$和系数向量$\mathbf{w}$是d+1维。

那么，多维下的预测可以写成

$$t_n = \omega_0 + \omega_1 x_{1,n} + \omega_2 x_{2,n} + \omega_3 x_{3,n} + \cdots + \omega_d x_{d,n} = \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \vdots \\ \omega_d \end{bmatrix}^{\mathrm{T}} \begin{pmatrix} 1 \\ x_{1,n} \\ \vdots \\ x_{d,n} \end{pmatrix}$$

$$= \mathbf{w}^{\mathrm{T}} \mathbf{x}_n$$

线性建模-多维问题：模型的矩阵求解

现将N个样本的属性向量$\mathbf{x}_n = \left[x_{1,n}, x_{2,n}, \cdots, x_{d,n}\right]^{\mathrm{T}}$合并成一个矩阵$\mathbf{X}$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{x}_N^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{1,1}, x_{2,1}, \cdots, x_{d,1} \\ x_{1,2}, x_{2,2}, \cdots, x_{d,2} \\ \vdots \\ x_{1,N}, x_{2,N}, \cdots, x_{d,N} \end{pmatrix} \in \mathfrak{R}^{N \times d}$$

也将N个样本的目标t_n也合并成一个向量

$$\mathbf{t} = (t_1, t_2, \cdots, t_N)^{\mathrm{T}}$$

对于{健康、患病}两种响应，通常可以用+1和-1表示目标t.

线性建模-多维问题：模型的矩阵求解

损失函数

$$L = \frac{1}{N}(\mathbf{Xw} - \mathbf{t})^{\mathrm{T}}(\mathbf{Xw} - \mathbf{t}) = \frac{1}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{Xw} - \frac{2}{N}\mathbf{w}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{t} + \frac{1}{N}\mathbf{t}^{\mathrm{T}}\mathbf{t}$$

通过计算$\frac{\partial L}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial L}{\partial \omega_1} \\ \frac{\partial L}{\partial \omega_2} \\ \vdots \\ \frac{\partial L}{\partial \omega_d} \end{bmatrix} = \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{Xw} - \frac{2}{N}\mathbf{X}^{\mathrm{T}}\mathbf{t} = 0$

可以推出

$$\mathbf{X}^{\mathrm{T}}\mathbf{Xw} = \mathbf{X}^{\mathrm{T}}\mathbf{t}$$

那么$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$（完毕）

线性建模-多维问题：模型的矩阵求解

举例：设有4个患者，测量血压值、心率 、体重三个属性。

现将4个样本的属性向量$\mathbf{x}_n = [x_{1,n}, x_{2,n}, x_{3,n}]^{\mathrm{T}}$合并成一个矩阵$\mathbf{X}$

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1^{\mathrm{T}} \\ \mathbf{x}_2^{\mathrm{T}} \\ \mathbf{x}_3^{\mathrm{T}} \\ \mathbf{x}_4^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} x_{1,1}, x_{2,1}, x_{3,1} \\ x_{1,2}, x_{2,2}, x_{3,2} \\ x_{1,3}, x_{2,3}, x_{3,3} \\ x_{1,4}, x_{2,4}, x_{3,4} \end{pmatrix} = \begin{pmatrix} 80,70,50 \\ 150,100,70 \\ 100,80,60 \\ 60,120,40 \end{pmatrix} \in \Re^{4\times 3}$$

也将N个样本的目标t_n也合并成一个向量

$$\mathbf{t} = (t_1, t_2, t_3, t_4)^{\mathrm{T}} = (1, -1, 1, -\mathbf{1})^{\mathrm{T}}$$

对于{健康、患病}两种响应，分别用+1和-1表示.

线性建模-多维问题：模型的矩阵求解

一般，X中的数值很大，需要对属性值进行归一化处理，以便于求解和计算。归一化方法，通常除以属性最大值实现。

$$\mathbf{X}=\begin{pmatrix}80,70,50\\150,100,70\\100,80,60\\60,120,40\end{pmatrix}\rightarrow\begin{pmatrix}\frac{80}{150},\frac{70}{120},\frac{50}{70}\\\frac{150}{150},\frac{100}{120},\frac{70}{70}\\\frac{100}{150},\frac{80}{120},\frac{60}{70}\\\frac{60}{150},\frac{120}{120},\frac{40}{70}\end{pmatrix}\rightarrow\begin{pmatrix}\frac{8}{15},\frac{7}{12},\frac{5}{7}\\1,\frac{5}{6},1\\\frac{2}{3},\frac{2}{3},\frac{6}{7}\\\frac{2}{5},1,\frac{4}{7}\end{pmatrix}$$

利用解析形式$\mathbf{w}=\left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$，很容易计算出模型的参数w

普通最小二乘线性回归总结：

- 模型的最优解，可以通过最小化误差的平方和（损失函数）进行计算。

$$L = \frac{1}{N}\sum_{n=1}^{N}\left(t_n - h(x_n;\omega_0,\omega_1)\right)^2$$

- 最小二乘法回归算法存在闭解（解析解、精确解）

$\mathbf{w} = \left(\mathbf{X}^{\mathrm{T}}\mathbf{X}\right)^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{t}$，其中$\mathbf{X}$是样本数据矩阵，$\mathbf{t}$是目标值矢量。

普通最小二乘线性回归与极大似然估计的关系？

最小二乘解即为极大似然解！

几个概念回顾：

大数定律(伯努利)

其通俗表示：当样本足够多时，样本均值收敛于数学期望（依概率收敛）。

1）伯努利大数定律

描述：N次伯努利试验中，事件1发生的频次为n，而事件1每次发生的概率为p，则对任意小的正数ε，有

$$\lim_{N\to\infty} P\left(\left|\frac{n}{N}-p\right|<\varepsilon\right)=1$$

通俗的表述：当执行无穷次试验，事件1发生频率几乎接近其发生的概率。

（抛硬币试验即是伯努利试验，只有发生或不发生）

几个概念回顾：

大数定律(伯努利)

其通俗表示：当样本足够多时，样本均值收敛于数学期望（依概率收敛）。

2）辛钦大数定律

描述：设$\{X_i, i = 1, \dots, N\}$中为独立同分布的随机变量序列，若期望存在为μ，则服从大数定律，有

$$\lim_{N\to\infty} P\left(\left|\frac{1}{N}\sum_i X_i - \mu\right| < \varepsilon\right) = 1$$

通俗的表述：当N趋于无穷大时，序列的均值几乎接近其期望值。

几个概念回顾：

大数定律(伯努利)

其通俗表示：当样本足够多时，样本均值收敛于数学期望。

3）切比雪夫大数定律

描述：与辛钦大数定律类似，区别在于其并不需要假设$\{X_i, i = 1, \dots, N\}$为同分布的随机变量序列。

更加具有一般性。

几个概念回顾：

中心极限定理(棣莫佛)

考虑独立同分布的随机变量Y1, Y2, ..., Yn的集合，它们服从一任意的概率分布(均匀分布、指数分布等)，均值为μ, 有限方差为σ^2。定义样本均值

$$\overline{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y_i$$

则当$n \to \infty$时，$\frac{\overline{Y}_n - \mu}{\frac{\sigma}{\sqrt{n}}}$，服从一正态分布$N(0,1)$，均值为0且标准差为1。

也即：$\overline{Y}_n \sim N(\mu, \frac{\sigma^2}{n})$

问题：均值为何为0？即$n \to \infty$, $\overline{Y}_n = \mu$?

普通最小二乘线性回归与极大似然估计的关系：

- 最小二乘实质是极大似然解。

已知N个训练样本的集合$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \ldots, (x_N, y_N)$。假设样本概率分布之间是相互条件独立的。

那么联合似然概率：

$$P(y|x;w) = \prod_{n=1}^{N} P(y_n|x_n;w) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_n - w^T x_n)^2}{2\sigma^2}}$$

极大似然估计，就是找出使得样本集的似然概率最大的一组参数**w**；

如何理解这句话？

普通最小二乘线性回归与极大似然估计的关系：

- 最小二乘实质是极大似然解。

根据之前的模型，我们希望预测值与真实值之间的误差尽可能小。令：

$$y_n = w^T x_n + \xi_n$$

其中，ξ_n是一个随机变量，假设$\xi_n \sim \mathcal{N}(0, \sigma^2)$（大数定律和中心极限定理）。

那么$y_n \sim \mathcal{N}(w^T x_n, \sigma^2)$，因此：

$$P(y_n | x_n; w) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_n - w^T x_n)^2}{2\sigma^2}}$$

普通最小二乘线性回归与极大似然估计的关系：

- 最小二乘实质是极大似然解(极大似然估计MLE: Maximum likelihood estimation)。

$$\max_{w} L(w) = P(y|x;w) = \prod_{n=1}^{N} P(y_n|x_n;w) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_n - w^T x_n)^2}{2\sigma^2}}$$

上述最大化问题等效于似然函数L(w)的对数！

普通最小二乘线性回归与极大似然估计的关系：

- 最小二乘实质是极大似然解(极大似然估计MLE: Maximum likelihood estimation)。

$$\max_w L(w) \triangleq \max_w logL(w)$$

$$= log\prod_{n=1}^{N} P(y_n|x_n;w) = log\prod_{n=1}^{N}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y_n-w^Tx_n)^2}{2\sigma^2}} = \sum_{n=1}^{N} log\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(y_n-w^Tx_n)^2}{2\sigma^2}}$$

$$\Leftrightarrow \min_w \sum_{n=1}^{N}\left((y_n - w^Tx_n)^2\right)$$ 最小二乘

存在的问题：当$\mathbf{X} \in \Re^{N \times D}$非列满秩时（且N<D），$\mathbf{X}^{\mathrm{T}}\mathbf{X}$是奇异矩阵，那么计算$(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$会产生较大误差，变成了一个不适定问题(病态问题，欠定)，该模型缺乏稳定性（对噪声的敏感性很强）！

奇异矩阵A：|A|=0; 矩阵A是非满秩矩阵；
非奇异矩阵A: |A|≠0; 矩阵A是满秩矩阵。

欠定问题、超定问题：

对于数据矩阵$\boldsymbol{X} \in \mathfrak{R}^{N\times d}$, d为属性维度，N为样本点数。

- 所谓欠定问题，即不适定问题，是由于样本点数目少于变量的数目，即X不是列满秩的，解不唯一（需要增加限定条件，即先验知识）；
- 超定问题，是样本点数目大于变量的数目，且X为列满秩。对于超定方程问题，解不存在，但存在近似解。最小二乘是用于解决超定问题，获得最佳近似解。

模型的性能评价

已经介绍了普通最小二乘法的原理，对于由N个样本，计算出的最小二乘模型参数w，如何说明该模型是好的呢？

- 一般地，在机器学习中，为了验证模型的好坏，还需要一组新的样本（测试样本集）进行测试，根据测试准确率，判定模型的好坏。注：测试样本集是独立于N个样本的（也称训练样本）。
- 交叉验证（Leave-one-out cross validation, LOO）：K折交叉验证，也是目前机器学习和人工智能领域具有代表性的验证模型泛化能力的训练方法。

模型的性能评价

K折交叉验证（Leave-one-out cross validation, LOO）（留一法）：

将数据集分成相同数量的K份（K折），每份数据分别轮流作为测试集，其余的K-1份作为训练集。执行K次测试的平均精度，作为最后的模型精度。

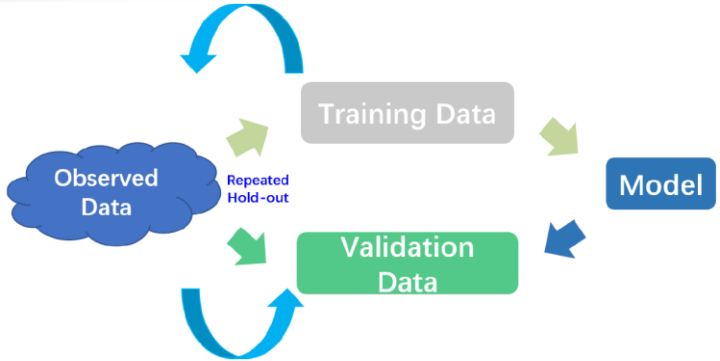
特别注意的是：当K=N时，N折交叉验证，变成了“留一法”，即每个样本分别轮流作为测试样本（测试集中只有一个样本），剩余的N-1个样本作为训练集。

为什么要提出模型性能的评价方式？？

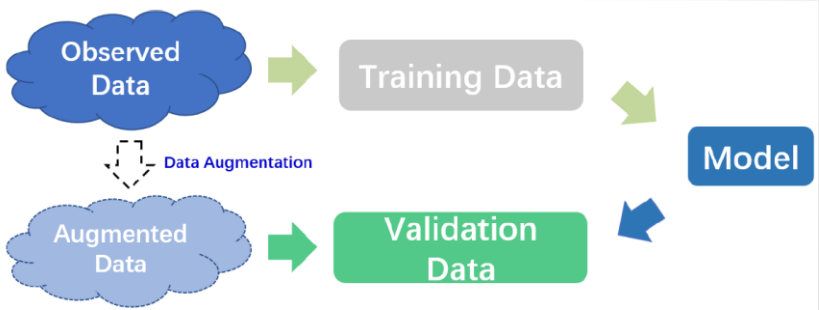
模型的性能评价

某些领域（医学）数据十分昂贵，如果仅有的少量数据还要拿出一些数据作为验证集，实属浪费。

From leave-one-out to leave-zero-out (data augmentation)



leave-one-out



leave-zero-out

广义线性回归

给定一组样本：$\langle \mathbf{x}_i, y_i \rangle_{i=1...m}$ ，我们要拟合下面的假设

$$h_{\mathbf{w}}(\mathbf{x}) = \sum_{k=0}^{K-1} w_k \phi_k(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

其中，ϕ_k 为基函数，可以是多阶的函数，比如平方、立方等。

为了求最佳的预测系数w, 我们需要最小化下列损失函数

$$\sum_{i=1}^{m} (y_i - h_{\mathbf{w}}(\mathbf{x}_i))^2$$

那么最优解w可以写为

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

广义线性回归

- 由前面的分析 可知，$h_w(x)$对于参数w来说，是线性模型。但并不意味着， $h_w(x)$是关于输入x的线性函数，比如$h_w(x) = wx^2$。(多项式回归)
- 因此，广义的线性模型可以写成

$$h_{\mathrm{w}}(\mathbf{x}) = \sum_{k=0}^{K-1} w_k \phi_k(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x})$$

其中，ϕ_k 为基函数。

- 线性假设模型还可以重新写成

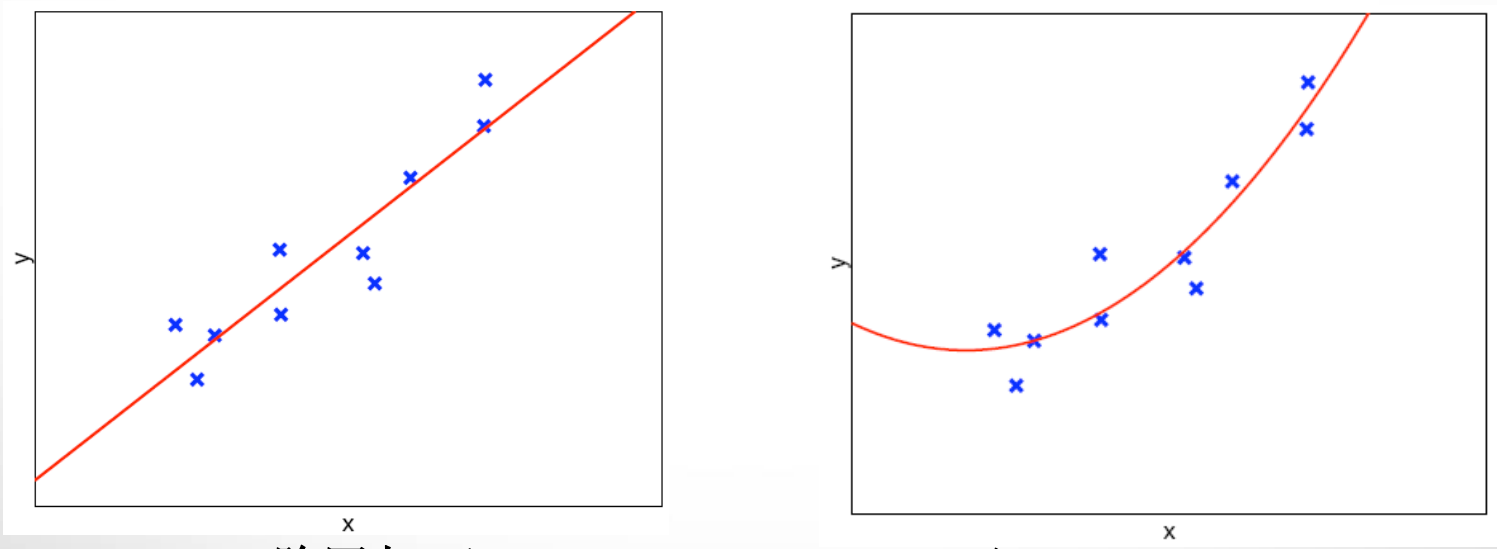
$$h_{\mathrm{w}}(\mathbf{x}) = \mathbf{\Phi w}$$

关于w的线性模型

其中，$\mathbf{\Phi}$是由 $\boldsymbol{\phi}(\mathbf{x}_j)$构成的矩阵。

广义线性回归

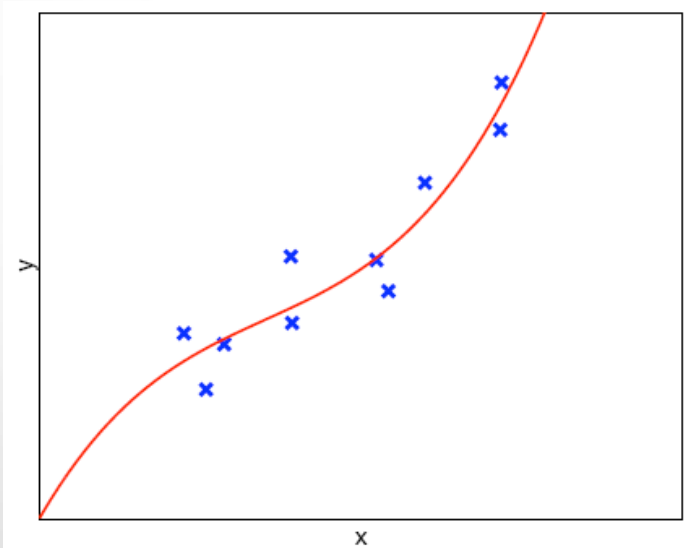
采用不同基函数（不同阶次）的回归模型效果

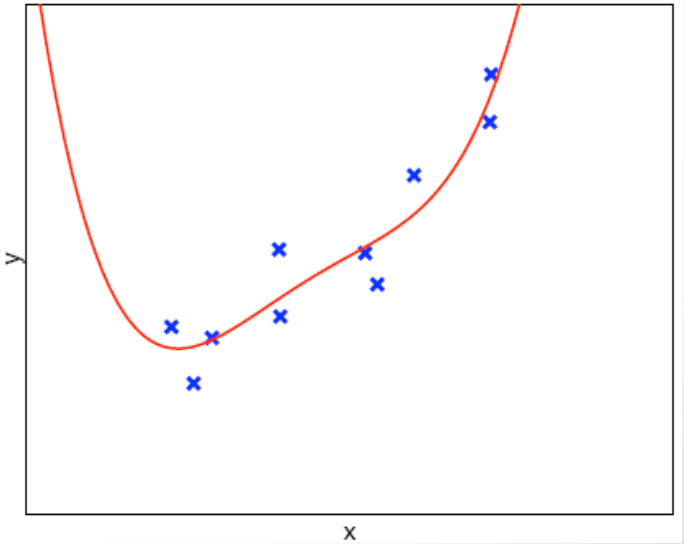


1阶回归 （is it better to fit the data?） 2阶回归

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

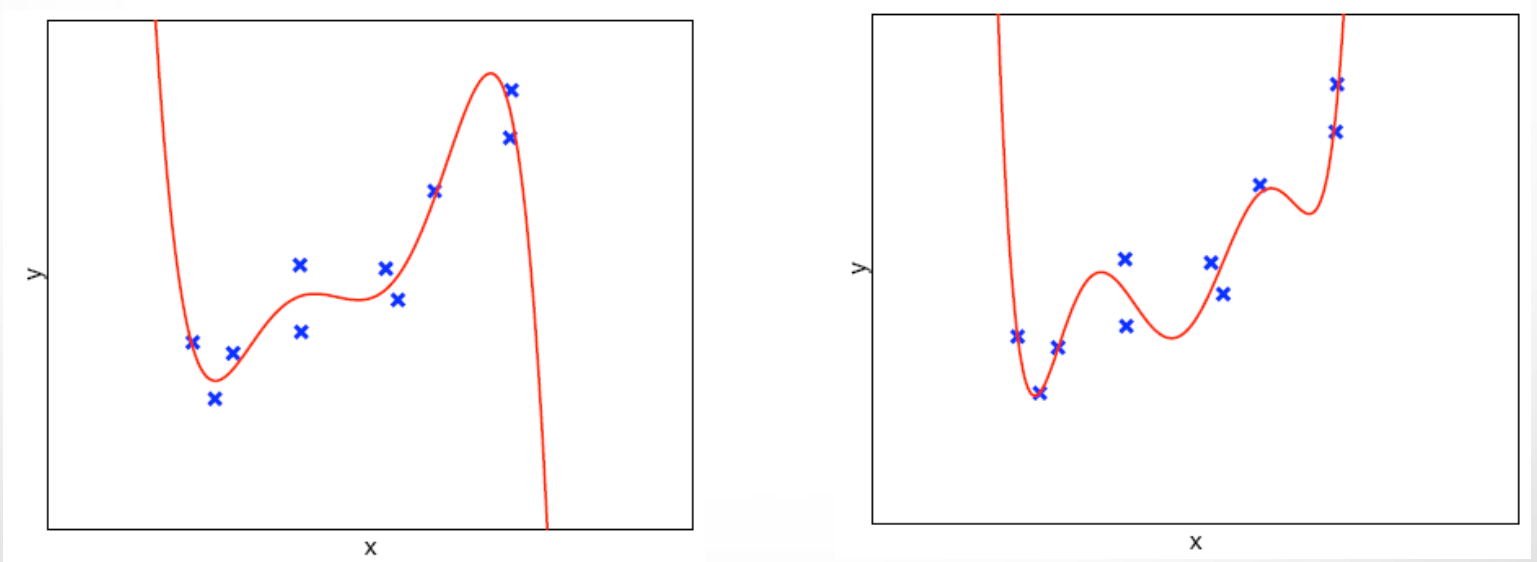




3阶回归 (is it better to fit the data?)　　4阶回归

广义线性回归

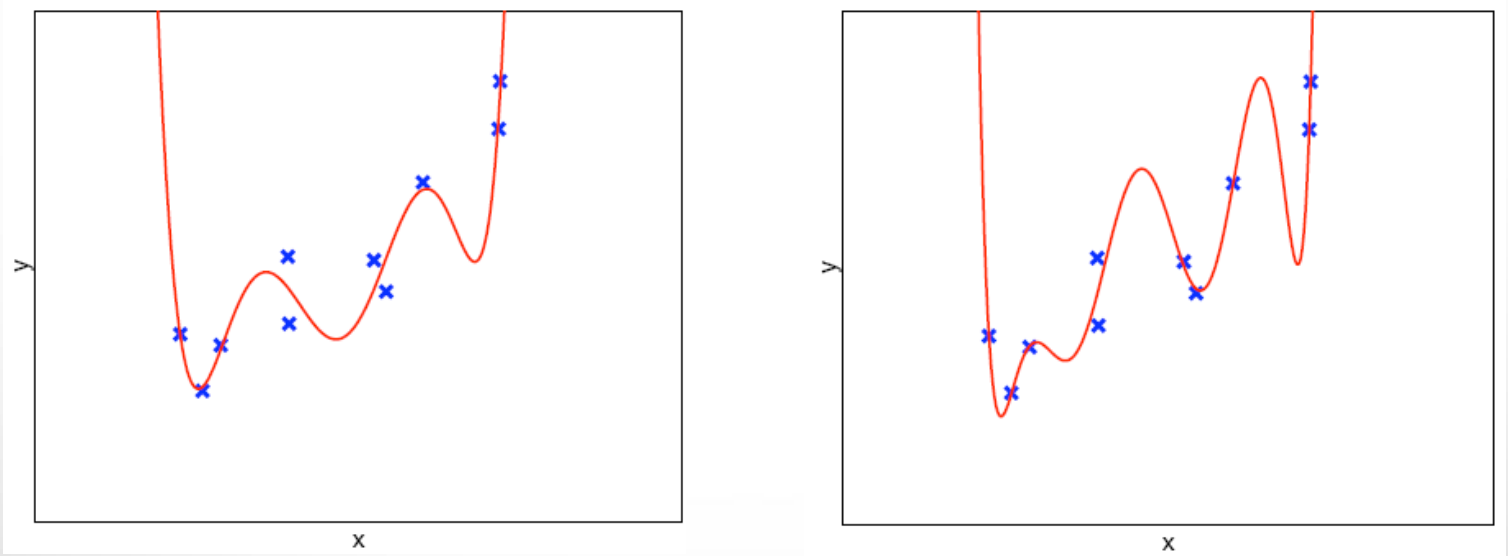
采用不同基函数（不同阶次）的回归模型效果



5阶回归 (is it better to fit the data?)　　6阶回归

广义线性回归

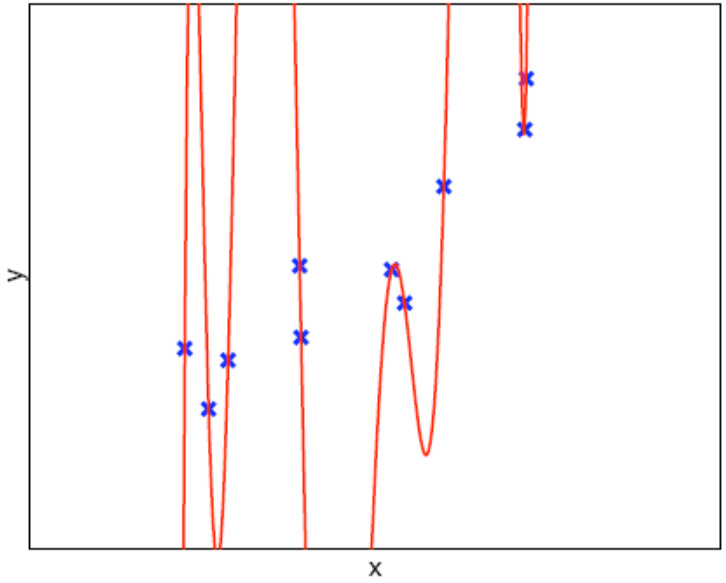
采用不同基函数（不同阶次）的回归模型效果



7阶回归 (is it better to fit the data?)　　8阶回归

广义线性回归

采用不同基函数（不同阶次）的回归模型效果

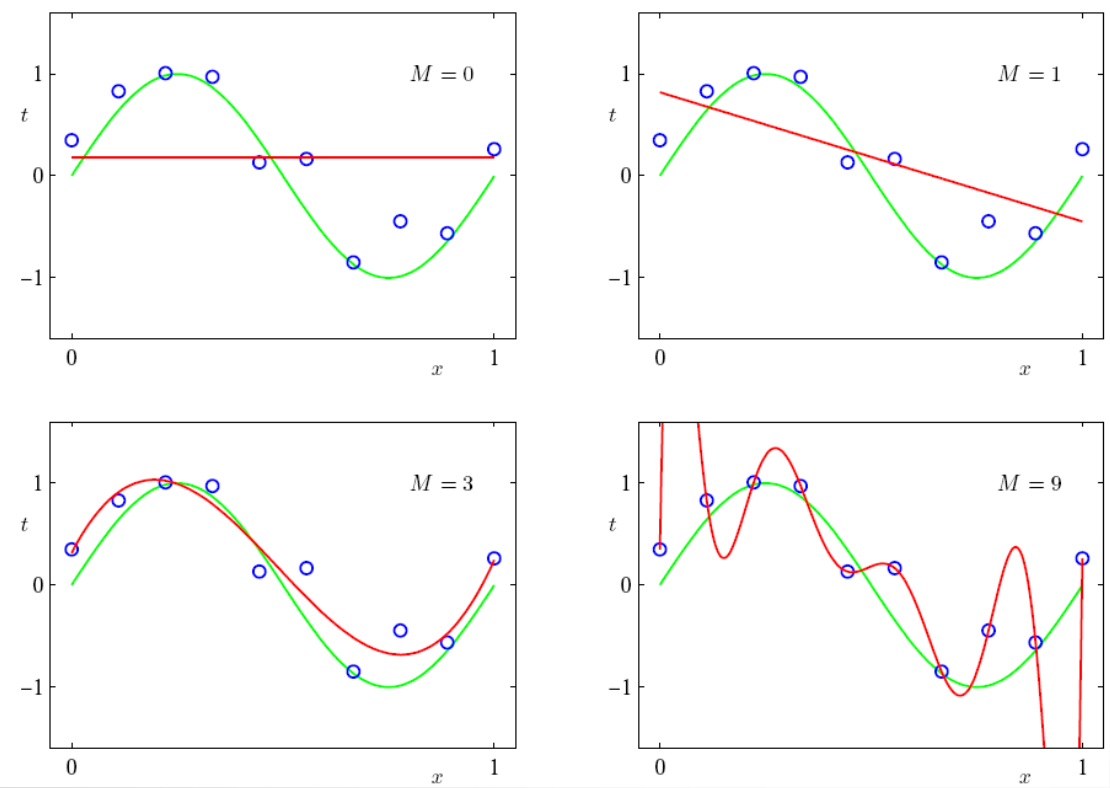


9阶回归 (is it better to fit the data?)

过拟合（overfitting）

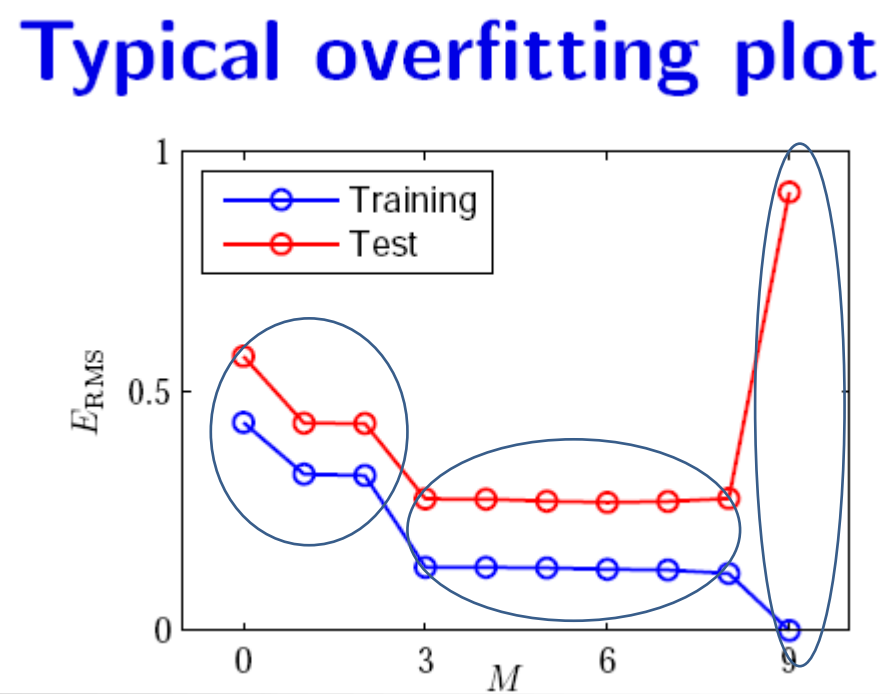
- 一般来说，过拟合是所有机器学习算法中的一个非常重要的问题；
- 通常，我们能够找到一种完美的假设，准确无误的预测训练数据。但是，对于新数据却不能够很好的预测（泛化能力很差）。
- 通过刚才的例子看到，如果参数越来越多，模型倾向于“记忆”训练数据点，而不是“推理”某种规律。

过拟合（overfitting）与欠拟合（underfitting）



- M为模型的阶数，反映模型的自由度和复杂度。
- **过拟合**是指，训练样本的误差非常低，而新测试样本的预测误差非常大。
- **欠拟合**是指，训练样本的预测误差非常高（学习不够）。
- 通常，**过拟合**是更为常见的问题，因此，在建模过程中，我们需要经验和理论分析，来避免这个问题。

过拟合（overfitting）与欠拟合（underfitting）



- 从图中可以看出，M=0,1,2时，属于**欠拟合**（训练误差较大）
- M=3,4,5,6,7,8时，属于**拟合情况良好**；但根据模型的几个准则（参数越少、模型越简单越好），显然M=3是最佳参数。
- M=9时，属于**过拟合**。（训练误差非常小，接近0，但测试误差非常大）

在设计模型寻求某种假设时，如何防止过拟合（overfitting）与欠拟合（underfitting）？

非常有效的交叉验证策略：

- 将整个数据集分成3个独立的部分：训练集、验证集、测试集

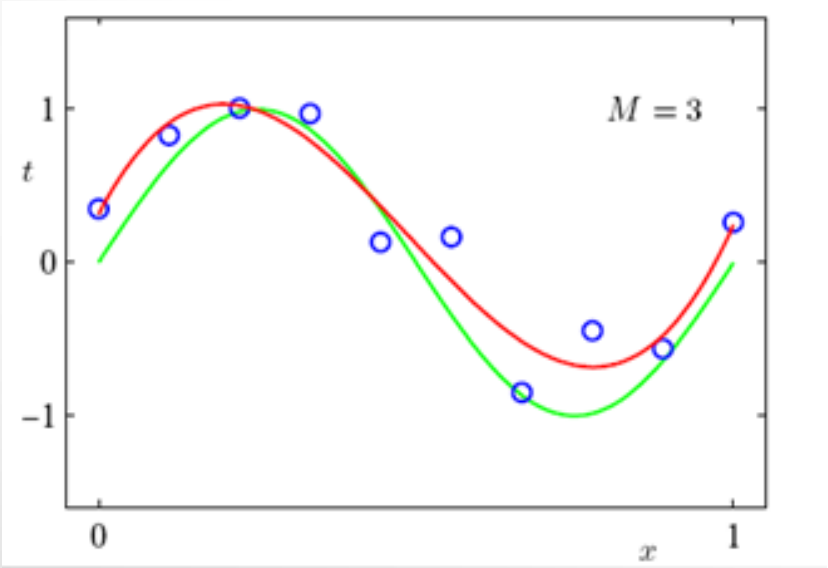
✓ 训练集是用于训练某种假设；

✓ 验证集用于检验假设的可靠性，并进一步修正模型（即M）；

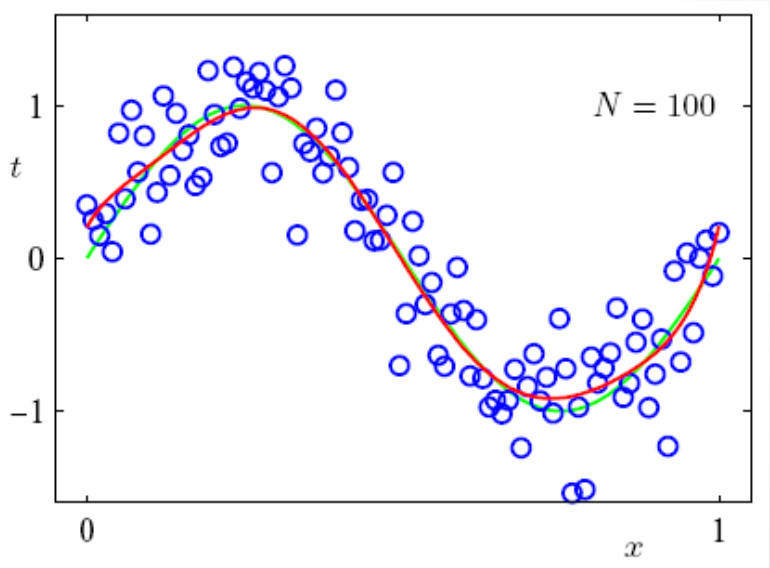
✓ 测试集用于检验最终模型假设的性能。

一般地，采用随机分割的形式，执行以上程序N次，每次测试集的准确率的平均值为最终的模型的性能评价指标。

最佳的假设h(x)依赖于训练样本数量和模型复杂度



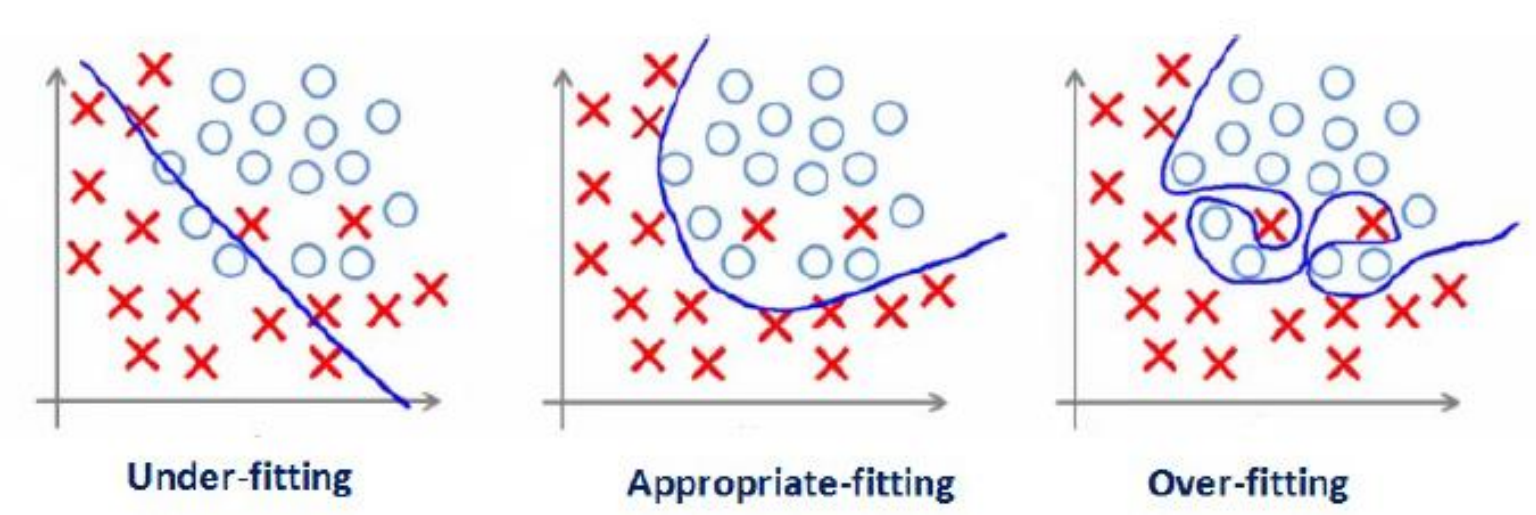
10个训练样本的回归曲线 （M=3）
红色表示假设曲线，绿色为真实值



100个训练样本的回归曲线 (M=3)
假设与真实更加接近

最佳的假设h(x)依赖于训练样本数量和模型复杂度

分类问题：



对假设的精度进行评估是机器学习的基本问题。

然而，由于有限的数据样本一般不代表数据的一般分布，所以利用有限样本得出的假设精度可能存在误差。

我们将介绍一种统计的方法，结合有关数据基准分布的假定，我们可以用有限数据样本的观察精度来逼近整个数据分布的真实精度。

评估假设

一、样本错误率

样本错误率：该假设在可见数据S上的错误率；

定义：假设h关于目标值和可见数据S的样本错误率为

$$error_S(h) = \frac{1}{n}\sum_{x \in S} \delta\big(f(x), h(x)\big)$$

其中，n为样本集S的数量。如果$f(x) \neq h(x)$，$\delta\big(f(x), h(x)\big) = 1$，否则$\delta\big(f(x), h(x)\big) = 0$

可类比抛掷硬币问题中计算硬币出现正面的概率：n个样例中假设出错的次数，等价于抛掷n次硬币出现正面的次数（可见样本的错误率）

评估假设

二、真实错误率

真实错误率：该假设在分布为D的整个实例集合上的错误率；

定义：假设h关于目标值和分布为D的真实错误率，为h按D分布随机抽取的实例被误分类的概率

$$error_D(h) = Pr_{x \in D}[f(x) \neq h(x)]$$

问题：$error_S(h)$在何种程度上提供了对$error_D(h)$的估计？

该真实错误率估计问题等价于估计抛出一枚硬币出现正面的概率p（即，随机抽取的实例被错误预测的概率$error_D(h)$）；

评估假设

三、真实错误率的估计

问题：$error_S(h)$在何种程度上提供了对$error_D(h)$的估计？

考虑假设h为离散的情况，利用假设h在S上观察到的样本错误率估计真实错误率。举例：

- 样本集S包含n个样本，其抽取方式按照概率分布D抽样，且相互独立；
- 样本数$n \geq 30$
- 假设h在这n个样例上犯错次数为r。显然，$error_S(h) = r/n$

统计发现，当重复上述实验多次(每次抽取n个实例)，随机变量$error_S(h)$呈现出二项分布；

二项分布刻画的是抛掷硬币n次出现r次正面的概率，即n个随机样例中出现r次预测错误的概率。

二项分布：

一个二项分布刻画了对于一个出现正面概率为p的硬币，独立投掷n次中观察到r次正面的概率。

给定一个随机变量X服从二项分布，X=r的概率表达式为

$$P(X=r)=\frac{n!}{r!\,(n-r)!}p^r(1-p)^{n-r}$$

X的期望E(X)=np

X的方差Var(X)=np(1-p)

n次伯努利实验：指“是/否”的决策

期望的定义：考虑随机变量X的可能取值为x1,…,xn, 则X的期望值E[X]为

$$E[X]=\sum_{i=1}^{n}x_iP(X=x_i)$$

方差的定义：

$$Var[X]=E[(X-E[X])^2]$$

评估假设

(1)考虑假设h的样本错误率$error_S(h)$，即n个随机样例组成的集合S中，出现r次犯错的比率：

$$error_S(h) = \frac{r}{n}$$

其中，随机变量$r \sim B(n,p)$，均值$E[r] = np$，方差$Var(r) = np(1-p)$

(2)$error_S(h)$是一个二项分布，其均值和标准差为

$$E[error_S(h)] = \frac{E[r]}{n} = p$$

$$\sigma[error_S(h)] = \frac{\sqrt{Var(r)}}{n} = \sqrt{\frac{p(1-p)}{n}}$$

可以发现：样本错误率的期望正是真实错误率p (即无偏估计，E(Y)-p=0)。**p的近似值为r/n**

评估假设

三、真实错误率的估计

问题：$error_S(h)$在何种程度上提供了对$error_D(h)$的估计？

真实错误率以概率P的可能性落在下面的区间内

$$error_D(h) = error_S(h) \pm z_P \sqrt{\frac{error_S(h)\left(1 - error_S(h)\right)}{n}}$$

其中z_p为双侧的P置信区间的常量。对于正态分布，当P=95%时，$z_P = 1.96$；当P=90%时，$z_P = 1.64$。

统计意义：当从D中随机抽取多个样本测试时，会发现95%的实验计算所得区间包含真实错误率。

置信区间

通常描述某个估计的不确定性的方法是使用置信区间，真实的值（真实错误率）以一定的概率落入该区间中。因此，估计真实错误率的问题转化为估计置信区间的问题。

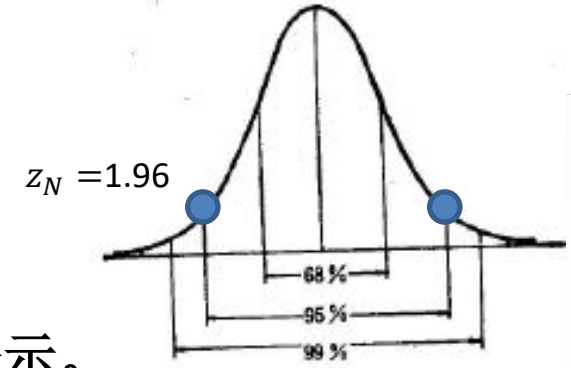
定义：某个参数q的N%置信区间是一个以N%的概率包含q的区间。

如果要估计$error_D(h)$的置信区间，只需要找到以$error_D(h)$为中心的区间，区间的宽度需要保证$error_S(h)$能有95%的机会落入其中。

对于二项分布来说，计算置信区间比较繁琐。当样本数n足够大时（研究表明np(1-p)>5时），二项分布可以近似为正态分布。而正态分布的置信区间有统计列表做指导。

置信区间

正态分布的置信区间（双侧）：



置信区间的半宽度利用常数z_N和标准差表示。

z_N: 包含N%概率质量的关于均值的最小区间的宽度。

总之，如果随机变量Y服从$\mathcal{N}(\mu, \sigma^2)$，那么Y的任意一个观察值y有N%的机会（概率）落入下面的区间：

$$\mu \pm z_N \sigma$$

双侧情况

置信度N%	50%	68%	80%	90%	95%	98%	99%
常量z_N	0.67	1.00	1.28	1.64	1.96	2.33	2.58

评估假设

三、真实错误率的估计

问题：$error_S(h)$在何种程度上提供了对$error_D(h)$的估计？

真实错误率以概率p的可能性落在下面的区间内

$$error_D(h) = \mu \pm z_N \sigma$$

利用近似$p=r/n$ $=error_S(h)$

$$= p \pm z_N \sqrt{\frac{p(1-p)}{n}}$$

$$error_D(h) = error_S(h) \pm z_p \sqrt{\frac{error_S(h)\left(1-error_S(h)\right)}{n}}$$

其中，z_p为双侧的p置信区间的常量。当p=95%时，$z_p = 1.96$；当p=90%时，$z_p = 1.64$。

推导中的两个近似：

1. $error_D(h)$近似为$error_S(h)$，当n≥30时，效果较好
2. 二项分布近似为正态分布，当np(1-p)≥5时，效果较好

评估假设

三、真实错误率的估计

问题：$error_S(h)$在何种程度上提供了对$error_D(h)$的估计？

举例：当数据集S包含40个样例时，产生了r=12个错误，对于置信水平p=0.95时，则有：

$$error_S(h) = \frac{12}{40} = 0.3$$

$$error_D(h) = error_S(h) \pm z_p \sqrt{\frac{error_S(h)\left(1 - error_S(h)\right)}{n}}$$

$$= 0.3 \pm 1.96 \sqrt{\frac{0.3(1-0.3)}{40}}$$

$$= 0.3 \pm 0.14$$

评估假设

四、算法假设评估与检验

问题：对于两个算法A和B，其获得的假设分别为h1和h2，使用大小为100的独立样本集S，且知道$error_S(h1) = 0.3, error_S(h2) = 0.2$。那么，“$error_D(h1) > error_D(h2)$的可能性有多大？”

分析：上述问题可转化为

$d = error_D(h1) - error_D(h2) > 0$的概率有多大？

评估假设

四、算法假设评估与检验

考虑估计量：

$$\hat{d} = error_S(h1) - error_S(h2) = 0.3 - 0.2 = 0.1$$

显然，$\hat{d}$是d的无偏估计量，即$E[\hat{d}] = d$。

根据前面所讲内容，$error_S(h1)$和$error_S(h2)$近似为正态分布，那么$\hat{d}$也服从正态分布。有：

$$\sigma_{\hat{d}} = \sqrt{\frac{error_S(h1)(1 - error_S(h1))}{n} + \frac{error_S(h2)(1 - error_S(h2))}{n}}$$

评估假设

四、算法假设评估与检验

那么，d的N%双侧置信区间估计为：

$$\hat{d} \pm z_N \sigma_{\hat{d}}$$

回到刚才的问题：“$error_D(h1) > error_D(h2)$的可能性有多大？”

即，“$d = error_D(h1) - error_D(h2) > 0$的概率有多大？”

问题转化：已知$\hat{d} = 0.1$, 实际上就是考查“$\hat{d} < d + 0.1$的概率有多大？”

按照标准的表达式，$\hat{d}$落入下列区间的概率（单侧区间）：

$$\hat{d} < \mu_d + z_N \sigma_{\hat{d}}$$

已知$\sigma_{\hat{d}} = 0.061$，那么$z_N \approx 1.64$，查表可知，对应置信度为90%的双侧区间。而对于单侧区间，则有95%的置信度。

评估假设

四、算法假设评估与检验

结论：对于观察到的$\hat{d} = 0.1$，$error_D(h1) > error_D(h2)$的可能性有95%。（具有统计显著意义）

统计学术语：接受(accept)“$error_D(h1) > error_D(h2)$”这一假设的置信度为0.95；

或者：以1-0.95=0.05的显著水平，拒绝(reject)对立假设。