机器学习
Machine Learning

第三讲：回归的基本学习算法

1. 线性回归模型

特征向量（输入向量）

$$\boldsymbol{x}=\left[x_1,x_2,\cdots,x_K\right]^T$$

扩充特征向量

$$\bar{\boldsymbol{x}}=\left[1,x_1,x_2,\cdots,x_K\right]^T \qquad \text{即}\ x_0=1$$

权系数向量

$$\boldsymbol{w}=\left[w_0,w_1,w_2,\cdots,w_K\right]^T$$

则线性回归模型为

$$\hat{y}(\boldsymbol{x},\boldsymbol{w})=\sum_{k=0}^{K} w_k x_k=\boldsymbol{w}^T\bar{\boldsymbol{x}}$$

线性回归模型（续）

给出训练序列

$$\mathbf{D}=\left\{\left(\boldsymbol{x}_1,y_1\right),\left(\boldsymbol{x}_2,y_2\right),\cdots,\left(\boldsymbol{x}_N,y_N\right)\right\}$$

$$=\left\{\left(\boldsymbol{x}_n,y_n\right)\right\}_{n=1}^{N}$$

对于给出的损失函数最小化，得到回归参数 $\boldsymbol{w}$

对于给出的新特征向量 $\boldsymbol{x}$， 得到预测值

$$\hat{y}(\boldsymbol{x},\boldsymbol{w})=\sum_{k=0}^{K}w_k x_k=\boldsymbol{w}^T\overline{\boldsymbol{x}}$$

2. 扩充：线性基函数回归模型

特点：对参数向量线性，对特征向量非线性

对特征向量映射基函数：

$$\boldsymbol{x}=\left[x_1,x_2,\cdots,x_K\right]^T \quad \Rightarrow$$

$$\boldsymbol{\phi}\left(\boldsymbol{x}\right)=\left[\phi_0(\boldsymbol{x}),\phi_1(\boldsymbol{x}),\phi_2(\boldsymbol{x}),\cdots,\phi_{M-1}(\boldsymbol{x})\right]^T$$

线性基函数模型：

$$\hat{y}(\boldsymbol{x},\boldsymbol{w})=\sum_{j=0}^{M-1}w_j\phi_j(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{\phi}\left(\boldsymbol{x}\right)$$

特例： <u>**线性回归是线性基函数回归的特例**</u>： $\boldsymbol{\phi}\left(\boldsymbol{x}\right)=\overline{\boldsymbol{x}}$

基函数集的例子-1

$$\boldsymbol{x} = \left[x_1, x_2, x_3\right]^T \quad \Rightarrow$$

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left[1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1 x_2, x_1 x_3, x_2 x_3\right]^T$$

对应线性回归模型

$$\hat{y}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

对应线性基函数回归模型

$$\hat{y}(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + + w_4 x_1^2 +$$

$$+ w_5 x_2^2 + w_6 x_3^2 + w_7 x_1 x_2 + w_8 x_1 x_3 + w_9 x_2 x_3$$

3. 基本线性回归模型的学习

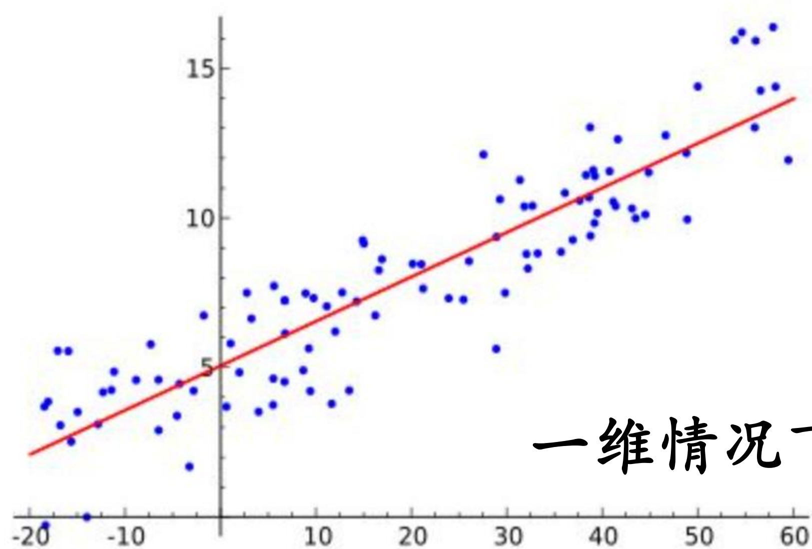
独立同分布条件（I.I.D）的训练数据集

$$\boldsymbol{D}=\left\{\left(\boldsymbol{x}_1,y_1\right),\left(\boldsymbol{x}_2,y_2\right),\cdots,\left(\boldsymbol{x}_N,y_N\right)\right\}=\left\{\left(\boldsymbol{x}_n,y_n\right)\right\}_{n=1}^{N}$$

对每个样本，模型与标注之间存在误差 ε_i

$$y_i=\hat{y}(\boldsymbol{x}_i,\boldsymbol{w})+\varepsilon_i=\boldsymbol{w}^{\mathrm{T}}\overline{\boldsymbol{x}}_i+\varepsilon_i$$

$$\varepsilon_i\sim N(0,\sigma_{\varepsilon}^2)$$



一维情况下，线性回归表示的例子

通过最大似然原理导出线性回归的学习算法

y_i 的概率密度函数

$$p_y(y_i|\boldsymbol{w}) = \frac{1}{(2\pi\sigma_\varepsilon^2)^{\frac{1}{2}}}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}\left(y_i - \hat{y}(\boldsymbol{x}_i,\boldsymbol{w})\right)^2\right]$$

所有样本的标注值表示为向量

$$\boldsymbol{y} = \left[y_1, y_2, y_3, \cdots, y_N\right]^{\mathrm{T}}$$

由样本集的I.I.D性得似然函数

$$p_y(\boldsymbol{y}|\boldsymbol{w}) = \prod_{i=1}^{N} p_y(y_i|\boldsymbol{w})$$

$$= \frac{1}{(2\pi\sigma_\varepsilon^2)^{\frac{N}{2}}}\exp\left[-\frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{N}\left(y_i - \hat{y}(\boldsymbol{x}_i,\boldsymbol{w})\right)^2\right]$$

ML导出线性回归的学习算法（续）

取对数似然为

$$\log p_y(\mathbf{y}|\boldsymbol{w}) = -\frac{N}{2}\log(2\pi\sigma_\varepsilon^2) - \frac{1}{2\sigma_\varepsilon^2}\sum_{i=1}^{N}\left(y_i - \hat{y}(\boldsymbol{x}_i, \boldsymbol{w})\right)^2$$

对数似然函数最大，对应平方误差和最小

$$J\left(\boldsymbol{w}\right) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \hat{y}(\boldsymbol{x}_i, \boldsymbol{w})\right)^2$$

重写误差平方和

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\varepsilon_i^2 = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \boldsymbol{w}^{\mathrm{T}}\overline{\boldsymbol{x}}_i\right)^2$$

$$= \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})$$

$$= \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\right\|_2^2$$

其中

$$\boldsymbol{X} = \begin{bmatrix} \overline{\boldsymbol{x}}_1^{\mathrm{T}} \\ \overline{\boldsymbol{x}}_2^{\mathrm{T}} \\ \vdots \\ \overline{\boldsymbol{x}}_N^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1K} \\ 1 & x_{21} & \cdots & x_{2K} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & x_{N1} & \cdots & x_{NK} \end{bmatrix}$$

ML导出线性回归的学习算法（续）

求解系数向量

$$\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \frac{\partial \frac{1}{2}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w})}{\partial \boldsymbol{w}} = -\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w} = \mathbf{0}$$

系数向量满足方程

$$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}_{ML} = \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

若$\boldsymbol{X}^{T}\boldsymbol{X}$可逆，解为 （线性回归的最小二乘（LS）解）

$$\boldsymbol{w}_{ML} = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

权系数向量得到后，线性回归函数确定为

$$\hat{y}(\boldsymbol{x},\boldsymbol{w}) = \boldsymbol{w}_{ML}^{T}\overline{\boldsymbol{x}}$$

线性回归的几何解释

线性回归在训练集上的输出向量

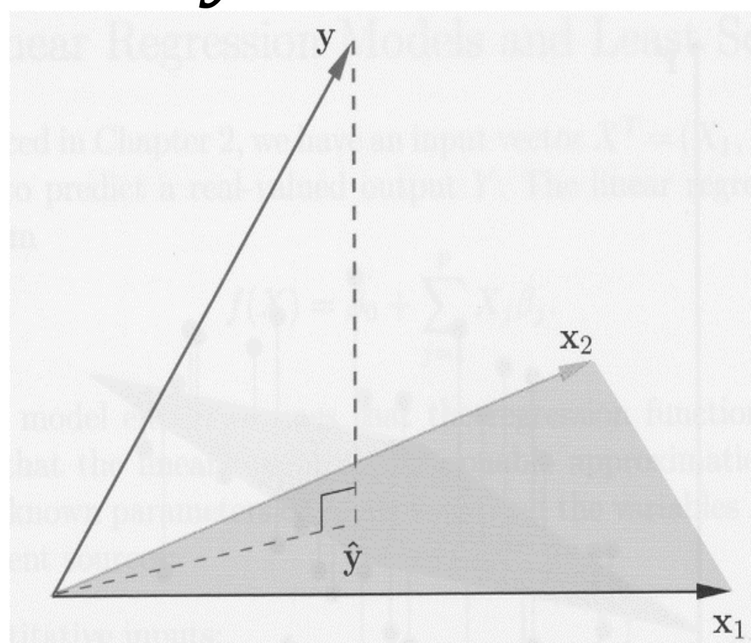
$$\hat{\boldsymbol{y}} = \boldsymbol{X}\boldsymbol{w}_{ML} = \begin{bmatrix} \tilde{\boldsymbol{x}}_0 & \tilde{\boldsymbol{x}}_1 & \cdots & \tilde{\boldsymbol{x}}_K \end{bmatrix} \boldsymbol{w}_{ML} = \sum_{i=0}^{K} w_{ML,i} \tilde{\boldsymbol{x}}_i$$

线性回归对每一个标注值的逼近误差向量

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{P}\boldsymbol{y} = \boldsymbol{P}^{\perp}\boldsymbol{y}$$

回归误差正交性

$$\boldsymbol{\varepsilon}^{\mathrm{T}} \hat{\boldsymbol{y}} = 0$$



4. 线性回归模型的递推学习

对数据集计算平均梯度

$$\frac{1}{N}\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}} = -\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_i\right)\bar{\boldsymbol{x}}_i$$

从初始猜测值 $\boldsymbol{w}^{(0)}$ 按梯度下降算法更新

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \eta_k \frac{1}{N}\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}$$

$$= \boldsymbol{w}^{(k)} + \frac{\eta_k}{N}\sum_{i=1}^{N}\left(y_i - \boldsymbol{w}^{(k)\mathrm{T}}\bar{\boldsymbol{x}}_i\right)\bar{\boldsymbol{x}}_i$$

随机梯度下降算法

（stochastic gradient descent，SGD）

误差和可分解为

$$J(\boldsymbol{w})=\frac{1}{2}\sum_{i=1}^{N}\left(y_i-\boldsymbol{w}^{\mathrm{T}}\overline{\boldsymbol{x}}_i\right)^2=\sum_{i=1}^{N}J_i(\boldsymbol{w})$$

一个样本的梯度

$$\left.\frac{J_i(\boldsymbol{w})}{\partial\boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}=-\left(y_i-\boldsymbol{w}^{(k)\mathrm{T}}\overline{\boldsymbol{x}}_i\right)\overline{\boldsymbol{x}}_i$$

利用一个样本梯度对权系数向量的更新

$$\boldsymbol{w}^{(k+1)}=\boldsymbol{w}^{(k)}-\eta\left.\frac{\partial J_i(\boldsymbol{w})}{\partial\boldsymbol{w}}\right|_{\boldsymbol{w}=\boldsymbol{w}^{(k)}}$$

$$=\boldsymbol{w}^{(k)}+\eta_k\left(y_i-\boldsymbol{w}^{(k)\mathrm{T}}\overline{\boldsymbol{x}}_i\right)\overline{\boldsymbol{x}}_i$$

梯度是随机的，
每次迭代样本
是随机选取的

小批量SGD算法

从数据集随机抽取一小批量样本

$$\boldsymbol{D}_{k+1} = \left\{ \left(\boldsymbol{x}_m, y_m \right) \right\}_{m=1}^{N_1}$$

小批量SGD算法如下

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} + \eta \frac{1}{N_1} \sum_{m=1}^{N_1} \left(y_m - \boldsymbol{w}^{(k)\mathrm{T}} \overline{\boldsymbol{x}}_m \right) \overline{\boldsymbol{x}}_m$$

一般收敛性条件

$$\sum_{k=1}^{\infty} \eta_k = \infty \qquad \sum_{k=1}^{\infty} \eta_k^2 < \infty$$

5. 正则化线性回归

<u>正则化目标函数与贝叶斯框架的等价性！</u>

正则化：在目标函数中增加约束参数向量的量

一种常用选择为参数向量的范数平方约束

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \boldsymbol{w}^{\mathrm{T}}\overline{\boldsymbol{x}}_i\right)^2 + \frac{\lambda}{2}\sum_{i=1}^{K} w_i^2$$

$$= \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}$$

参数向量的正则化LS解为

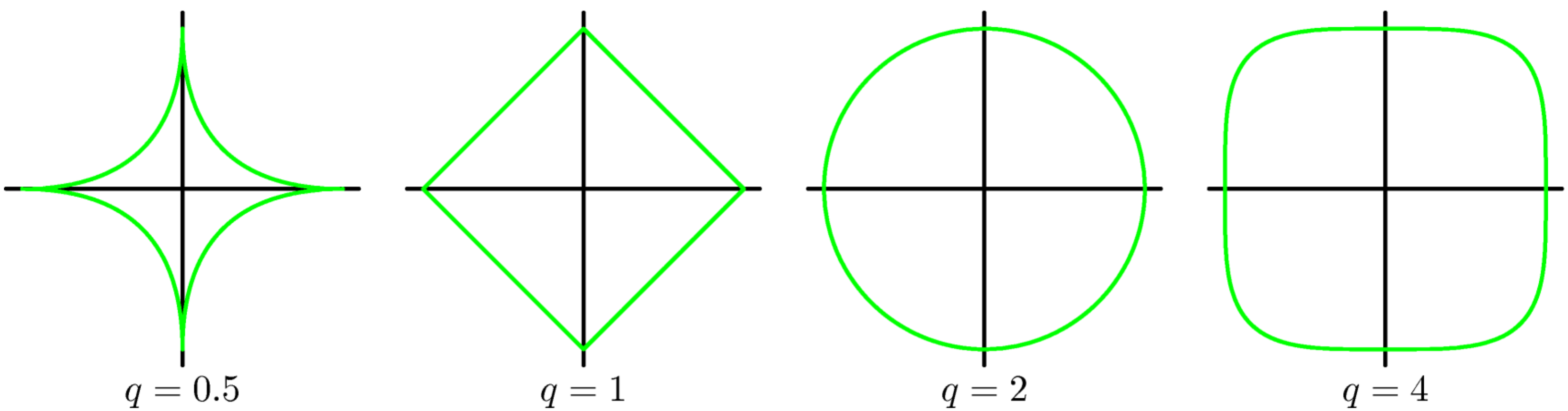
$$\boldsymbol{w}_R = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$$

更一般的正则化

注：贝叶斯框架下，一个q取值对应参数向量的一种不同的先验概率假设

对更一般的q值，定义正则化目标函数

$$J(\boldsymbol{w})=\frac{1}{2}\sum_{i=1}^{N}\left(y_i-\boldsymbol{w}^{\mathrm{T}}\bar{\boldsymbol{x}}_i\right)^2+\frac{\lambda}{2}\sum_{i=1}^{K}\left|w_i\right|^q$$

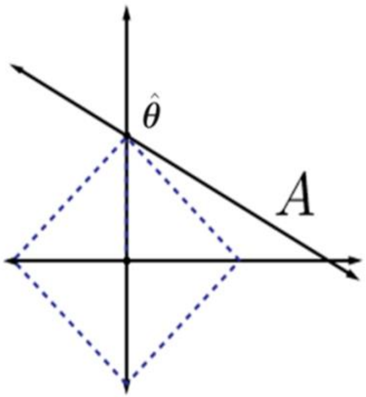


Lasso

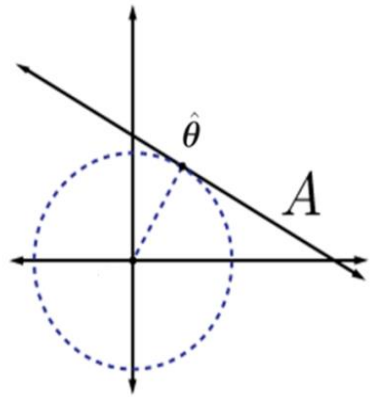
Quadratic

更一般的正则化

q=1，Lasso解趋向于稀疏化

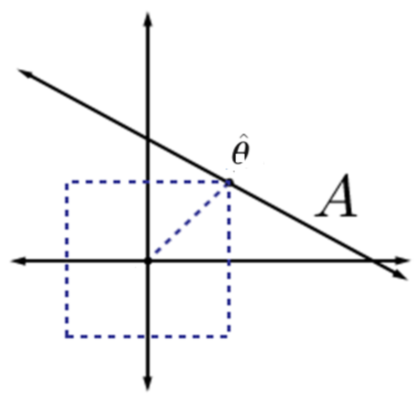


$p = 1$

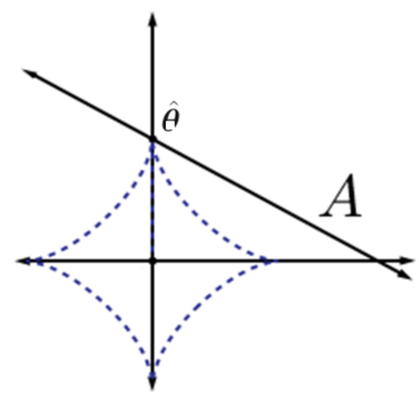


$p = 2$

q<1:稀疏化
q>1:非稀疏化



$p = \infty$



$p = \frac{1}{2}$

6. 线性基函数回归

输入向量映射： $\boldsymbol{x} \mapsto \boldsymbol{\phi}(\boldsymbol{x})$

$$\boldsymbol{\phi}(\boldsymbol{x}) = \left[\phi_0(\boldsymbol{x}), \phi_1(\boldsymbol{x}), \cdots, \phi_M(\boldsymbol{x})\right]^{\mathrm{T}}$$

线性基函数回归模型

$$\hat{y}(\boldsymbol{\phi}, \boldsymbol{w}) = \sum_{k=0}^{M} w_k \phi_k(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x})$$

线性基函数回归系数向量的解为

$$\boldsymbol{w}_{ML} = (\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{y}$$

基函数数据矩阵为

$$\boldsymbol{\Phi}=\begin{bmatrix}\boldsymbol{\phi}^{\mathrm{T}}\left(\boldsymbol{x}_1\right)\\ \boldsymbol{\phi}^{\mathrm{T}}\left(\boldsymbol{x}_2\right)\\ \vdots\\ \boldsymbol{\phi}^{\mathrm{T}}\left(\boldsymbol{x}_N\right)\end{bmatrix}$$

$$=\begin{bmatrix}\phi_0\left(\boldsymbol{x}_1\right) & \phi_1\left(\boldsymbol{x}_1\right) & \cdots & \phi_M\left(\boldsymbol{x}_1\right)\\ \phi_0\left(\boldsymbol{x}_2\right) & \phi_1\left(\boldsymbol{x}_2\right) & \cdots & \phi_M\left(\boldsymbol{x}_2\right)\\ \vdots & \vdots & \cdots & \vdots\\ \phi_0\left(\boldsymbol{x}_N\right) & \phi_1\left(\boldsymbol{x}_N\right) & \cdots & \phi_M\left(\boldsymbol{x}_N\right)\end{bmatrix}$$

三维向量例子 $\boldsymbol{x}_n=\left[x_{n,1},x_{n,2},x_{n,3}\right]^{\mathrm{T}}$

$$\boldsymbol{\phi}\left(\boldsymbol{x}_n\right)=\left[\phi_0\left(\boldsymbol{x}_n\right),\phi_1\left(\boldsymbol{x}_n\right),\cdots,\phi_9\left(\boldsymbol{x}_n\right)\right]^{\mathrm{T}}$$

$$=\left[1,x_{n,1},x_{n,2},x_{n,3},x_{n,1}^2,x_{n,2}^2,x_{n,3}^2,x_{n,1}x_{n,2},x_{n,2}x_{n,3},x_{n,1}x_{n,3}\right]^{\mathrm{T}}$$

数值实例

内在输入输出模型

$$f(x)=\frac{1}{1+\exp(-5x)}$$

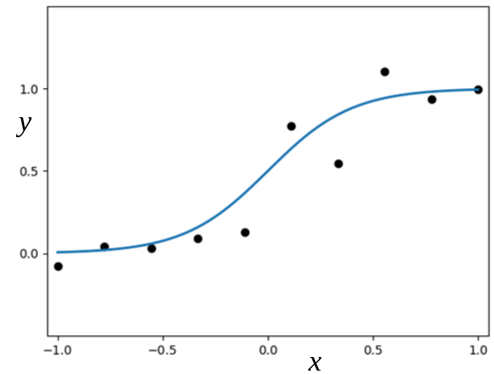
采样样本方式

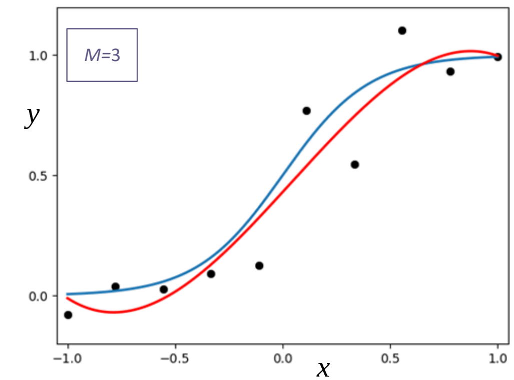
$$y_n = f(x_n)+\varepsilon_n$$

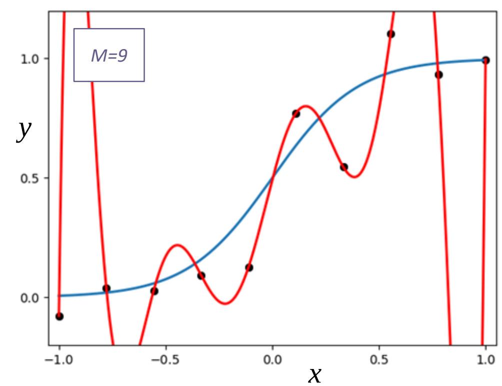
$$\varepsilon_n \sim N(0, 0.15^2)$$

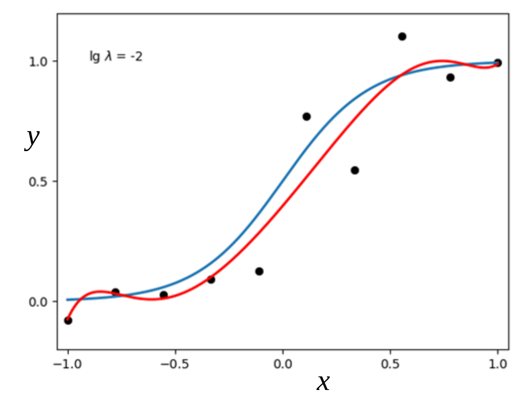
基函数回归

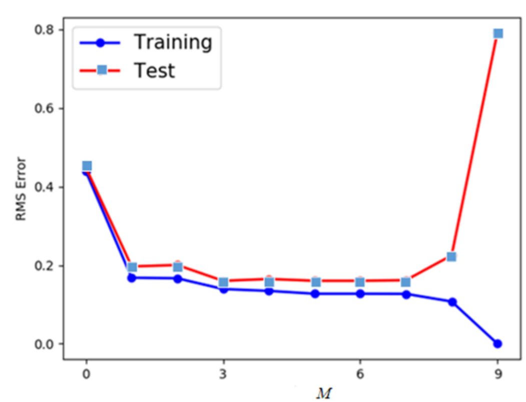
$$\phi(x_n)=\left[1, x_n, x_n^2, \cdots, x_n^M\right]^{\mathrm{T}}$$

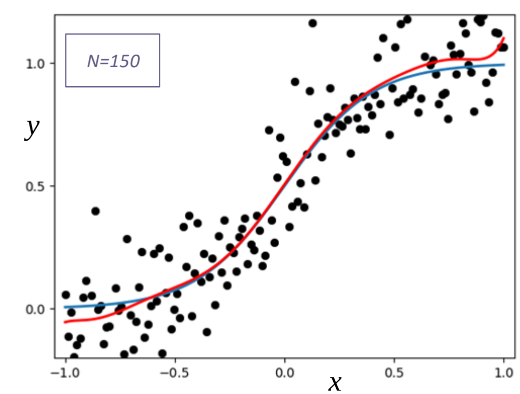












<u>右图：不同参数的实验结果</u>

<u>注意过拟合和正则化</u>

7. 机器学习模型的误差分解

以回归模型作为讨论对象，考虑平方误差

$$L(\hat{y}(\boldsymbol{x}), y) = \left(\hat{y}(\boldsymbol{x}) - y\right)^2$$

模型的误差期望（泛化误差）

$$E(L) = \int\int\left(\hat{y}(\boldsymbol{x}) - y\right)^2 p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y$$

最优回归模型是 $h(\boldsymbol{x}) = E\left(y|\boldsymbol{x}\right)$

误差分解

$$E(L) = \int\int\left(\hat{y}(\boldsymbol{x}) - h(\boldsymbol{x}) + h(\boldsymbol{x}) - y\right)^2 p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y$$

$$= \int\left(\hat{y}(\boldsymbol{x}) - h(\boldsymbol{x})\right)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \int\int\left(E\left(y|\boldsymbol{x}\right) - y\right)^2 p(\boldsymbol{x}, y)\mathrm{d}\boldsymbol{x}\mathrm{d}y$$

模型的误差分解

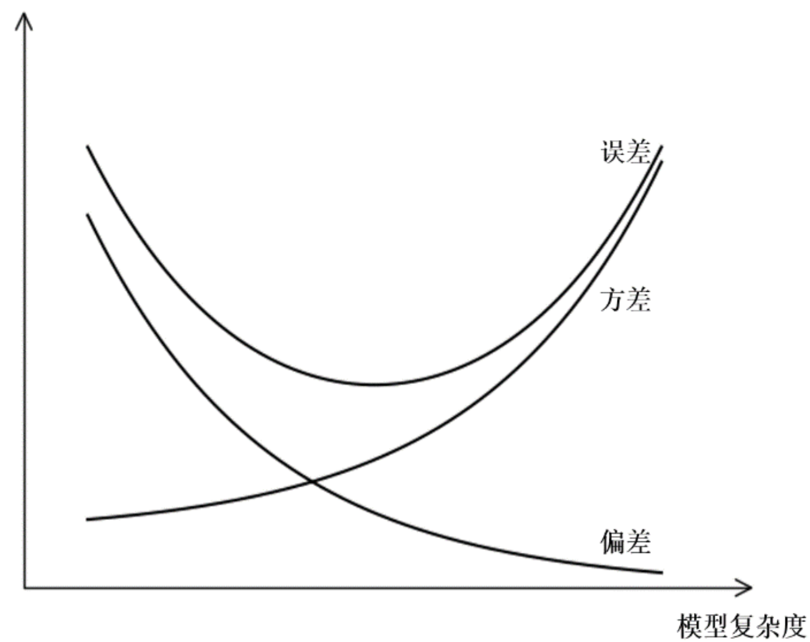
$$E(L)=$$

$$=\int\left[E_D\left(\hat{y}\left(\boldsymbol{x};\boldsymbol{D}\right)\right)-h(\boldsymbol{x})\right]^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$$+\int E_{\boldsymbol{D}}\left\{\left[\hat{y}\left(\boldsymbol{x};\boldsymbol{D}\right)-E_{\boldsymbol{D}}\left(\hat{y}\left(\boldsymbol{x};\boldsymbol{D}\right)\right)\right]^2\right\} p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$$+\int\int\left(E\left(y|\boldsymbol{x}\right)-y\right)^2 p(\boldsymbol{x},y)\mathrm{d}\boldsymbol{x}\mathrm{d}y$$

$$=\left(bias\right)^2+\text{方差}+\text{固有误差}$$

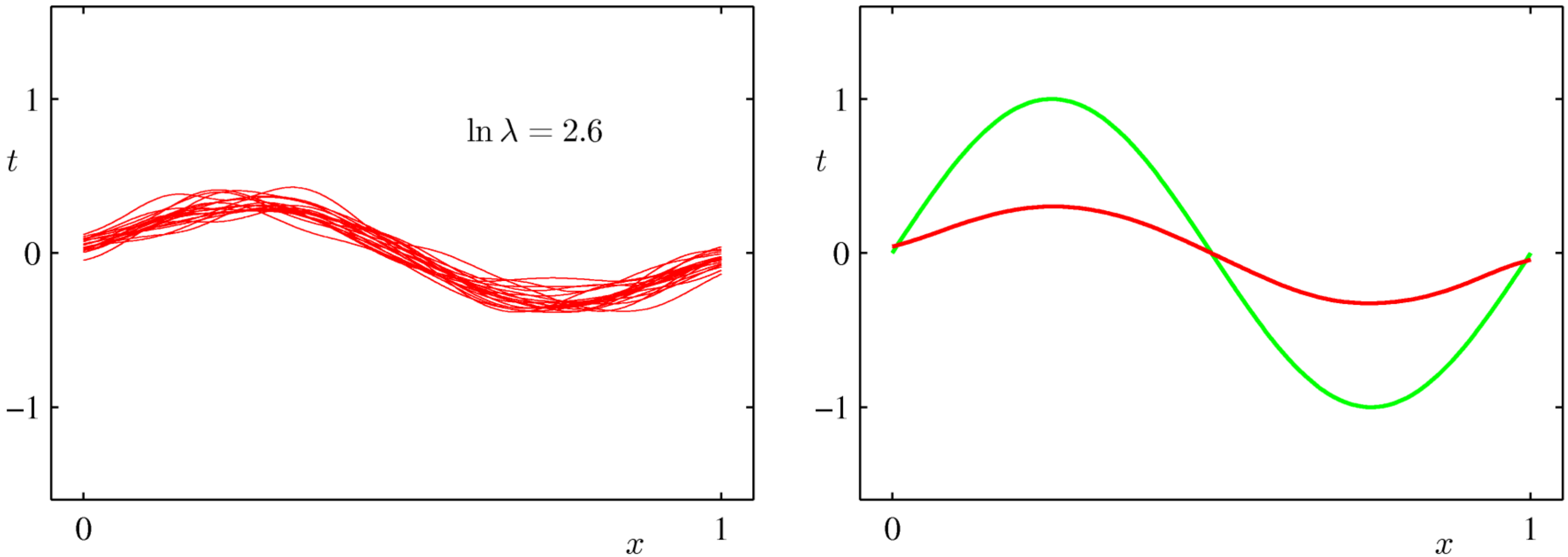


泛化误差由三部分组成:
偏、方差和固有误差

模型简单，方差小，偏大;
模型复杂，方差大，偏小

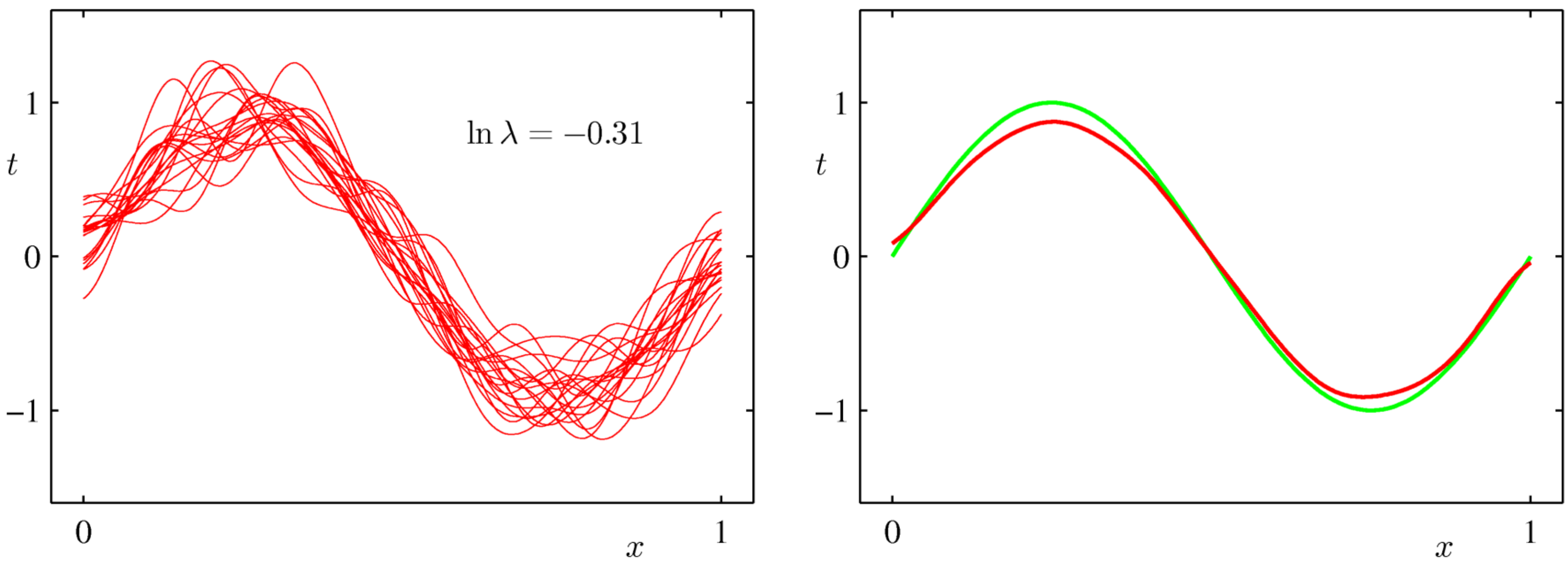
误差分解的数值实例

例: 25 个数据集，在基函数回归下，变化正则化参数 λ. 模型误差的方差和偏折中示例



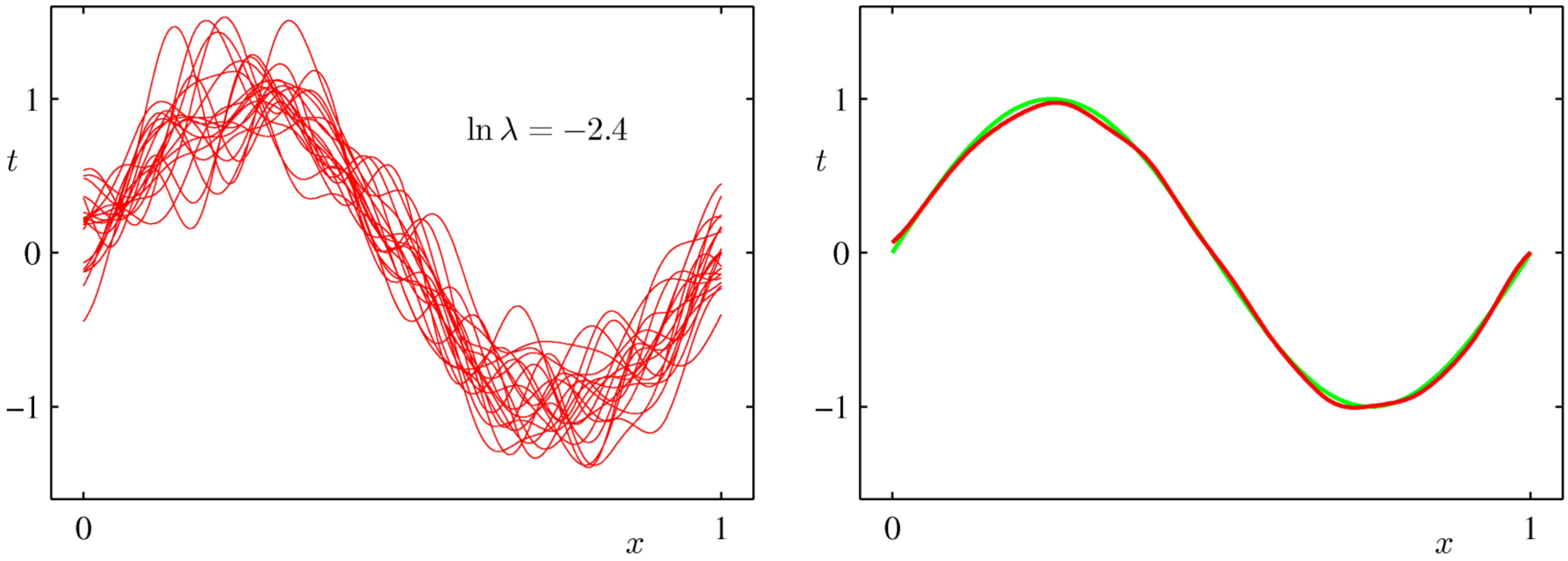
误差分解的数值实例

例: 25 个数据集，在基函数回归下，变化正则化参数 λ. 模型误差的方差和偏折中示例



误差分解的数值实例

例: 25 个数据集，在基函数回归下，变化正则化参数 λ. 模型误差的方差和偏折中示例



误差分解实例

$f(x)$无法直接观测到，采样过程为

$$y = f(x) + v$$

采样数据构成**I.I.D**数据集 $\{x_i, y_i\}_{i=1}^{N}$

采用K近邻回归算法训练模型为
（非参数模型，且 K 越小对应模型越复杂）

$$\hat{y} = \hat{f}(x) = \frac{1}{K}\sum_{l=1}^{K} y_{(l)}$$

该问题的统计最优模型为

$$h(x) = E(y|x) = f(x)$$

直接求得误差分解为

$$
\begin{aligned}
E(L) &= E\left\{\left(\hat{y}-h(x)\right)^2\right\}+\sigma_v^2 \\
&= E\left\{\left(\left(\frac{1}{K}\left(\sum_{l=1}^{K} f\left(x_{(l)}\right)+v_l\right)-f(x)\right)\right)^2\right\}+\sigma_v^2 \\
&= E\left\{\left[\frac{1}{K}\sum_{l=1}^{K} f\left(x_{(l)}\right)-f(x)\right)\right]^2\right\}+E\left\{\left(\frac{1}{K}\sum_{l=1}^{K} v_l\right)^2\right\}+\sigma_v^2 \\
&= \left[\frac{1}{K}\sum_{l=1}^{K} f\left(x_{(l)}\right)-f(x)\right)\right]^2+\frac{\sigma_v^2}{K}+\sigma_v^2
\end{aligned}
$$

第一项是偏，K越大模型越简单，偏越大；第二项是方差，K越大，模型越简单方差越小

-**注释** 测试误差和泛化误差曲线呈现一种类似“U”曲线。对于传统的单一机器学习模型“U”曲线具有一般性；

-但在深度学习中，当深度网络复杂度达到一定规模后，测试误差的表现更加复杂；

-对于集成学习中一些方法，如随机森林和提升算法，测试误差一般也并没有呈现出“U”曲线，换言之，集成学习更不易出现过拟合问题。

-机器学习是仍在快速发展中的领域，在发展中一些传统结论，可能被不断补充和修改。