第7章 统计学习理论

机器学习为什么是可行的?

大纲

- 结构风险最小化
- 误差的偏差-方差分解
- 学习曲线

Recall: 最小化风险决策

- 决策时引入损失函数或代价函数，描述每个决策所付出的代价的大小。
- 期望风险：平均损失

$$R_{\exp}(\hat{y}(\boldsymbol{x})) = \int L(\hat{y}(\boldsymbol{x}), y)p(\boldsymbol{x}, y)d\boldsymbol{x}dy = \mathbb{E}[L(\hat{y}(\boldsymbol{x}), y)]$$

$$= \int \left(\int L(\hat{y}(\boldsymbol{x}), y)p(y|\boldsymbol{x})dy \right) p(\boldsymbol{x})d\boldsymbol{x} = \mathbb{E}_{\boldsymbol{x}}[R(\hat{y}(\boldsymbol{x})|\boldsymbol{x})]$$

- 其中$R(\hat{y}(\boldsymbol{x})|\boldsymbol{x}) = \int L(\hat{y}(\boldsymbol{x}), y)p(y|\boldsymbol{x})dy$为给定样本时的条件风险。
- 选择对每个样本条件风险最小的决策规则$\hat{y}(\boldsymbol{x})$，将使期望风险最小化。

分类

- 对分类任务，取0-1损失：$L(\hat{y}, y) = \begin{cases} 0 & y = \hat{y} \\ 1 & y \neq \hat{y} \end{cases}$
- 条件风险为：

$$R(\hat{y}(\boldsymbol{x})|\boldsymbol{x}) = \sum_{y=1}^{C} L(c, y) P(Y = y|\boldsymbol{x}) = \sum_{y \neq c} P(Y = y|\boldsymbol{x}) = 1 - P(Y = c|\boldsymbol{x})$$

- 此时最小风险为最大后验：$\hat{y}(\boldsymbol{x}) = \underset{c}{\operatorname{argmax}} P(Y = c|\boldsymbol{x})$

回归

- 对回归任务，取L2损失： $L(\hat{y}, y) = (\hat{y} - y)^2$
- 条件风险为：

$$R(\hat{y}(\boldsymbol{x})|\boldsymbol{x}) = \int L(\hat{y}(\boldsymbol{x}), y)p(y|\boldsymbol{x})dy = \int (\hat{y}(\boldsymbol{x}) - y)^2 p(y|\boldsymbol{x})dy$$

$$= \int [\hat{y}(\boldsymbol{x})^2 - 2\hat{y}(\boldsymbol{x})y + y^2]p(y|\boldsymbol{x})dy$$

$$= \hat{y}(\boldsymbol{x})^2 - 2\hat{y}(\boldsymbol{x})\mathbb{E}[y|\boldsymbol{x}] + \mathbb{E}[y^2|\boldsymbol{x}]$$

- $R(\hat{y}(\boldsymbol{x})|\boldsymbol{x})$对$\hat{y}(\boldsymbol{x})$求导，并令其为0，得到最小风险为条件期望：

$$\hat{y}(\boldsymbol{x}) = \mathbb{E}[y|\boldsymbol{x}] = \int yp(y|\boldsymbol{x})dy$$

统计学习

- 输入空间：集合$\mathcal{X}$
- 输出空间：集合$\mathcal{Y}$
- $\boldsymbol{x} \in \mathcal{X}$，$y \in \mathcal{Y}$，$\mathcal{X}$与$\mathcal{Y}$的联合概率分布$p(\boldsymbol{x}, y)$
- 学习一个$\mathcal{X}$到$\mathcal{Y}$的映射f： $\hat{y} = f(\boldsymbol{x})$
  - f的取值范围为$\mathcal{F}$（假设空间）
- 定义在$\mathcal{X}$,$\mathcal{Y}$和f 上的损失函数 $L(f(\boldsymbol{x}), y) \to \mathbb{R}$，其中$\mathbb{R}$表示实数集合
- 统计学习的目标：**找一个映射f，使得f的期望风险最小**
- 所以，统计学习本质上是一个最优化问题

$$f^* = \underset{f}{\operatorname{argmin}}\, \mathbb{E}\big(L(f(\boldsymbol{x}), y)\big) = \underset{f}{\operatorname{argmin}} \int L(f(\boldsymbol{x}), y) p(\boldsymbol{x}, y) d\boldsymbol{x} dy$$

- 但统计学习中，我们只有训练样本。可以用训练样本上的风险替代期望风险？二者之间的差异有多大？

没有免费午餐（No Free Lunch，NFL）定理

- NFL定理：在真实目标函数为均匀分布的条件下，所有算法的期望性能相同。
- 在比较两个机器学习算法时，不存在一种算法在解决所有的问题时都优于另一种算法。
- 如果考虑所有潜在问题，那么所有学习算法的总体表现都是一致的。评价学习算法的优劣，必须结合具体的问题进行分析。
- 现实中，定理成立的条件（真实目标函数为均匀分布）通常不满足，因此总会有一个方法在解决这一问题上总体情况上优于另一个方法。
- NFL定理的指导意义：机器学习一定要关注问题本身的特点（问题的先验知识）。只有当模型和问题匹配时，模型才能发挥最大的作用。

David H. Wolpert and William G. Macready, No Free Lunch Theorems for Optimization, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1, APRIL 1997

Hoeffding不等式

- N个独立的随机变量$Z_1, Z_2, \ldots, Z_N$，每个Z_i取值在区间$[a, b]$内，则对任意$\varepsilon > 0$，有

$$P\left(\left|\frac{1}{N}\sum_{i=1}^{N}(Z_i - \mathbb{E}[Z_i])\right| \geq \varepsilon\right) \leq e^{-\frac{2N\varepsilon^2}{(b-a)^2}}$$

- 注意：这里并不要求Z_i服从同分布。

- 证明：略

马尔可夫不等式 → 切诺夫上界

霍夫丁引理

} 霍夫丁不等式

Hoeffding不等式：训练误差与测试误差

- 下面我们以二分类为例，探讨训练误差与测试误差之间的关系。结论可推广到其他任务。
- 在机器学习中，给定N个训练样本$\{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$，
- 分类器f在N个样本上的预测结果为$f(\boldsymbol{x}_i)$，$i = 1, 2, \ldots, N$，
- 分类结果正确还是错误，记为随机变量$Z_i = \mathbb{I}(f(\boldsymbol{x}_i) \neq y_i)$，$Z_i \in [0,1]$
- 则训练集上的平均错误率为$Z = E_{\text{train}} = \frac{1}{N}\sum_{i=1}^N Z_i$，
- 测试集/总体上的期望错误率为$E_{\text{test}} = \frac{1}{N}\sum_{i=1}^N \mathbb{E}[Z_i]$

$$P(|E_{\text{train}} - E_{\text{test}}| \geq \varepsilon) \leq e^{-2N\varepsilon^2}$$

- 二者之间的差异超过ε的概率很小，且样本数N越大，概率越低。

训练误差与测试误差可能很接近，尤其当训练样本足够多时。

有限个模型的机器学习

- 上面我们讨论了对给定的某个分类器f，验证了

$$P(|E_{\text{train}}(f) - E_{\text{test}}(f)| \geq \varepsilon) \leq e^{-2N\varepsilon^2}$$

- 如果能找到最佳的f^*，使得$E_{\text{train}}(f^*) \to 0$，从而$E_{\text{test}}(f^*) \to 0$。
- 从一组有限的函数集合$\{f_1, f_2, \ldots, f_M\}$中选择训练误差最小的函数为最佳函数$f^*$。只要$E_{\text{train}}(f^*) \to 0$，则$E_{\text{test}}(f^*) \to 0$
- 令$P(|E_{\text{train}}(f^*) - E_{\text{test}}(f^*)| \geq \varepsilon) = P(\text{坏}f^*)$

$$\leq P(\text{坏}f_1 \text{ 或者 坏}f_2 \ \ldots \text{或者 坏}f_M)$$

$$\leq P(\text{坏}f_1) + P(\text{坏}f_2) + \cdots + P(\text{坏}f_M)$$

$$= Me^{-2N\varepsilon^2}$$

- 若M是有限的、N较大，则$P(\text{坏}f^*) \leq Me^{-2N\varepsilon^2}$小，则机器学习可行。

无限个模型的机器学习

- 但事实上通常我们是从包含无限个模型的函数族中学习。
  - 例如线性模型，$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}$
  - $\boldsymbol{w}$的取值有无限多种，对应无限多个函数。
- 从而

$$P(\text{坏}f^*) \leq Me^{-2N\varepsilon^2} = \frac{M}{e^{2N\varepsilon^2}}$$

- 是个很大的数。

- 问题：上述结论的问题在于推导中利用了不等式

$$P(\text{坏}f^*) \leq P(\text{坏}f_1) + P(\text{坏}f_2) + \cdots + P(\text{坏}f_M)$$

- 实际上坏f_1、坏f_2、…，基本上是等效的，使用上述联合上界会使得最后得到的上界过松。

VC维

- Vapnik和Chervonenkis基于VC（Vapnik-Chervonenkis）维的概念，提出VC不等式，得到更紧致的上界：

$$P(|E_{\text{train}}(f^*) - E_{\text{test}}(f^*)| \geq \varepsilon) \leq 4\frac{\sum_{i=0}^{k-1}\binom{2N}{i}}{e^{\frac{1}{8}N\varepsilon^2}}$$

- 其中k是第一个无法被函数族打散的点的数目

- 因为$k-1$是最后能被打散的点的数目，定义 VC 维为：$d_{\text{vc}} = k - 1$

- 代入上述不等式，得到

只保留多项式的最高项

$$P(|E_{\text{train}}(f^*) - E_{\text{test}}(f^*)| \geq \varepsilon) \leq 4\frac{\sum_{i=0}^{d_{\text{vc}}}\binom{2N}{i}}{e^{\frac{1}{8}N\varepsilon^2}} \leq 4\frac{(2N)^{d_{\text{vc}}}+1}{e^{\frac{1}{8}N\varepsilon^2}}$$

插播：打散

正射线

某个点的右边全是正例

正间隔

某两个点的中间全是正例

一维感知器

某个点的一边全是正例

二维感知器

- **正射线**：二分类不能打散 2 个点，因为在正射线定义下红心一定要在绿球右边。
- **正间隔**：二分类不能打散 3 个点，因为在正间隔定义下红心一定要在绿球中间。
- **一维感知器**：二分类不能打散 3 个点，因为在红心或绿球一定要连在一起。
- **二维感知器**：二分类不能打散 4 个点

VC维

- 根据VC不等式

$$P(|E_{\text{train}}(f^*) - E_{\text{test}}(f^*)| \geq \varepsilon) \leq 4\frac{\sum_{i=0}^{d_{\text{vc}}}\binom{2N}{i}}{e^{\frac{1}{8}N\varepsilon^2}} \leq 4\frac{(2N)^{d_{\text{vc}}}+1}{e^{\frac{1}{8}N\varepsilon^2}}$$

- 只要 d_{vc}是有限的，当N很大时，不等式右边是一个很小的数，则真实误差 $E_{\text{test}}(f^*)$ 逼近训练误差 $E_{\text{train}}(f^*)$，函数f^* 有很好的泛化能力。

- 所以有限的VC维是机器学习可行的条件。
  - 无需知道数据的分布
  - 只需知道训练样本和函数集合

模型复杂度

■设定一个概率δ，计算样本数N和容忍度ε之间的关系：

$$P(|E_{\text{train}}(f^*) - E_{\text{test}}(f^*)| \geq \varepsilon) \leq 4\frac{(2N)^{d_{\text{vc}}} + 1}{e^{\frac{1}{8}N\varepsilon^2}} = \delta$$

■得到 $\varepsilon = \sqrt{\frac{8}{N}\ln\left(4\frac{(2N)^{d_{\text{vc}}} + 1}{\delta}\right)}$

■因此，在$1 - \delta$的概率下，$|E_{\text{train}}(f^*) - E_{\text{test}}(f^*)| < \varepsilon$

$$E_{\text{train}}(f^*) - \sqrt{\frac{8}{N}\ln\left(4\frac{(2N)^{d_{\text{vc}}} + 1}{\delta}\right)} \leq E_{\text{test}}(f^*) \leq E_{\text{train}}(f^*) + \sqrt{\frac{8}{N}\ln\left(4\frac{(2N)^{d_{\text{vc}}} + 1}{\delta}\right)}$$

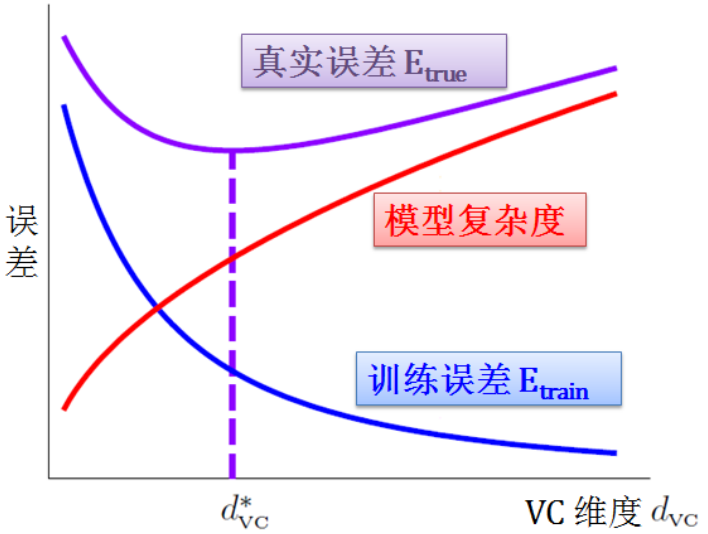
$$E_{\text{train}}(f^*) - \Omega(d_{\text{vc}}, N, \delta) \leq E_{\text{test}}(f^*) \leq E_{\text{train}}(f^*) + \Omega(d_{\text{vc}}, N, \delta)$$

模型复杂度

- $E_{\text{train}}(f^*) - \Omega(d_{\text{vc}}, N, \delta) \leq E_{\text{test}}(f^*) \leq E_{\text{train}}(f^*) + \Omega(d_{\text{vc}}, N, \delta)$

- $\Omega(d_{\text{vc}}, N, \delta) = \sqrt{\frac{8}{N} \ln\left(4\frac{(2N)^{d_{\text{vc}}+1}}{\delta}\right)}$ 称为模型复杂度
  - 假设空间$\mathcal{F}$越大，d_{vc}越大，$\Omega(d_{\text{vc}}, N, \delta)$越大，模型越难学习。

- 当d_{vc}增大时，训练误差减少（模型越复杂，越容易解释训练集），模型复杂度增大。
- 因为测试误差 = 训练误差 + 模型复杂度，因此测试误差不是d_{vc}的单调函数。
  - 最佳d_{vc}（d_{vc}^*）对应的测试误差最小



样本复杂度

- 我们也可以设定想要的容忍度ε，看需要多少样本数N能实现，即计算样本复杂度：

$$\varepsilon = \sqrt{\frac{8}{N}\ln\left(4\frac{(2N)^{d_{\mathrm{vc}}}+1}{\delta}\right)} \qquad \Rightarrow N = \frac{8}{\varepsilon^2}\ln\left(4\frac{(2N)^{d_{\mathrm{vc}}}+1}{\delta}\right)$$

- 上式左右两边都含N，需要用迭代方法 (如牛顿法) 解出。
- 例如：设定
  - $\varepsilon = 0.1$，希望 $E_{\mathrm{test}}(f^*)$ 和$E_{\mathrm{train}}(f^*)$ 之间不要超过 0.1
  - $\delta = 0.1$，有 90% 的可能上述情况会发生
- 经过迭代法算出：
  - 当 $d_{\mathrm{vc}} = 3$时，$N \approx 30000$
  - 当 $d_{\mathrm{vc}} = 4$时，$N \approx 40000$
- 因此理论上来讲，$N \approx 10000 d_{\mathrm{vc}}$。但是从实践上来讲 $N \approx 10 d_{\mathrm{vc}}$。

样本复杂度

- 为什么需要的样本数量可以从 10000 倍减到 10 倍呢?
- 因为在推导时，我们用
  - 霍夫丁不等式适用于**任何数据分布**和**任何目标函数**
  - VC 维度适用于**任何假设空间**
- 联合上界适用于**最差的状况**，所以因为上面推出的 VC 上界很松。
  - 在实践时， 各种“任何”和“最差” 不太可能同时发生。

机器能学习

- 虽然机器学习是可行的，但要使机器能学好，需要的几个条件：
  - **好的假设空间**：使得训练误差和真实误差能够接近
  - **好的数据**：数据足够多，使得训练误差和真实误差很接近
  - **好的算法**：算法可以选出一个训练误差很小的假设
  - **好的运气**：前三点说明的在概率上近似正确 (probably approximately correct, PAC)，最后还需要一点运气，使得坏事不会发生。

- 在模型复杂度上，找一个**最优 VC 维度**最小化真实误差。
- 在样本复杂度上，至少用“**10 倍的 VC 维度**”数量的训练数据。

结构风险最小化

- 我们希望期望风险最小化

$$R_{\exp}(f) = \int L(f(\boldsymbol{x}), y)p(\boldsymbol{x}, y)d\boldsymbol{x}dy$$

- 但机器学习中，只给定了训练样本，无法计算$R_{\exp}(f(\boldsymbol{x}))$，只能计算经验风险

$$R_{\text{emp}}(f) = \frac{1}{N}\sum_{i=1}^{N} L(f(\boldsymbol{x}_i), y_i)$$

- 经验风险最小化会产生过拟合 → 结构风险最小化

$$R_{\text{str}}(f) = \frac{1}{N}\sum_{i=1}^{N} L(f(\boldsymbol{x}_i), y_i) + \lambda R(f)$$

$$E_{\text{test}}(f^*) \leq E_{\text{train}}(f^*) + \Omega(d_{\text{vc}}, N, \delta)$$

奥卡姆剃刀（Occam's Razor）原理

- “Entities” (or explanations) should not be multiplied beyond necessity. 如无必要，勿增实体
- Among competing hypotheses, the one with the fewest assumptions should be selected. 在所有的假设中，应该选择假定条件最小的那一个。
- For PR/ML, NOT use machines that are more complicated than necessary. 模式识别/机器学习应该不使用哪些过于复杂（超过必要）的模型。
  - “necessary” can be determined by the quality of fit to the training data. 必要由（与训练数据）匹配的质量决定。

最小描述长度准则（MDL Principle）

- MDL （Minimizing description length）准则与奥卡姆剃刀原理等价
- 最小化：模型的算法复杂度 + 用该模型描述训练数据的长度

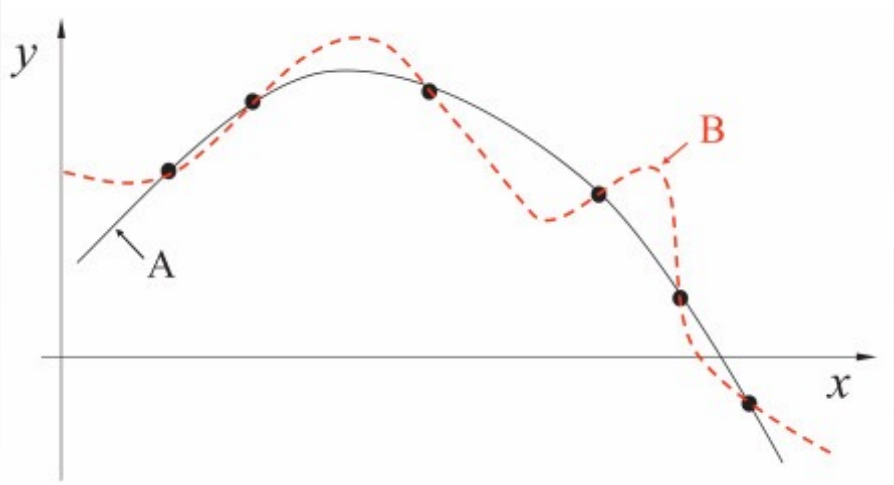
$$K(f, \mathcal{D}) = K(f) + K(\mathcal{D}\ using\ f)$$

- $K()$：Kolmogorov 复杂度
- 理论上，当数据越来越多时，基于MDL准则设计的分类器会收敛到理想模型。
- 关注点是模型复杂度，更倾向于简单模型（$K(f)$小）。

例：

■训练数据和模型A&B

- A线和B线都能够很好的拟合这几个数据点。
- 哪条曲线更好？

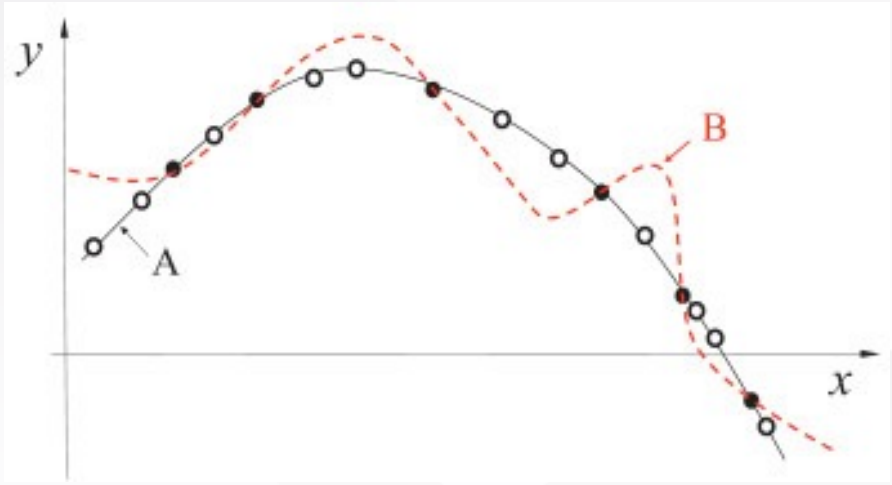


仅仅从这几个数据点来看，我们无法判断哪个更好，或者说，A和B一样好。

例：

■ 更多测试数据1（空心点）

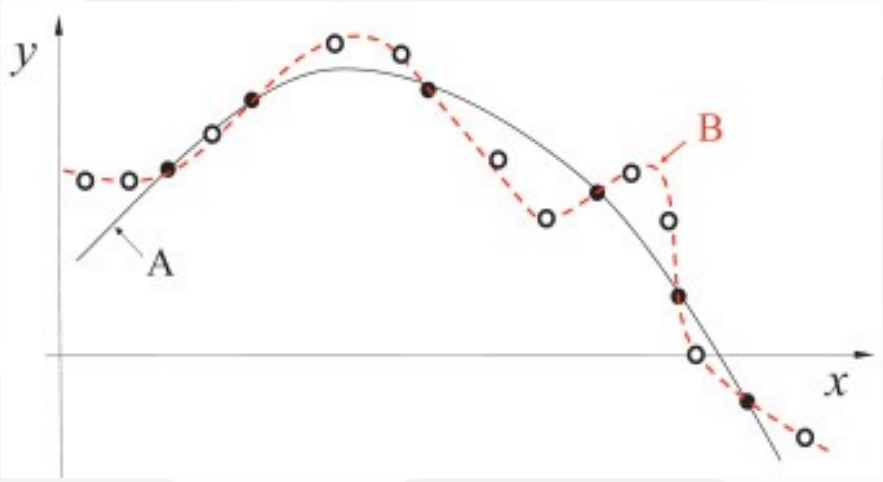
- A更好



NFL：具体哪一个函数更好，取决于数据本身的规律。而这个规律，从有限的观测数据中，是不可能绝对准确把握的。

■更多测试数据2（空心点）

- B更好



Occam ‘s Razor：A更好，因为它足够简单，且拟合得足够好。这是因为我们所面临的多数问题并不复杂，通常使用比较简单的方法就可以取得很好的效果。

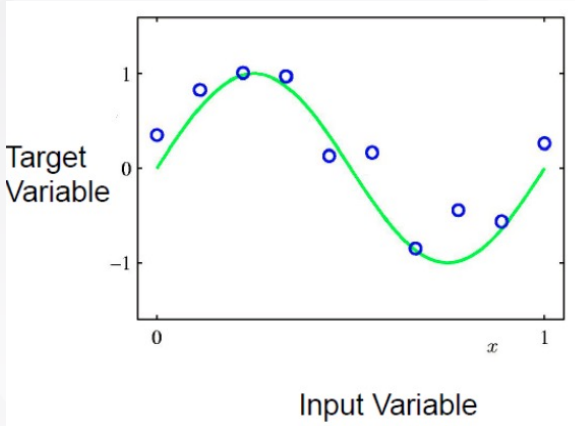
机器学习实践

- 机器学习可行的理论看上去很美，但实践是检测真理的唯一标准。
- 训练集：训练模型

- 验证集：选择模型
  - 用样本外误差，估计测试误差
  - 验证误差是真实误差的无偏估计，两者的差距与验证集的大小成反比
- 测试集：评估模型
  - 期望风险（真实误差）要求期望，不知道数据分布无法计算
  - 测试集是从总体选出来的部分样本，与训练集不重合，模拟没有见过但未来可能遇到的数据
  - 用测试误差估计真实误差

例: sin曲线的多项式拟合

■训练数据
- 输入：x在[0,1]均匀采样10个点
- 输出： $y = \sin(2\pi x) + \varepsilon$, $\varepsilon \sim N(0, 0.3^2)$



■机器学习：利用训练数据来找到一个预测函数 f

■函数集合（假设空间）：多项式

$$f(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_m x^m$$

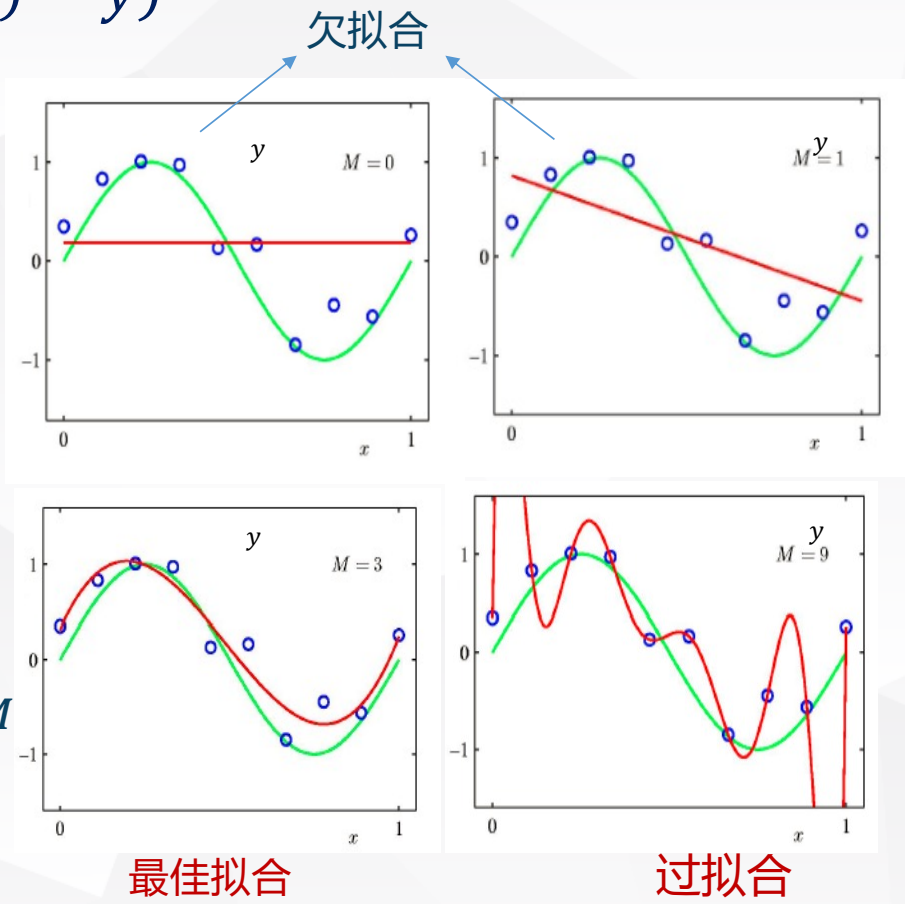
例: sin曲线的多项式拟合

- 损失函数取L2损失：$L(f(\boldsymbol{x}), y) = (f(\boldsymbol{x}) - y)^2$
- 目标：经验风险最小

$$R_{\exp} = \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2$$

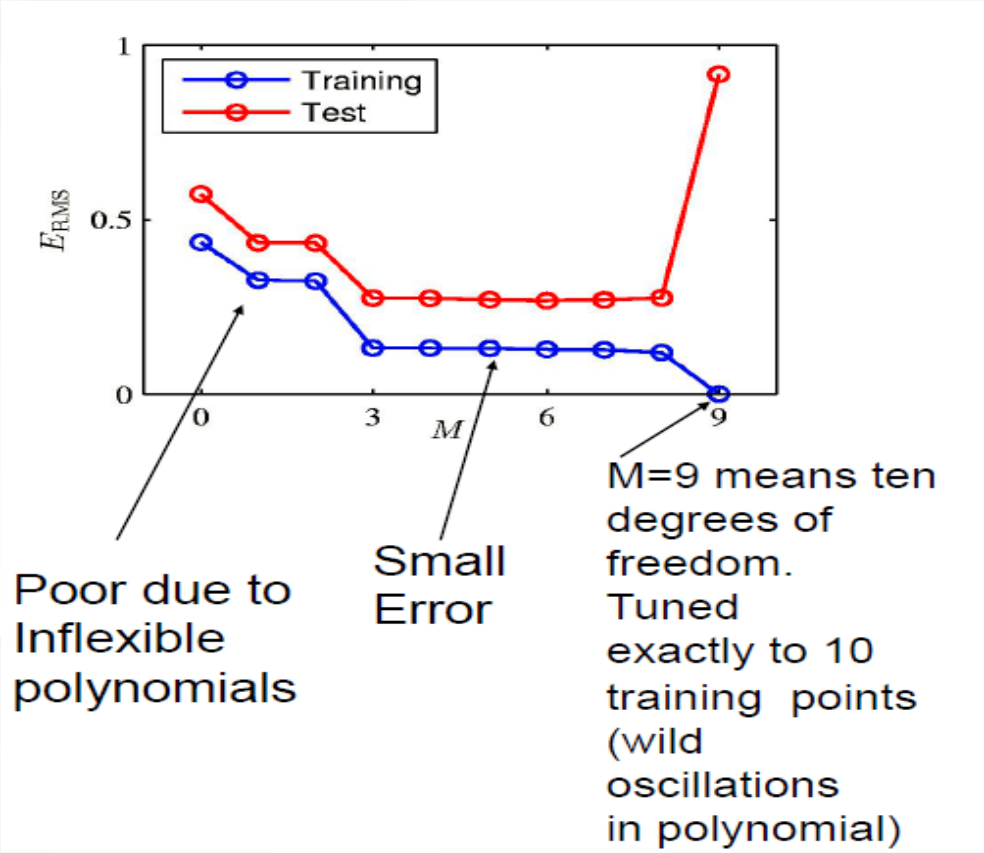
- 当$N = 10$时，
- 不同M阶数多项式的拟合结果：红线

$$f(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M$$



例: sin曲线的多项式拟合

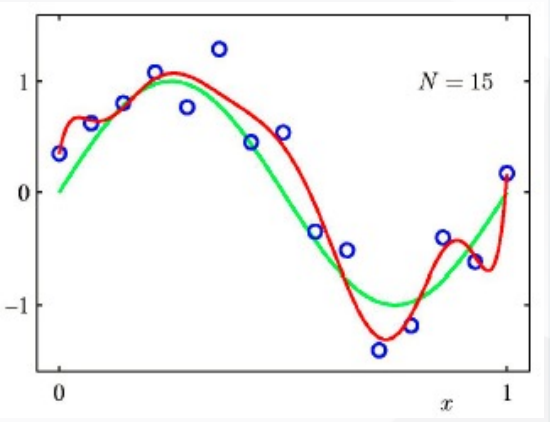
- 当$N = 10$时，各阶多项式的训练误差和测试误差（均方根误差）
  - 另外采样100个测试样本

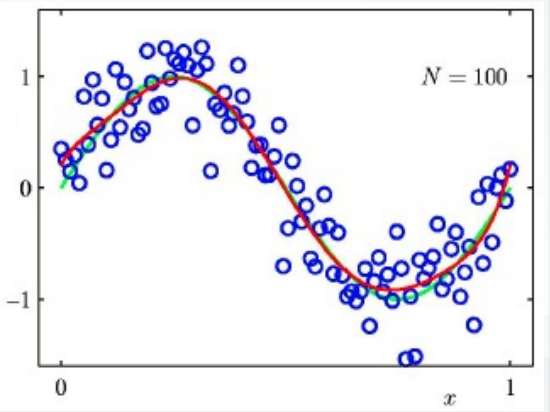


$$E_{RMS}(f) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2}$$

例: sin曲线的多项式拟合

- 当$N = 15$，100时，9阶数多项式的拟合结果





- 兼顾两个方面:
  - 模型对训练数据拟合得好：需要复杂的模型
  - 模型具有一定的能力，容忍测试数据的不同：需要稳定的模型（不那么复杂的模型）
- 模型复杂度与数据集的大小
  - 对于一个给定复杂度的模型，过拟合问题会随着训练数据集的增加而减轻。
  - 训练数据集越大，越能支持越复杂的模型。
  - 数据量大小：大于模型自适应参数数目的5-10倍

例: sin曲线的多项式拟合

■不同阶多项式拟合的系数

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

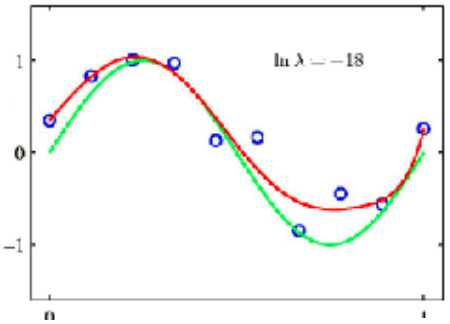
随着M的增加，系数的绝对值增加

当$M=9$时，对目标变量中的噪声也进行了很好地微调。

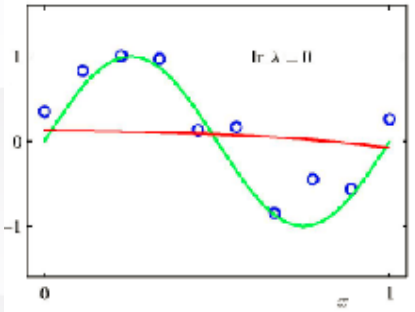
正则项的作用

- $M = 9$ 时,带L2正则的多项式
- 目标：结构风险最小 $J(f,\lambda) = \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\sum_{j=1}^{M} w_m^2$

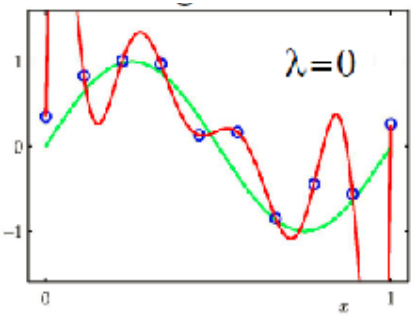
最优



大正则系数



无正则项



无正则项 $\lambda=0$　　大正则系数 $\lambda=1$

	$\ln\lambda = -\infty$	$\ln\lambda = -18$	$\ln\lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

正则项的作用

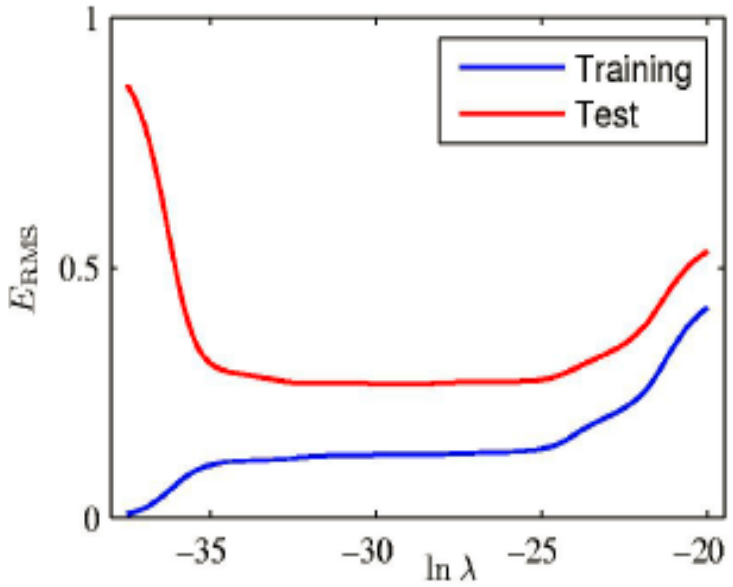
■结构风险：$J(f,\lambda)=\frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i)-y_i)^2+\lambda\sum_{j=1}^{M}w_m^2$

■λ控制模型的复杂度，因此也控制着过拟合的程度

- 类似于M的选择

■方法: 验证

- 训练集：对于不同的M或λ，确定系数$\mathbf{w}$
- 验证集：选择最优的模型复杂度（M或λ）



$M=9$

大纲

- 结构风险最小化
- 误差的偏差-方差分解
- 学习曲线

例：sin曲线拟合

- 训练数据
  - 输入：x在$[0,1]$均匀采样25个点
  - 输出：$y = \sin(2\pi x) + \varepsilon$, $\varepsilon \sim N(0, 0.3^2)$
- 机器学习：利用训练数据来找到一个预测函数f
  - 函数集合（假设空间）：$M = 25$阶多项式

$$f(x, \boldsymbol{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M$$

- 目标函数：结构风险最小

$$J(f, \lambda) = \frac{1}{N} \sum_{i=1}^{N} (f(\boldsymbol{x}_i) - y_i)^2 + \lambda \sum_{j=1}^{M} w_m^2$$

  - λ在$[10^{-6}, 10^{1}]$之间的log空间均匀采样40个点

例：sin曲线拟合

- 重复$L = 100$次试验
  - 训练数据集记为$\mathcal{D}$
  - 利用训练数据集$\mathcal{D}$训练得到模型$f_{\mathcal{D}}$
  - 模型$f_{\mathcal{D}}$对测试样本$\boldsymbol{x}$进行预测，得到预测结果$\hat{y}_{\mathcal{D}} = f_{\mathcal{D}}(\boldsymbol{x})$

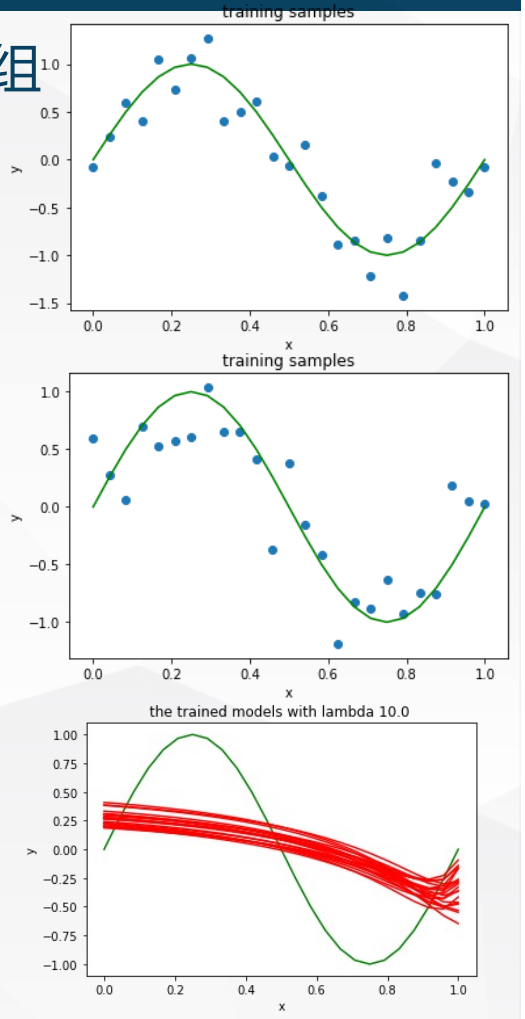
- 偏差：模型预测的期望与真实值之间的偏离程度，刻画模型本身的拟合能力（对所有训练数据得到的所有模型求平均，平均与真实值的差异）

$$bias^2(\hat{y}_{\mathcal{D}}) = (\mathbb{E}[\hat{y}_{\mathcal{D}}] - y)^2$$

- 方差：训练集的变动所导致的模型的变化，刻画数据扰动造成的影响（不同训练数据得到不同模型之间的差异）

$$\mathrm{Var}[\hat{y}_{\mathcal{D}}] = \mathbb{E}[(\hat{y}_{\mathcal{D}} - \mathbb{E}[\hat{y}_{\mathcal{D}}])^2]$$

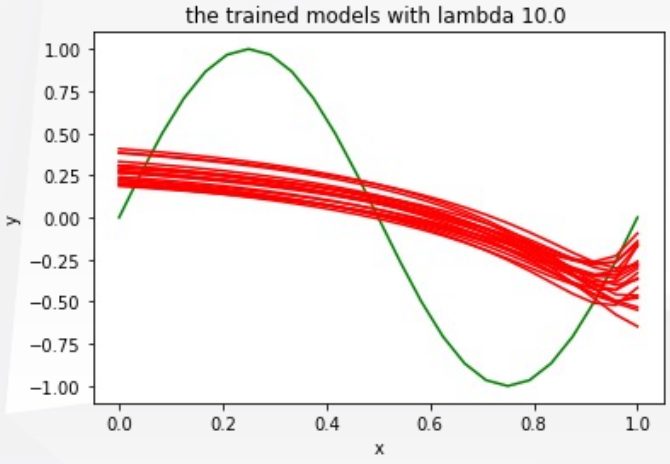
随机采样的两组训练样本集



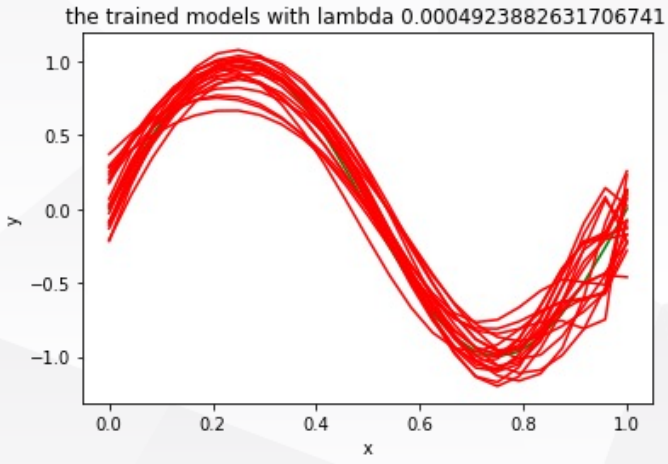
$\lambda = 10$的20个模型

■正则参数λ控制模型复杂性：对偏差和方差的影响

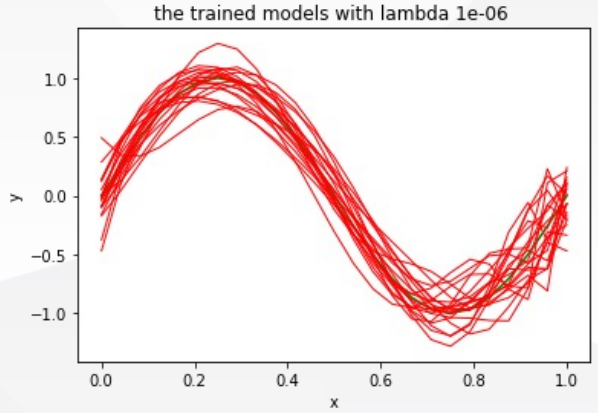
简单模型：
低方差
高偏差



最佳模型：
偏差和方差适中



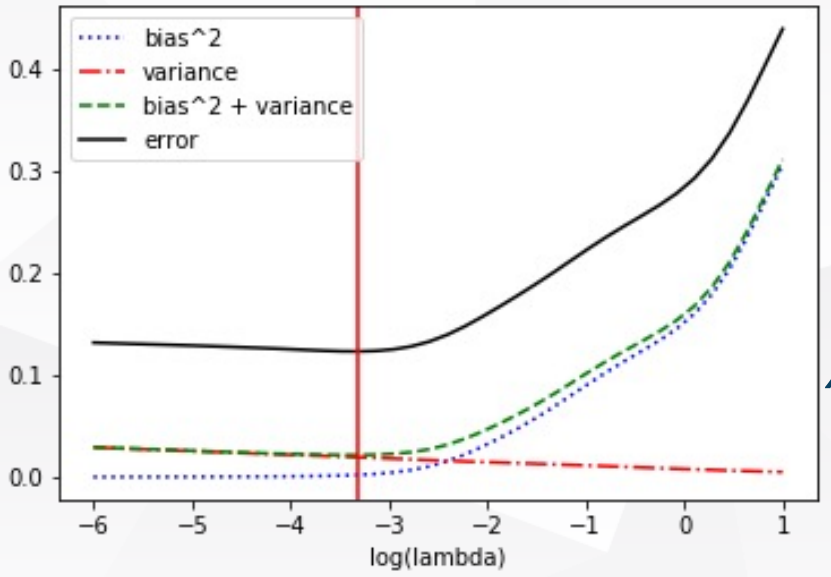
复杂模型：
高方差
低偏差



例：sin曲线拟合

- 重复$L = 100$次试验
- 偏差：$bias^2(\hat{y}_{\mathcal{D}}) = (\mathbb{E}[\hat{y}_{\mathcal{D}}] - y)^2$
- 方差：$\mathrm{Var}[\hat{y}_{\mathcal{D}}] = \mathbb{E}[(\hat{y}_{\mathcal{D}} - \mathbb{E}[\hat{y}_{\mathcal{D}}])^2]$
- 噪声：刻画学习问题本身的难度 $N(0, 0.3^2)$

$$J(f, \lambda) = \frac{1}{N}\sum_{i=1}^{N}(f(\boldsymbol{x}_i) - y_i)^2 + \lambda\sum_{j=1}^{M} w_m^2$$



λ较小：不同训练样本集得到的模型变化较大，即方差大，偏差几乎为0

λ较大：方差很小，但模型过于平滑偏差很大

误差的偏差-方差分解

■令$\mathbb{E}[\hat{y}_{\mathcal{D}}] = \bar{y}$，则

$$\mathbb{E}[(\hat{y}_{\mathcal{D}} - y)^2] = \mathbb{E}\left[\left(\hat{y}_{\mathcal{D}} - (y^* + \varepsilon)\right)^2\right]$$

$$= \mathbb{E}[(\hat{y}_{\mathcal{D}} - y^*)^2] + \mathbb{E}[\varepsilon^2]$$

$$= \mathbb{E}[(\hat{y}_{\mathcal{D}} - \bar{y}) + (\bar{y} - y^*)^2] + \mathrm{Var}[\varepsilon]$$

$$= \mathbb{E}[(\hat{y}_{\mathcal{D}} - \bar{y})^2] + \mathbb{E}[(\bar{y} - y^*)^2] - 2\mathbb{E}[(\hat{y}_{\mathcal{D}} - \bar{y})(\bar{y} - y^*)] + \mathrm{Var}[\varepsilon]$$

$$= \mathrm{Var}[\hat{y}_{\mathcal{D}}] + (\bar{y} - y^*)^2 - 2(\bar{y} - y^*)\mathbb{E}[(\hat{y}_{\mathcal{D}} - \bar{y})] + \mathrm{Var}[\varepsilon]$$

$$= \mathrm{Var}[\hat{y}_{\mathcal{D}}] + (\bar{y} - y^*)^2 - 2(\bar{y} - y^*)^2(\mathbb{E}[\hat{y}_{\mathcal{D}}] - \bar{y}) + \mathrm{Var}[\varepsilon]$$

$$= \mathrm{Var}[\hat{y}_{\mathcal{D}}] + (\bar{y} - y^*)^2 + \mathrm{Var}[\varepsilon]$$

方差　　偏差^2　　噪声

偏差-方差权衡

■对模型复杂度问题的深刻理解

- 非常灵活的模型具有低偏差和高方差。
- 相对刚性的模型有大的偏差和低的方差。
- 具有最佳预测能力的模型是使得偏差和方差之间最佳平衡的模型。

■偏差-方差分解的实际应用价值有限

- 偏差和方差无法计算，因为它依赖于x和y的真实分布。
- 偏差-方差分解基于数据集集合的平均值，而实际上**我们只有单个观测数据集。**

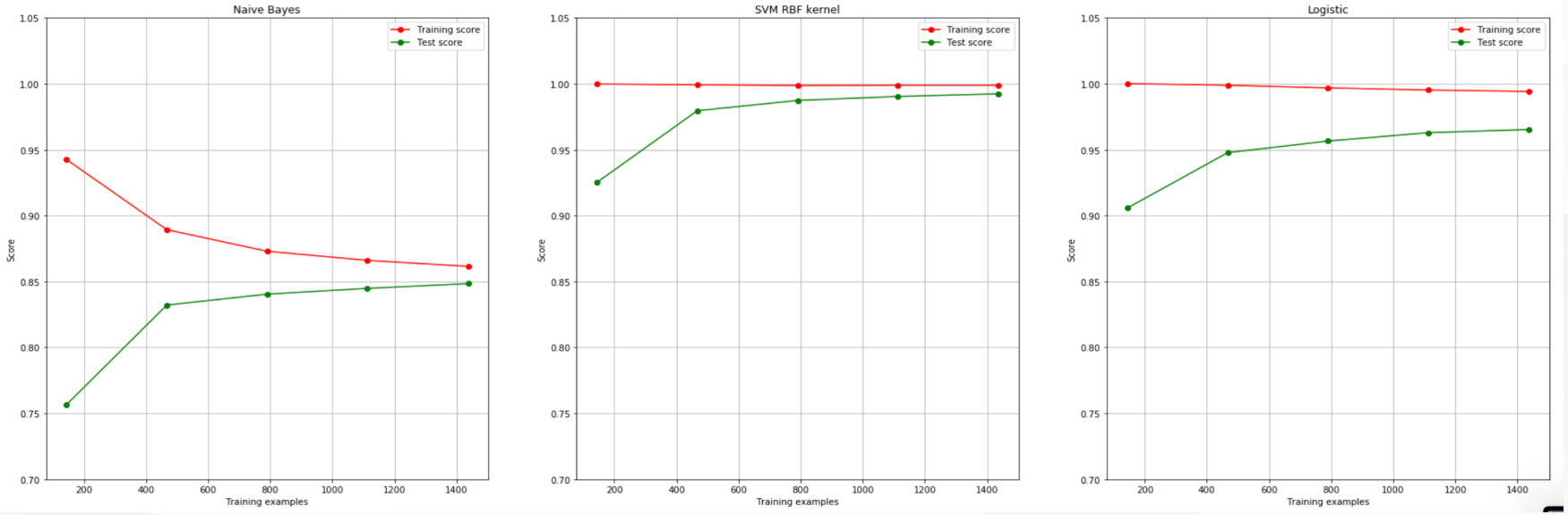
大纲

- 结构风险最小化
- 误差的偏差-方差分解
- 学习曲线

实际应用技巧

- **学习曲线**：不同训练集大小对应的训练集和验证集上的性能
  - 观察机器学习算法是否为欠拟合或过拟合
  - 亦可用于诊断偏差与方差

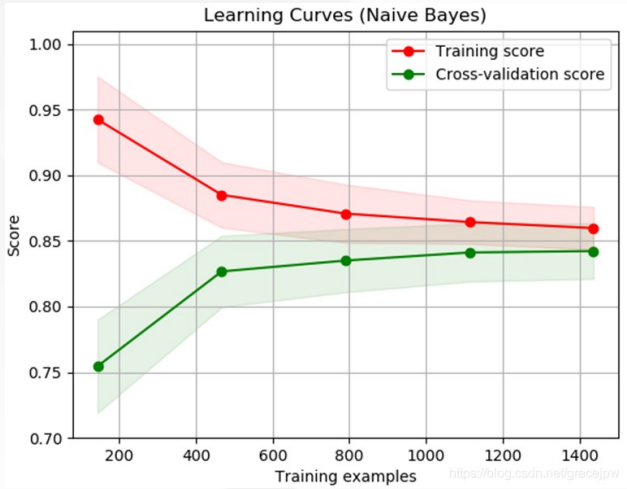
```
Naive Bayes:00:00:705469
SVM RBF kernel:00:07:008655
Logistic:00:18:424287
```



学习曲线 （Learning Curve）

■学习曲线：不同训练集大小时训练集和验证集的性能

- 横轴：训练样本的数量
- 纵轴：模型性能

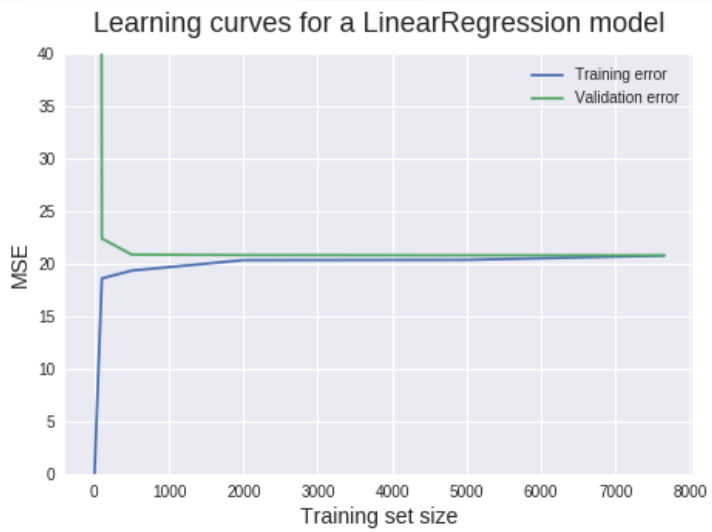


train_sizes, train_scores, validation_scores = learning_curve (*estimator, X, y, *, groups=None, train_sizes=array([0.1, 0.33, 0.55, 0.78, 1.]), cv=None, scoring=None, exploit_incremental_learning=False, n_jobs=None, pre_dispatch='all', verbose=0, shuffle=False, random_state=None, error_score=nan, return_times=False*)

返回值：训练集大小、训练集和验证集上的误差得分
参数：学习器、数据、训练集大小、交叉验证参数、性能评价指标

学习曲线 （Learning Curve）

- 学习曲线：通过画出不同训练集大小时训练集和验证集的性能
  - 训练误差随训练集增大而增大，然后趋于稳定
  - 验证误差随训练集增大而减少，然后趋于稳定
  - 二者之间的差异随训练集增大而减少，然后趋于稳定
- 不同模型区域稳定的样本集合大小不同
  - 简单模型需要更少的训练数据

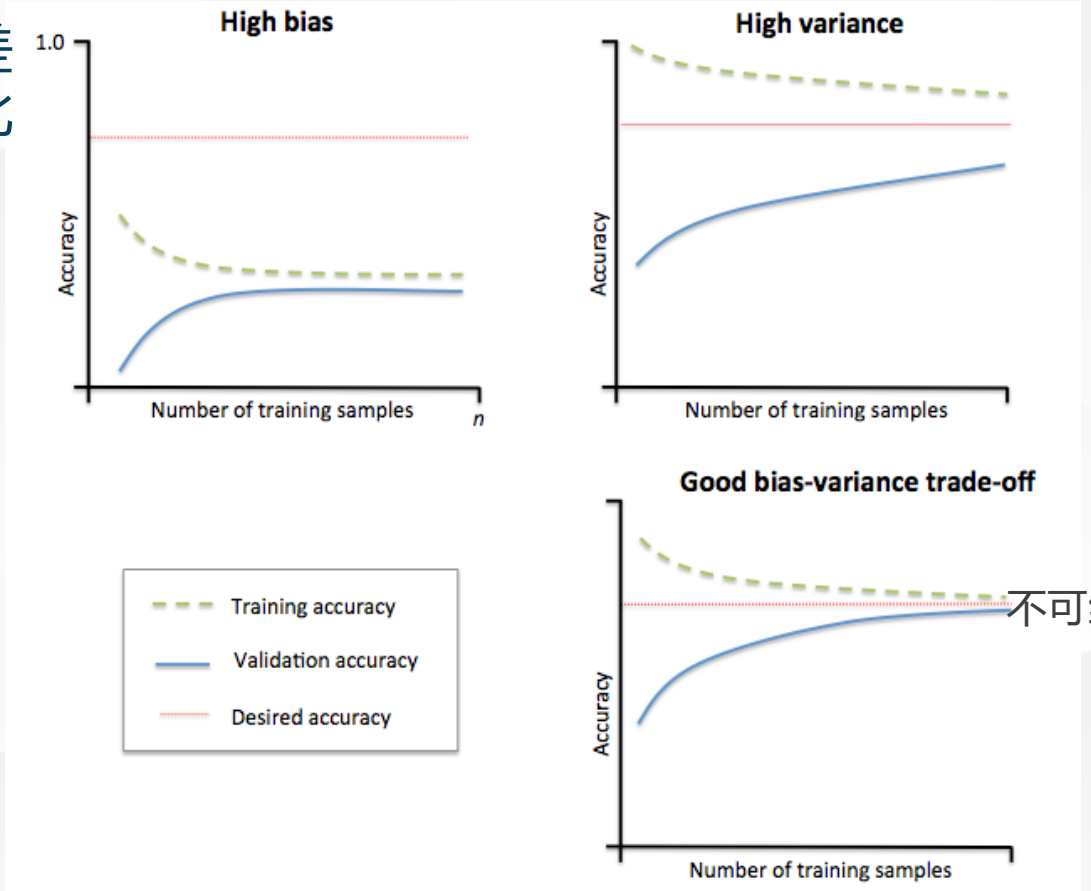


学习曲线

- 偏差：当训练误差稳定时，训练误差的大小可视为模型偏差（训练充分时，模型与训练数据的拟合程度）
  - 随机森林偏差小、线性模型偏差大
- 方差：当训练误差稳定时，训练误差与验证误差之间的差异可视为模型的方差（由于数据不同模型性能的差异）
  - 随机森林方差大、线性模型方差小

学习曲线 （Learning Curve）

验证集和训练集的误差值都很大，偏差大，此时为欠拟合



训练集误差非常小，但验证集误差远大于训练集误差，此时为过拟合

不可约误差

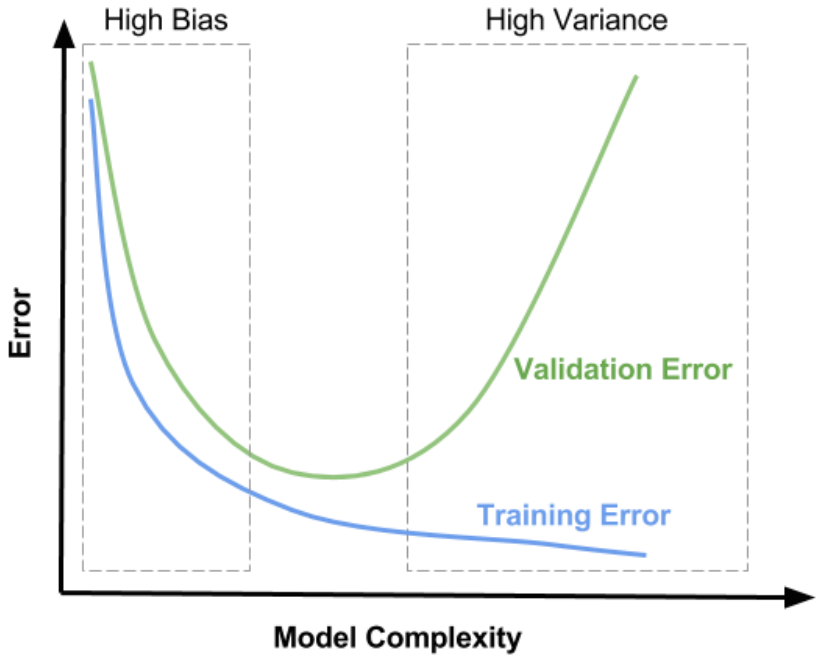
欠拟合和过拟合的外在表现

- 在实际样应用中，有时候我们很难计算模型的偏差与方差，只能通过外在表现，判断模型的拟合状态是欠拟合还是过拟合。

训练误差随着模型复杂度增加一直减小。

验证误差随着模型复杂度的变化先减小（欠拟合程度减轻）；

当模型复杂度超过一定值后，验证误差随模型复杂度增加而增大，此时模型进入过拟合状态。



提高模型性能

- 欠拟合：当模型处于欠拟合状态时，根本的办法是增加模型复杂度。
  - 修改模型架构（增大假设空间）
  - 增加模型的迭代次数（训练更充分）
  - 更多特征（增大假设空间）
  - 降低模型正则化水平（L2、L1、Dropout）

- 过拟合：当模型处于过拟合状态时，根本的办法是降低模型复杂度。
  - 修改模型架构
  - 及早停止迭代
  - 减少特征数量
  - 提高模型正则化水平
  - 扩大训练集：可以帮助解决方差问题，但对偏差通常没有明显影响

小结

- 无免费午餐定理：模型的选取要以问题的特点为根据。
- 奥卡姆剃刀：在性能相同的情况下，应该选取更加简单的模型。
- 过于简单的模型会导致欠拟合，过于复杂的模型会导致过拟合。
- 从误差分解的角度看，欠拟合模型的偏差较大，过拟合模型的方差较大。