第3章 模型评估与选择

- 经验误差与过拟合
- 评估方法
- 性能度量
- 模型性能对比

经验误差与过拟合

- 错误率&误差：
  - 错误率：错分样本的占比：　　$E = a/m$
  - 误差：样本真实输出与预测输出之间的差异
    - 训练(经验)误差：训练集上
    - 测试误差：测试集
    - 泛化误差：除训练集外所有样本

- 由于事先并不知道新样本的特征，我们只能努力使经验误差最小化；
- 很多时候虽然能在训练集上做到分类错误率为零，但多数情况下这样的学习器并不好

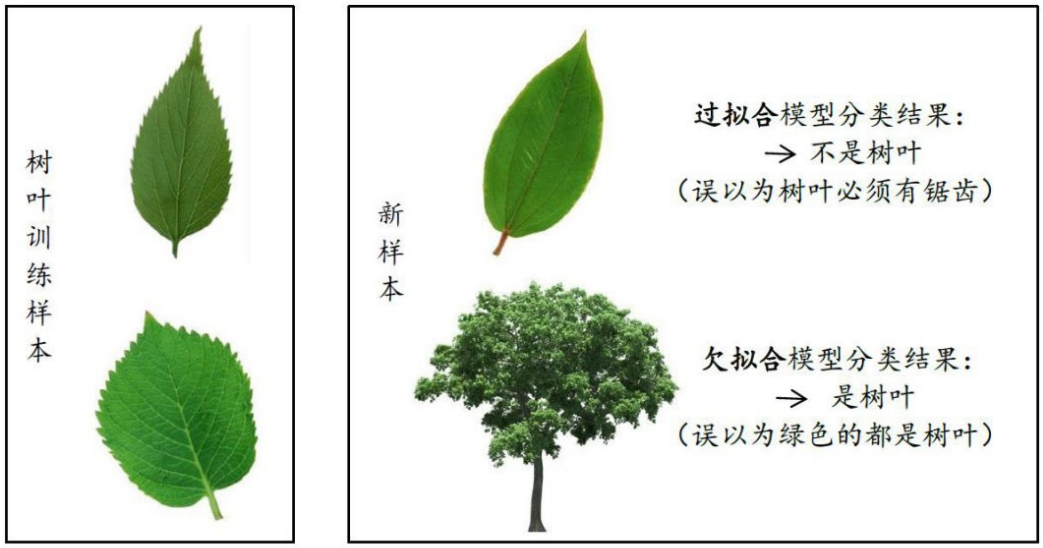
经验误差与过拟合

- **过拟合：**

  学习器把训练样本学习的“太好”，将训练样本本身的特点，当做所有样本的一般性质，导致泛化性能下降

- **欠拟合：**

  对训练样本的一般性质尚未学好



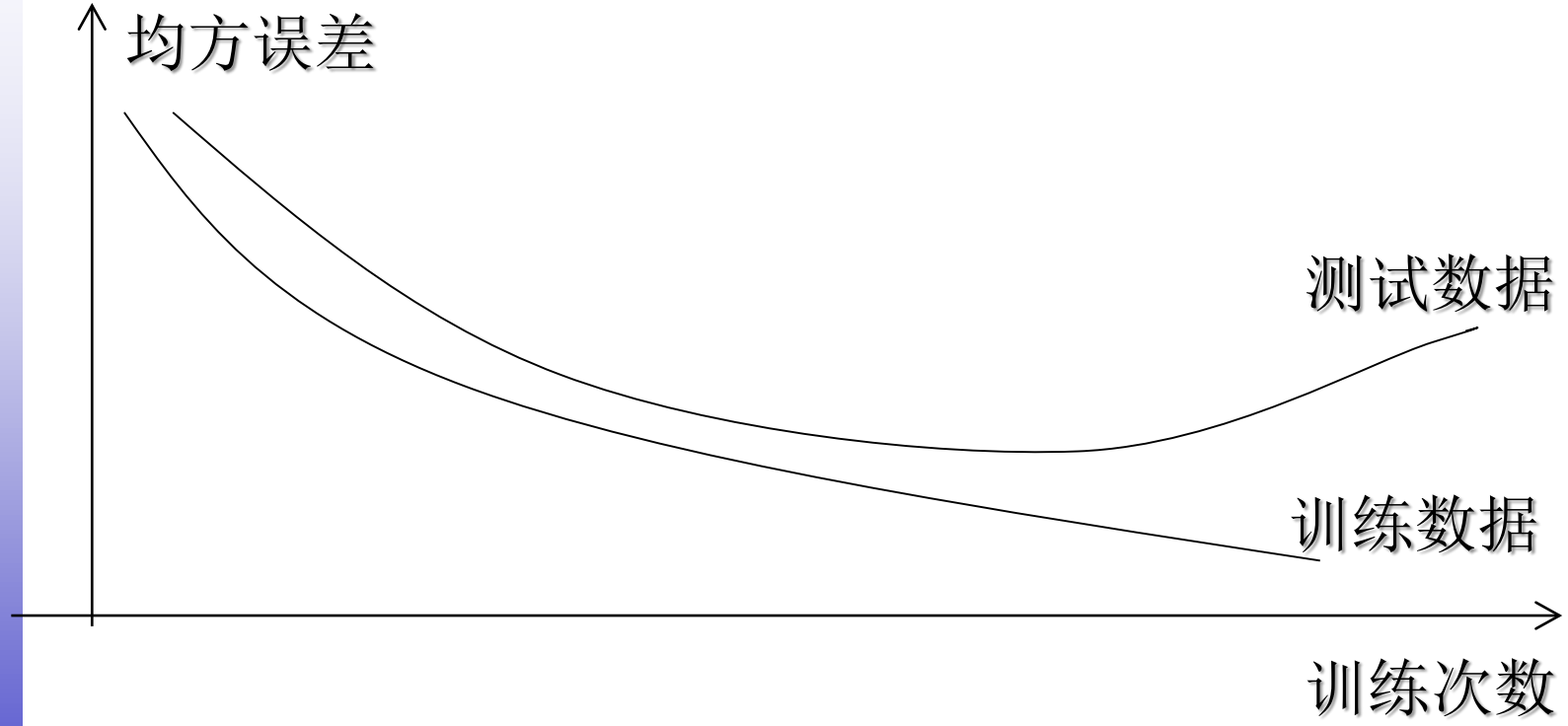
过拟合、欠拟合的直观类比

经验误差与过拟合

- 过拟合：
  - 优化目标加正则项
  - early stop
- 欠拟合：
  - 决策树:拓展分支
  - 神经网络:增加训练轮数

适可而止，过犹不及

实际操作时应该训练和测试交替进行，即每训练一次，同时用测试数据测试一遍，画出均方误差随训练次数的变换曲线



在用测试数据检验时，均方误差开始逐渐减小，当训练次数再增加时，测试检验误差反而增加，误差曲线上极小点所对应的即为恰当的训练次数，若再训练即为“过度训练”了。

评估方法

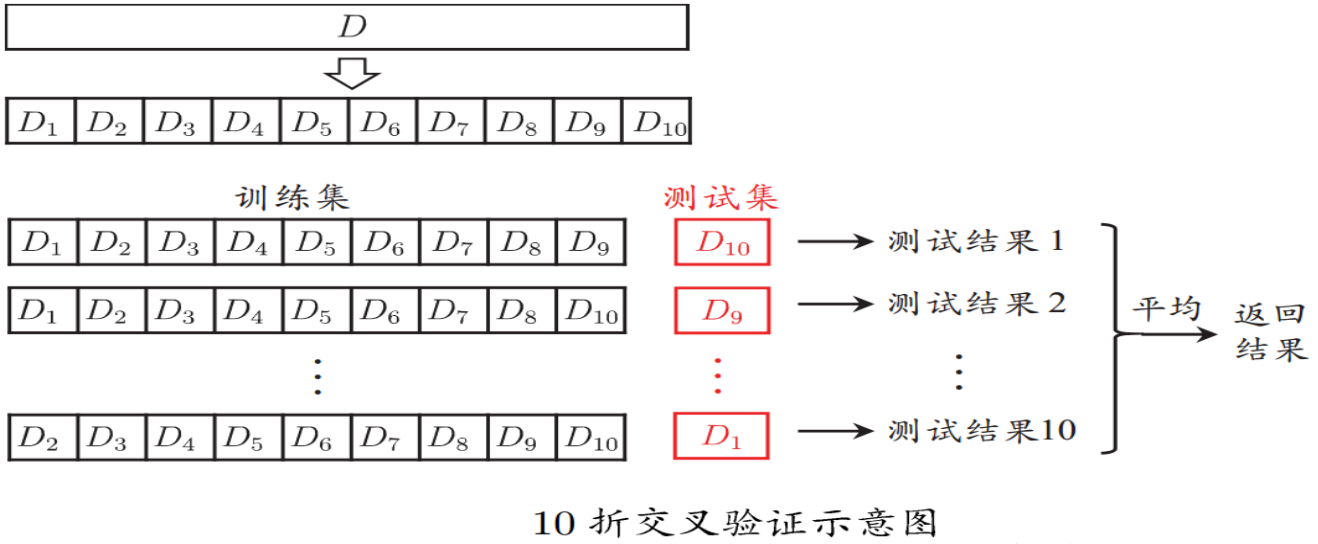
- 假设测试集是从样本真实分布中独立采样获得，将测试集上的“测试误差”作为泛化误差的近似，所以测试集要和训练集中的样本尽量互斥。
- 通常将包含m个样本的数据集拆分成训练集S和测试集T：$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$
- 常用方法
  - 留出法
  - 交叉验证法
  - 自助法

留出法

- 直接将数据集划分为两个互斥集合
- 训练/测试集划分要尽可能保持数据分布的一致性
- 一般若干次随机划分、重复实验取平均值
- 训练/测试样本比例通常为2:1~4:1
- 具体做法（100次留出法）
  - 进行100次随机划分，每次产生一个训练/测试集用于实验评估，100次后产生得到100个结果，留出法返回这100个结果的平均。

交叉验证法

- 将数据集分层采样划分为k个大小相似的互斥子集，每次用k-1个子集的并集作为训练集，余下的子集作为测试集，最终返回k个测试结果的均值，k最常用的取值是10.



10 折交叉验证示意图

- Leave one out cross validation：N个样本中，N-1个作为训练，剩下一个作为测试。

自助法(Bootstrapping)

随机从D中挑选一个样本，将其拷贝放入D'，再挑选，使得该样本在下次采样仍有可能被采到；过程重复，D'作为训练集，$D \setminus D'$作为测试集；

样本在m次不被采集到的概率为，

$$\lim_{m\to\infty}(1-\frac{1}{m})^m \mapsto \frac{1}{e} \approx 0.368$$

亦被称为包外估计（Out-of-bag estimate）

自助法(Bootstrapping)

- 约有1/3的样本没在训练集中出现；
- 从初始数据集中产生多个不同的训练集，对集成学习有很大的好处；
- 自助法在数据集较小、难以有效划分训练/测试集时很有用；
- 由于改变了数据集分布可能引入估计偏差；
- 在数据量足够时，留出法和交叉验证法更常用

性能评价

- 性能度量是衡量模型泛化能力的评价标准，反映了任务需求；使用不同的性能度量往往会导致不同的评判结果
- 在预测任务中，给定样例集$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$评估学习器的性能 f 也即把预测结果$f(\boldsymbol{x})$和真实标记比较.
- 回归采用均方误差 $E(f; D) = \frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i) - y_i)^2$
- 分类任务：错误率和精度
  - 错误率：分错样本占样本总数的比例
  - 精度：分对样本占样本总数的比例

分类器性能评价

列联表如下，1代表正类，0代表负类：

		预测		
		1	0	合计
实际	1	True Postive **TP**	Frue Negative **FN**	Actual Postive(**TP+FN**)
	0	False Postive **FP**	True Negative **TN**	Actual Negative(**FP+TN**)
合计		Predicted Postive (**TP+FP**)	Predicted Negative (**FN+TN**)	**TP+FN+FP+TN**

- 其中，**TP** 表示预测正确的正样本；**TN** 表示预测正确的负样本；**FP** 表示预测错误的负样本；**FN**表示预测错误的正样本。

分类器性能评价

- 评价指标
  - 敏感性(SE)
  - 特异性(SP)
  - 准确率(ACC)
  - 马修相关系数(MCC)
  - 查全率
  - 查准率

$$SE = \frac{TP}{TP + FN}$$

$$SP = \frac{TN}{TN + FP}$$

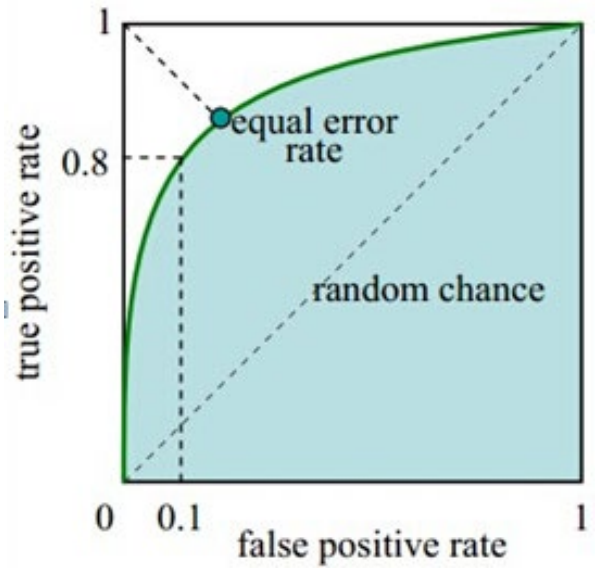
$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN) \times (TP + FP) \times (TN + FP) \times (TN + FN}}$$

$$R = \frac{TP}{TP + FN}$$

$$P = \frac{TP}{TP + FP}$$

- **AUC**
  - **AUC(Area under curve)是一种用来度量分类模型好坏的一个标准.**
  - **ROC(Receiver Operating Characteristic)，是从医疗分析领域引入了一种新的分类模型性能评判方法。平面的横坐标是false positive rate(FPR)，纵坐标是true positive rate(TPR)**



$$TPR = SE = \frac{TP}{TP + FN}$$

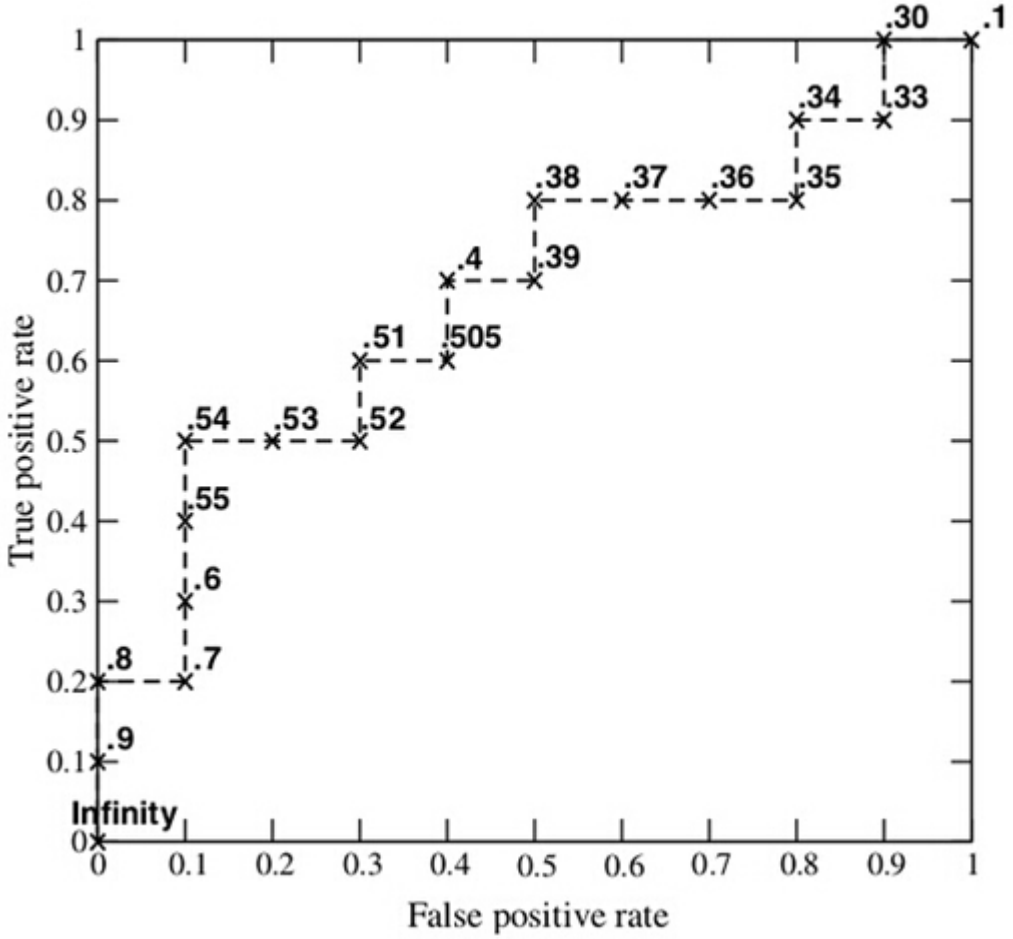
$$FPR = 1 - SP = \frac{FP}{FP + TN}$$

- 假设采用贝叶斯分类器，其给出针对每个实例为正类的阈值(0.6)。
- 对应的就可以算出一组(FPR,TPR),在平面中得到对应坐标点。随着阈值的逐渐减小，越来越多的实例被划分为正类，但是这些正类中同样也掺杂着真正的负实例，即TPR和FPR会同时增大。
- 阈值最大时，对应坐标点为(0,0),阈值最小时，对应坐标点(1,1)。

- 假设已经得出一系列样本被划分为正类的概率，然后按照大小排序。

Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

- 选取不同的阈值，画出如下ROC曲线



模型性能对比

- 关于性能比较：
  - 测试性能并不等于泛化性能
  - 测试性能随着测试集的变化而变化
  - 很多机器学习算法本身有一定的随机性
- 直接选取相应评估方法在相应度量下比大小的方法不可取！
- 假设检验为学习器性能比较提供了重要依据，基于假设检验结果我们可以推断出若在测试集上观察到学习器A比B好，则A的泛化性能是否在统计意义上优于B，以及这个结论的把握有多大。

T检验

- 对多次重复留出法或者交叉验证法进行多次训练/测试时可使用“T检验”——针对单个学习算法的检验

假定得到了k个测试错误率，$\hat{\epsilon}_1, \hat{\epsilon}_2, \ldots, \hat{\epsilon}_k$ 考虑到这k个测试错误率看做泛化错误率 ε_0 的独立采样，则

$$\tau_t = \frac{\sqrt{k}(\mu - \varepsilon_0)}{\sigma}$$

服从自由度为k−1的t分布。其中 $\mu = \frac{1}{k}\sum_{i=1}^{k}\hat{\varepsilon}_i$,

$$\sigma^2 = \frac{1}{k-1}\sum_{i=1}^{k}(\hat{\varepsilon}_i - \mu)^2$$

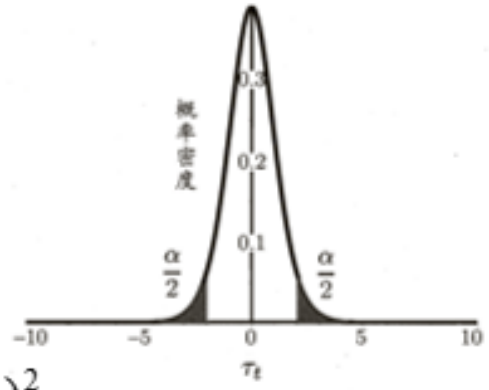


图 2.7 t 分布示意图($k = 10$)

假设 $\epsilon = \epsilon_0$ 对于显著度α，若 $|\mu - \epsilon_0|$ 位于临界范围$[t_{-\alpha/2}, t_{\alpha/2}]$内，则假设不能被拒绝，即可认为泛化错误率$\epsilon = \epsilon_0$，其置信度为$1 - \alpha$。

T检验

- 现实任务中，更多时候需要对不同学习器的性能进行比较——比较两个学习算法性能相同与不同

对两个学习器A和B, 若k折交叉验证得到的测试错误率分别为 $\epsilon_1^A, \ldots, \epsilon_k^A$ $\epsilon_1^B, \ldots, \epsilon_k^B$ 可用k折交叉验证“成对t检验”进行比较检验。若两个学习器的性能相同，则他们使用相同的训练/测试集得到的测试错误率应相同，$\epsilon_i^A = \epsilon_i^B$.

先对每对结果求差，$\Delta_i = \epsilon_i^A - \epsilon_i^B$，若两个学习器性能相同，则差值应该为0，继而用 $\Delta_1, \ldots, \Delta_k$ 来对“学习器A与B性能相同”这个假设做t检验。

若变量 $\tau_t = \left| \frac{\sqrt{k}\mu}{\sigma} \right|$ 小于临界值 $t_{\alpha/2,k-1}$ ，则假设不能被拒绝，即认为两个学习器没有显著差别；

偏差与方差

- 通过实验可以估计学习算法的泛化性能，而“偏差-方差分解”可以用来帮助解释泛化性能。

- “偏差-方差分解”试图对学习算法期望的泛化错误率进行拆解。

  - 偏差：期望输出与真实标记的差别；
  - 方差：反映的是学习器在不同数据集上学习能力的波动；
  - 噪声：标记的误差；

偏差与方差

- 通过实验可以估计学习算法的泛化性能，而“偏差-方差分解”可以用来帮助解释泛化性能。

对测试样本x，令y_D为x在数据集中的标记，y为x的真实标记，$f(\boldsymbol{x};D)$为训练集D上学得模型f在x上的预测输出。以回归任务为例：学习算法的期望预期为：

$$\bar{f}(\boldsymbol{x}) = \mathbb{E}_D[f(\boldsymbol{x};D)]$$

使用样本数目相同的不同训练集产生的方差为

$$var(\boldsymbol{x}) = \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right) - \bar{f}\left(\boldsymbol{x}\right)\right)^2\right]$$

噪声为

$$\varepsilon^2 = \mathbb{E}_D\left[\left(y_D - y\right)^2\right]$$

偏差与方差

- 期望输出与真实标记的差别称为偏差，即

$$bias^2(\boldsymbol{x}) = \left(\bar{f}(\boldsymbol{x}) - y\right)^2$$

- 为便于讨论，假定噪声期望为0，也即

$$\mathbb{E}_D[y_D - y] = 0$$

对泛化误差分解

$$\begin{aligned}
E(f;D) &= \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}) + \bar{f}(\boldsymbol{x}) - y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_D\left[\left(\bar{f}(\boldsymbol{x}) - y_D\right)^2\right] \\
&\quad + \mathbb{E}_D\left[2\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)\left(\bar{f}(\boldsymbol{x}) - y_D\right)\right] \\
&= \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right] + \mathbb{E}_D\left[\left(\bar{f}(\boldsymbol{x}) - y_D\right)^2\right]
\end{aligned}$$

偏差与方差

$$
\begin{aligned}
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y+y-y_D\right)^2\right] \\
&= \mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y\right)^2\right]+\mathbb{E}_D\left[\left(y-y_D\right)^2\right] \\
&\quad +2\mathbb{E}_D\left[\left(\bar{f}\left(\boldsymbol{x}\right)-y\right)\left(y-y_D\right)\right]
\end{aligned}
$$

又由假设中噪声期望为0，可得

$$
E(f;D)=\mathbb{E}_D\left[\left(f\left(\boldsymbol{x};D\right)-\bar{f}\left(\boldsymbol{x}\right)\right)^2\right]+\left(\bar{f}\left(\boldsymbol{x}\right)-y\right)^2+\mathbb{E}_D\left[\left(y_D-y\right)^2\right]
$$

于是有 $E(f;D)=bias^2\left(\boldsymbol{x}\right)+var\left(\boldsymbol{x}\right)+\varepsilon^2$

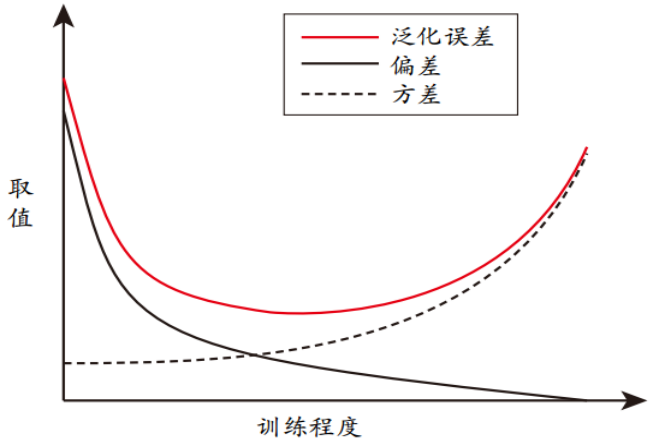
也即泛化误差可分解为方差、偏差与噪声之和。

偏差与方差

- 偏差度量了学习算法期望预测与真实结果的偏离程度；即刻画了学习算法本身的拟合能力；
- 方差度量了同样大小训练集的变动所导致的学习性能的变化；即刻画了数据扰动所造成的影响；
- 噪声表达了在当前任务上任何学习算法所能达到的期望泛化误差的下界；即刻画了学习问题本身的难度。
- 泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度所共同决定的。给定学习任务为了取得好的泛化性能，需要使偏差小(充分拟合数据)而且方差较小(减少数据扰动产生的影响)。

偏差与方差

- 一般来说，偏差与方差是有冲突的，称为偏差-方差窘境
  - 在训练不足时，学习器拟合能力不强，训练数据的扰动不足以使学习器的拟合能力产生显著变化，此时偏差主导泛化错误率；
  - 随着训练程度加深，学习器拟合能力逐渐增强，方差逐渐主导泛化错误率；
  - 训练充足后，学习器的拟合能力非常强，训练数据的轻微扰动都会导致学习器的显著变化，若训练数据自身非全局特性被学到则会发生过拟合。



泛化误差与偏差、方差的关系示意图