一、章节思路

正如标题，本章的核心思想是：**什么是好的模型，怎样选择好的模型**。

2.1 什么是好的模型：**泛化能力**强。

2.2 怎样评估泛化能力：对数据集 *D* 进行划分，产生出训练集 *S* 和测试集 *T*, 用测试集上的**测试误差**作为泛化误差的近似。常见的数据集划分方法有三种：**留出法**(2.2.1)、**交叉验证法**(2.2.2)、**自助法**(2.2.3)。

2.3 用什么作为测试误差：**性能度量**。

性能度量方法

回归任务	1 均方误差
分类任务	1 错误率与精度 2 查准率、查全率与 F1 3 ROC 与 AUC 4 代价敏感错误率与代价曲线

2.4 评估方法和性能度量都有了，能否用算出来的性能度量直接比较学习器的性能：不能。因为：(1)我们希望比较泛化性能，而实际求得的是测试性能，两者未必相同。

(2)测试性能与测试集本身的选择有关，不同的测试集(或即使相同测试集)算出的测试性能有可能不同。

(3)机器学习算法有随机性，同样的参数多次运行的结果可能不同。

那怎样评估性能：**统计假设检验**(2.4)。具体方法包括：

1)单学习器评估：a）**假设检验**(2.4.1)

2)两个学习器比较：a）**交叉验证 t 检验**(2.4.2)(采用同一数据集)

b）**McNemar 检验**(2.4.3)(采用同一数据集)

3)多个学习器比较：a）**Friedman 检验**(多数据集)。若 Friedman 检验被拒绝，则需要使用 **Nemenyi 检验**(2.4.4)进一步区分算法。

2.5 至此，我们已求出了学习器的泛化性能，但我们还希望知道学习器为什么具有这样的性能，即怎样解释学习器的泛化性能：**偏差-方差分解**。

二、算法原理

1 数据集划分：

1.1 留出法：

将数据集 *D* 划分为两个互斥的集合。一个作为训练集，一个作为测试集。

采用若干次随机划分/重复进行实验评估后取平均值作为评估结果。

训练集 *S* 过大，则测试集 *T* 过小，评估结果不够稳定准确。相反，训练集 *S* 过小，会造成训练集 *S* 与样本 *D* 的差别太大，评估保真性(fidelity)过低。

1.2 交叉验证法：

将数据集 *D* 划分为 *k* 个大小相似的互斥子集。用 *k*-1 个子集的并集作为训练集，余下那个子集作为测试集。这样获得了 *k* 组训练/测试集，从而进行 *k* 次训练和测试，最终返回 *k* 个测试结果的均值。由于交叉验证的稳定性和保真性很大程度取决于 *k* 的取值，因此交叉验证法又称为 *k* 折交叉验证(k-fold cross validation)。

与留出法类似，*k* 折交叉验证也要随机划分重复 *p* 次，取结果平均值。如常见的 10 次 10 折交叉验证。

假定数据集 D 中包含 m 个样本，若令 $k=m$，则得到了交叉验证法的一个特例：留一法(Leave One Out, LOO)。显然，留一法不受随机样本划分方式的影响。留一法的缺点是当数据集较大时，计算开销很大。

1.3 自助法

每次随机从数据集 D 中挑选一个样本，将其拷贝放入 D'，然后在将样本放回 D，使得样本下次采样时仍有可能被采到。重复 m 次，就得到了包含 m 个样本的数据集 D'。D'作为训练集，其余样本作为测试集。

自助法在数据集小、难以有效划分训练/测试集时很有用；在初始数据量足够时，留出法和交叉验证法更常用。

2 性能度量：

2.1 均方误差

给定样例集 $D=\{(x_1,y_1),(x_2,y_3)\cdots(x_m,y_m)\}$，其中 y_i 是示例 x_i 的真实标记。评估学习器 f 的性能，就是把预测结果 $f(x)$与真实标记 y 进行比较。

回归任务最常用的是均方误差(mean squared error)

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i)-y_i)^2$$

对于数据分布 D 和概率密度函数 $p(\cdot)$，均方误差可描述为：

$$E(f;\mathcal{D})=\int_{\boldsymbol{x}\sim\mathcal{D}}(f(\boldsymbol{x})-y)^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

2.2 错误率与精度

对分类任务，分类错误率定义为：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i)\neq y_i)$$

其中$\mathrm{II}(\cdot)$为指示函数，在$\cdot$为真和假时分别取值为 1，0。

精度定义为：

$$\begin{aligned}\mathrm{acc}(f;D) &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i)=y_i)\\ &= 1-E(f;D)\ .\end{aligned}$$

对于数据分布 D 和概率密度函数 $p(\cdot)$，错误率和精度可分别描述为：

$$E(f;\mathcal{D})=\int_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{I}(f(\boldsymbol{x})\neq y)\,p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$$

$$\begin{aligned}\mathrm{acc}(f;\mathcal{D}) &= \int_{\boldsymbol{x}\sim\mathcal{D}}\mathbb{I}(f(\boldsymbol{x})=y)\,p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\\ &= 1-E(f;\mathcal{D})\ .\end{aligned}$$

2.3 查准率和查全率

分类混淆矩阵(T=True; F=False; P=Positive; N=Negative)

真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

查准率(precision)P 与查全率(recall)R 分别定义为：

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

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

以查准率为纵轴，查全率为横轴作图，可以得到查准率-查全率曲线，称为 **P-R 曲线**，图称为 **P-R 图**。示意图如下。

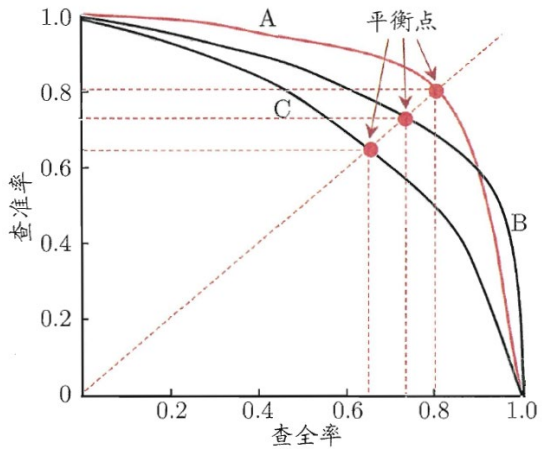


图 2.3 P-R曲线与平衡点示意图

查准率=查全率的取值称为平衡点(Break-Event Point,BEP)。但 BEP 过于简单，更常用的是 F1 度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

F_β 是加权调和平均，β=1 时为 F1；β>1 查全率影响更大；β<1 时查准率影响更大：

$$\frac{1}{F_\beta} = \frac{1}{1+\beta^2} \cdot \left(\frac{1}{P} + \frac{\beta^2}{R} \right)$$

很多时候，我们希望在多个二分类混淆矩阵上综合考察查准率与查全率。一种做法时在各混淆矩阵上分别计算出查准率和查全率，记为$(P_1,R_1),(P_2,R_2),\cdots(Pn,Rn)$，在计算平均值，这样就得到了**宏查准率**(macro-P)、**宏查全率**(macro-R)和**宏 F1**(macro-F1)：

$$\text{macro-}P = \frac{1}{n}\sum_{i=1}^{n} P_i \ ,$$

$$\text{macro-}R = \frac{1}{n}\sum_{i=1}^{n} R_i \ ,$$

$$\text{macro-}F1 = \frac{2 \times \text{macro-}P \times \text{macro-}R}{\text{macro-}P + \text{macro-}R}$$

还可将各混淆矩阵对应元素进行平均，得到微查准率、微查全率和微 F1:

$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}}$$

$$\text{micro-}R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} \ ,$$

$$\text{micro-}F1 = \frac{2 \times \text{micro-}P \times \text{micro-}R}{\text{micro-}P + \text{micro-}R}$$

2.4 ROC 与 AUC

ROC 全称是受试者工作特征(Receiver Operating Characteristic)曲线。ROC 的纵轴是真正例率(True Positive Rate, TPR)，横轴是假正例率(False Positive Rate, FPR)。曲线绘制过程：根据学习器预测结果(结果是一个样本为正例的概率)对样例进行排序，按此顺序逐个把样本作为正例进行预测，计算 TPR 和 FPR。AUC 即为 ROC 曲线下面的面积(就是梯形面积计算公式)。

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

$$\text{FPR} = \frac{FP}{TN + FP}$$

$$\text{AUC} = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i) \cdot (y_i + y_{i+1})$$

进行学习器比较时，若一个学习器的 ROC 曲线被另一个学习器的曲线完全包住，则后者的性能优于前者。怎样理解：如果能包住，说明该曲线的真正例对应的预测概率很高，即曲线 Y 轴方向的上升速率更高，因此面积也更大。

2.5 代价敏感错误率与代价曲线

前面的性能度量计算中默认分类错误的代价相等。如果错误代价不同，则代价敏感错误率为：

$$E(f;D;cost) = \frac{1}{m}\left(\sum_{\boldsymbol{x}_i \in D^+} \mathbb{I}(f(\boldsymbol{x}_i) \neq y_i) \times cost_{01} + \sum_{\boldsymbol{x}_i \in D^-} \mathbb{I}(f(\boldsymbol{x}_i) \neq y_i) \times cost_{10}\right)$$

代价曲线：ROC 曲线上每一点对应了代价平面的一条线段，设 ROC 曲线上点的坐标为(TPR,FPR)，则可计算出 FNR=1-TPR 是假反例率，然后在代价平面上绘制一条从(0,FPR)到(1,FNR)的线段，线段下的面积即表示了该条件下的期望总体代价。将 ROC 曲线上的每个点

转化为代价平面上的一条线段，然后取所有线段的下界，围成的面积即为在所有条件下学习器的期望总体代价。如下图所示。

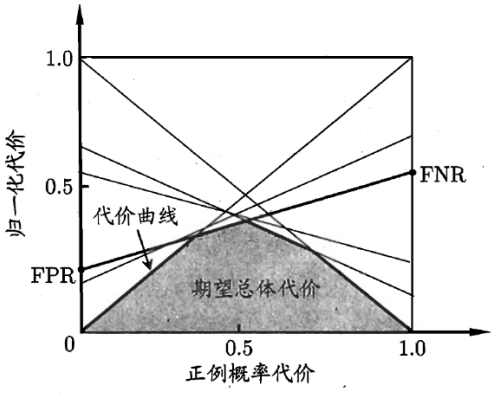


图 2.5　代价曲线与期望总体代价

3 统计假设检验

统计假设检验(hypothesis test)的定义是：　在总体的分布函数完全未知或已知其形式，但不知其参数的情况，为了推断总体的某些未知特性，提出某些关于总体的假设。我们要根据样本对所提出的假设作出是接受还是拒绝的决策。具体到机器学习中就是假设泛化错误率等于一个常数 $_0$，在通过置信度判断是否接受这个假设。

3.1 假设检验

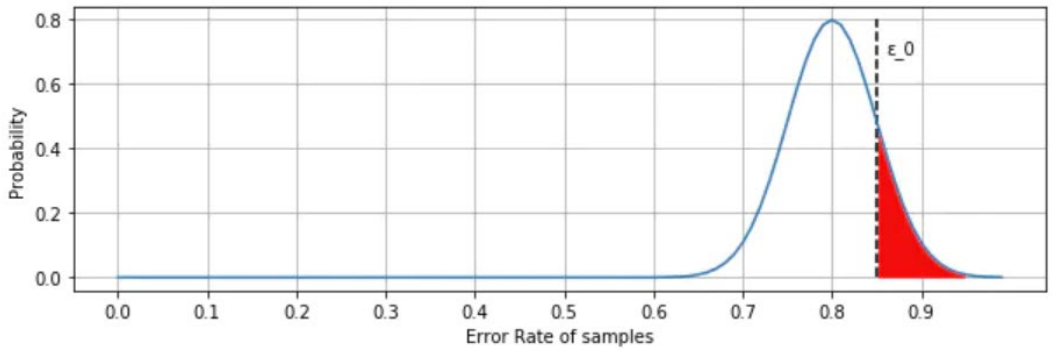
泛化错误率为ϵ的学习器在一个样本上犯错的概率是ϵ；测试错误率为$\hat{\epsilon}$意味着 m 个测试样本中有$\hat{\epsilon}\times m$ 个样本被误分类。泛化错误率为ϵ的学习器被测得测试错误率为$\hat{\epsilon}$的概率：

$$P(\hat{\epsilon};\epsilon)=\binom{m}{\hat{\epsilon}\times m}\epsilon^{\hat{\epsilon}\times m}(1-\epsilon)^{m-\hat{\epsilon}\times m}$$

利用“二项检验”对“$\epsilon<\epsilon_0$”这个假设进行检验，在 1-α的概率内所能观测到的最大错误率如下计算，其中 1- α 反映了结论的**置信度**(confidence)，α称为**显著度**。

$$\bar{\epsilon}=min\epsilon\ \ s.t.\ \sum_{i=\epsilon_0\times m+1}^{m}\binom{m}{i}\epsilon^i(1-\epsilon)^{m-i}<\alpha$$

公式理解：上文说到，$\binom{m}{i}\epsilon^i(1-\epsilon)^{m-i}$就是当泛化错误率为$\epsilon$时，测得测试错误率 $\hat{\epsilon}=\epsilon_0=i/m$ 的概率，在二项分布里就是概率密度。那么，加上求和$\sum_{i=\epsilon_0\times m+1}^{m}$符号，就是$\epsilon\geq\epsilon_0$的概率，即下图中红色部分，跟书中图 2.6 的阴影是一个意思。



我们要检验的是“$\epsilon\ <\epsilon_0$”，也就是$\epsilon\ \geq\epsilon_0$的概率要足够小，要小于α ，满足这样条件的ϵ 很多，根据二项分布特性，在样本数 m 固定的条件下，ϵ 越小，整条曲线向左移，即红色区域面积越小，假设成立的概率就越大。因此取min ϵ 的意义就是使假设成立概率最大的那个值。

很多时候，我们会做多次训练/测试，得到多个测试错误率。此时可采用“t 检验”。

假设得到了 k 个测试错误率，则平均测试错误率μ和方差σ^2为

$$\mu = \frac{1}{k}\sum_{i=1}^{k}\hat{\epsilon}_i \ ,$$

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

考虑到这 k 个测试错误率可看作泛化错误率ϵ_0的独立采样，则变量τ_t(统计量 t 值)服从自由度为 k-1 的 t 分布，如下图所示。考虑双边假设，则α/2 两边的阴影面积为拒绝域，如果τ_t落在拒绝域内，则拒绝假设，否则接受。

$$\tau_t = \frac{\sqrt{k}(\mu - \epsilon_0)}{\sigma}$$

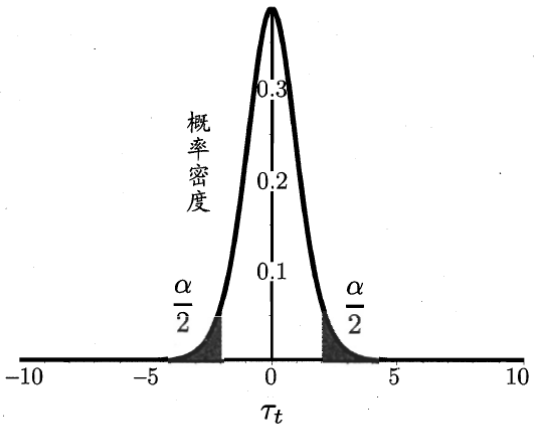


图 2.7 t 分布示意图$(k = 10)$

3.2 交叉验证 t 检验

具体步骤：

1)两个学习器 A 和 B，使用 k 折交叉验证，得到的测试错误率分别为$\epsilon_1^A, \epsilon_2^A, \cdots, \epsilon_k^A$和$\epsilon_1^B, \epsilon_2^B, \cdots, \epsilon_k^B$。

2)先对每对结果求差，得到差值$\Delta_i = \epsilon_i^A - \epsilon_i^A$，若两个学习器性能相同，则差值均值应为 0。

3)计算差值Δ_i的均值μ和方差σ^2，在显著度α下，若变量$\tau_t = \left|\frac{\sqrt{k}\mu}{\sigma}\right|$小于临界值$t_{\alpha/2,k-1}$，则认为两个学习器的性能没有显著差别。

假设检验有效的前提是，测试错误率均为泛化错误率的独立采样。但实际情况测试错误率并不独立，会导致过高的估计假设成立的概率。可采用“5*2 交叉验证”法缓解这个问题。

3.3 McNemar 检验

对于二分类问题，在得到学习器 A 和 B 的测试错误率后，还可获得两学习器分类结果的差别，即两者都正确、错误、一个正确另一个错误的样本数，得到列联表

两学习器分类差别列联表

算法 B	算法 A	
	正确	错误
正确	e_{00}	e_{01}
错误	e_{10}	e_{11}

McNemar 检验考虑变量$\tau_{\chi^2}=\frac{(|e_{01}-e_{10}|-1)^2}{e_{01}+e_{10}}$服从自由度为 1 的$\chi^2$分布。给定显著度$\alpha$，当以上变量值小于临界值$\chi_\alpha^2$时，不能拒绝假设。

3.4 Friedman 检验与 Nemenyi 后续检验

Friedman 检验

(1) 使用留出法或交叉验证法得到每个算法在每个数据集上的测试结果。

(2) 将每个算法根据测试结果进行排序，并赋予序值 1，2，3，•••(如果结果相同，则平分序值)。

(3) 求出每个算法在所有数据集上的平均序值。假定在 N 个数据集上比较 k 个算法。令 r_i 表示第 i 个算法的平均序值，则 r_i 的均值和方差分别为$(k+1)/2$，$(k^2-1)/12N$。变量

$$\tau_{\chi^2}=\frac{k-1}{k}\cdot\frac{12N}{k^2-1}\sum_{i=1}^{k}\left(r_i-\frac{k+1}{2}\right)^2$$

$$=\frac{12N}{k(k+1)}\left(\sum_{i=1}^{k}{r_i}^2-\frac{k(k+1)^2}{4}\right)$$

在 k 和 N 都比较大时，服从自由度为 k-1 的χ^2分布(卡方分布)。上述的“原始 Friedman 检验”过于保守，现在通常使用τ_F变量如下，其中τ_{χ^2}用上式求得。τ_F服从自由度为 k-1 和$(k-1)(N-1)$的 F 分布。下表给出了一些常用的临界值

$$\tau_F=\frac{(N-1)\tau_{\chi^2}}{N(k-1)-\tau_{\chi^2}}$$

表 2.6 F 检验的常用临界值

$\alpha=0.05$									
数据集	算法个数 k								
个数 N	2	3	4	5	6	7	8	9	10
4	10.128	5.143	3.863	3.259	2.901	2.661	2.488	2.355	2.250
5	7.709	4.459	3.490	3.007	2.711	2.508	2.359	2.244	2.153
8	5.591	3.739	3.072	2.714	2.485	2.324	2.203	2.109	2.032
10	5.117	3.555	2.960	2.634	2.422	2.272	2.159	2.070	1.998
15	4.600	3.340	2.827	2.537	2.346	2.209	2.104	2.022	1.955
20	4.381	3.245	2.766	2.492	2.310	2.179	2.079	2.000	1.935
$\alpha=0.1$									
数据集	算法个数 k								
个数 N	2	3	4	5	6	7	8	9	10
4	5.538	3.463	2.813	2.480	2.273	2.130	2.023	1.940	1.874
5	4.545	3.113	2.606	2.333	2.158	2.035	1.943	1.870	1.811
8	3.589	2.726	2.365	2.157	2.019	1.919	1.843	1.782	1.733
10	3.360	2.624	2.299	2.108	1.980	1.886	1.814	1.757	1.710
15	3.102	2.503	2.219	2.048	1.931	1.845	1.779	1.726	1.682
20	2.990	2.448	2.182	2.020	1.909	1.826	1.762	1.711	1.668

若“所有算法的性能相同”这个假设被拒绝，则说明算法的性能显著不同。这是需要进行后续检验(post-hoc test)来进一步区分算法.常用的有 Nemenyi 后续检验。

Nemenyi 检验计算出平均序值差别的临界值域

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

下表给出了α=0.05 和 0.1 时常用的q_α值。若两个算法的平均序值之差超出了临界值域 CD，则以相应的置信度拒绝“两个算法性能相同”这一假设。

表 2.7 Nemenyi 检验中常用的 q_α 值

α	算法个数 k								
	2	3	4	5	6	7	8	9	10
0.05	1.960	2.344	2.569	2.728	2.850	2.949	3.031	3.102	3.164
0.1	1.645	2.052	2.291	2.459	2.589	2.693	2.780	2.855	2.920

4 偏差与方差

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，$f(x;D)$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，学习算法的期望预测为

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

使用样本数相同的不同训练集产生的方差为

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

噪声为

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

期望输出与真实标记的偏差为

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

为方便讨论，假定噪声期望为零，对期望泛化误差进行分解，得到

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

于是得到，泛化误差可分解为偏差、方差与噪声之和。

$$E(f;D) = bias^2(x) + var(x) + \varepsilon^2$$

偏差度量了学习算法的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**度量了同样大小的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**则表达了在当前任务上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。偏差-方差分解说明，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。