**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，能否直接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。

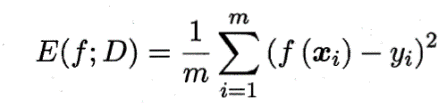
自助法在数据集小、难以有效划分训练/测试集时很有用；在初始数据量足够时，留出法和交叉验证法更常用。

**2性能度量：**

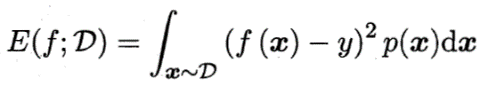
**2.1 均方误差**

给定样例集*D*={(*x*1,*y*1),(*x*2,*y*3)···(*xm*,*ym*)}，其中*yi*是示例*xi*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)

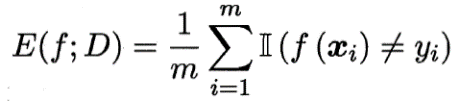


对于数据分布*D*和概率密度函数*p*(·)，均方误差可描述为：



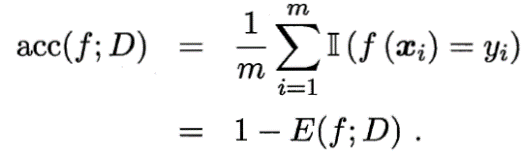
**2.2错误率与精度**

对分类任务，分类错误率定义为：

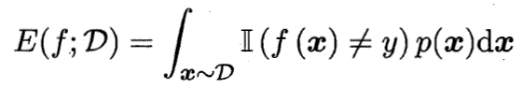


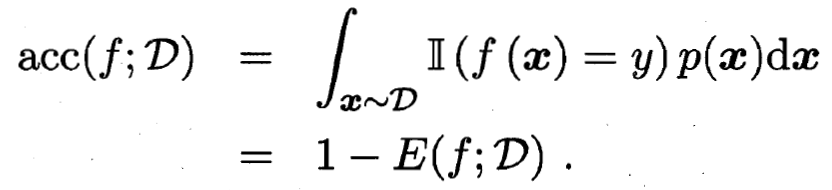
其中Ⅱ(·)为指示函数，在·为真和假时分别取值为1，0。

精度定义为：



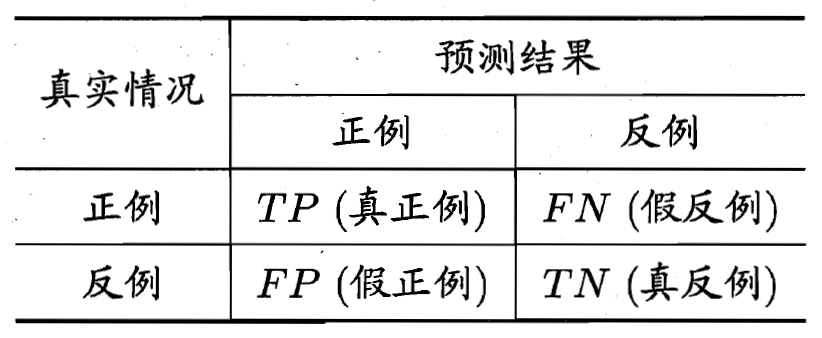
对于数据分布*D*和概率密度函数*p*(·)，错误率和精度可分别描述为：



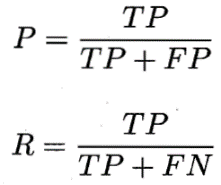


**2.3查准率和查全率**

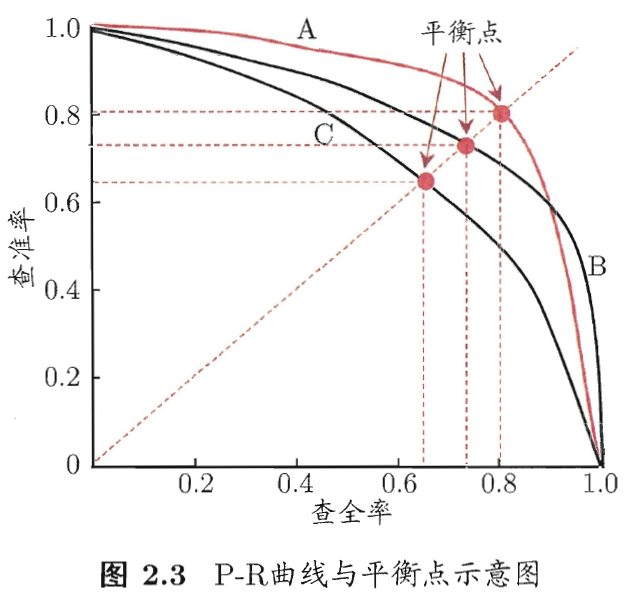
分类混淆矩阵(T=True; F=False; P=Positive; N=Negative)



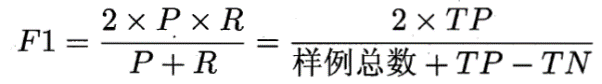
查准率(precision)P与查全率(recall)R分别定义为：



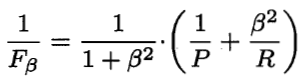
以查准率为纵轴，查全率为横轴作图，可以得到查准率-查全率曲线，称为**P-R曲线，**图称为**P-R图。**示意图如下。



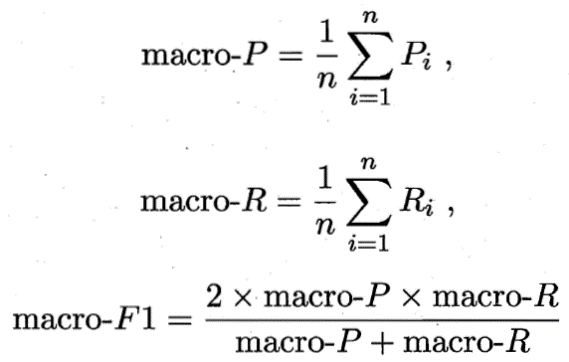
查准率=查全率的取值称为平衡点(Break-Event Point,BEP)。但BEP过于简单，更常用的是F1度量：



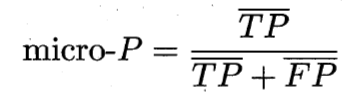
*Fβ*是加权调和平均，*β*=1时为F1；*β*>1查全率影响更大；*β*<1时查准率影响更大：

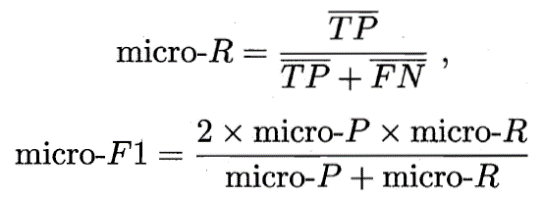


很多时候，我们希望在多个二分类混淆矩阵上综合考察查准率与查全率。一种做法时在各混淆矩阵上分别计算出查准率和查全率，记为(P1,R1),(P2,R2),···(Pn,Rn)，在计算平均值，这样就得到了**宏查准率**(macro-P)、**宏查全率**(macro-R)和**宏F1**(macro-F1)：



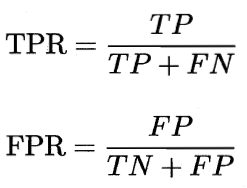
还可将各混淆矩阵对应元素进行平均，得到微查准率、微查全率和微F1:

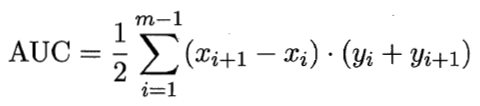




**2.4 ROC与AUC**

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)。

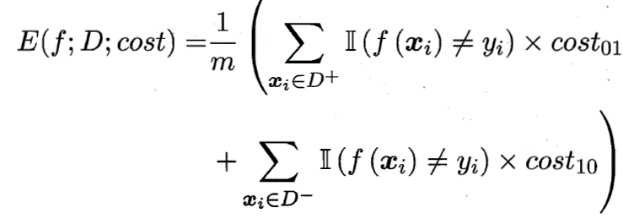




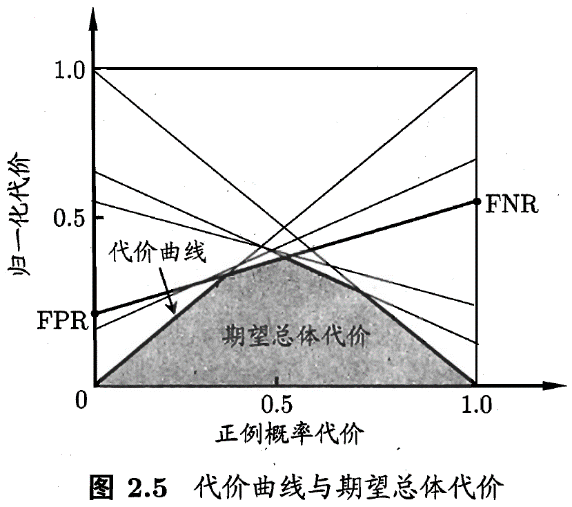
进行学习器比较时，若一个学习器的ROC曲线被另一个学习器的曲线完全包住，则后者的性能优于前者。怎样理解：如果能包住，说明该曲线的真正例对应的预测概率很高，即曲线Y轴方向的上升速率更高，因此面积也更大。

**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：



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。

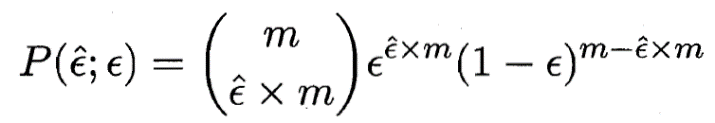


**3统计假设检验**

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。

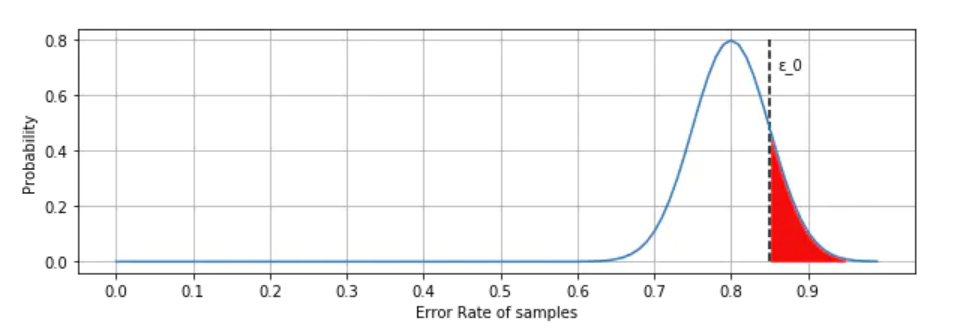
**3.1 假设检验**

泛化错误率为的学习器在一个样本上犯错的概率是；测试错误率为意味着*m*个测试样本中有×*m*个样本被误分类。泛化错误率为的学习器被测得测试错误率为的概率：



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

公式理解：上文说到，就是当泛化错误率为时，测得测试错误率*i*/*m*的概率，在二项分布里就是概率密度。那么，加上求和，就是的概率，即下图中红色部分，跟书中图2.6的阴影是一个意思。

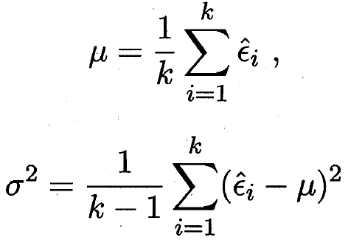


我们要检验的是“”，也就是的概率要足够小，要小于，满足这样条件的很多，根据二项分布特性，在样本数*m*固定的条件下，越大，整条曲线向右移，即红色区域面积越大，因此需要取最小的那个值，这既是的意义。

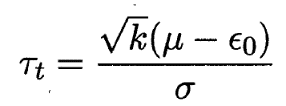
那么，如果得到的测试错误率，即比最小的那个还小，那么发生的概率一定小于，也就是“”发生的概率>1-，也就是发生的概率很高，这时我们就接受假设。

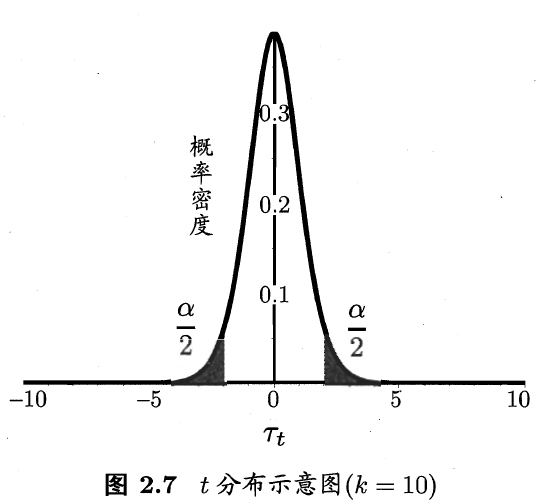
很多时候，我们会做多次训练/测试，得到多个测试错误率。此时可采用“*t*检验”。

假设得到了*k*个测试错误率，则平均测试错误率和方差为



考虑到这*k*个测试错误率可看作泛化错误率的独立采样，则变量(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/2两边的阴影面积为拒绝域，如果落在拒绝域内，则拒绝假设，否则接受。





**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和。

2)先对每对结果求差，得到差值，若两个学习器性能相同，则差值均值应为0。

3)计算差值的均值和方差，在显著度下，若变量小于临界值，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 |

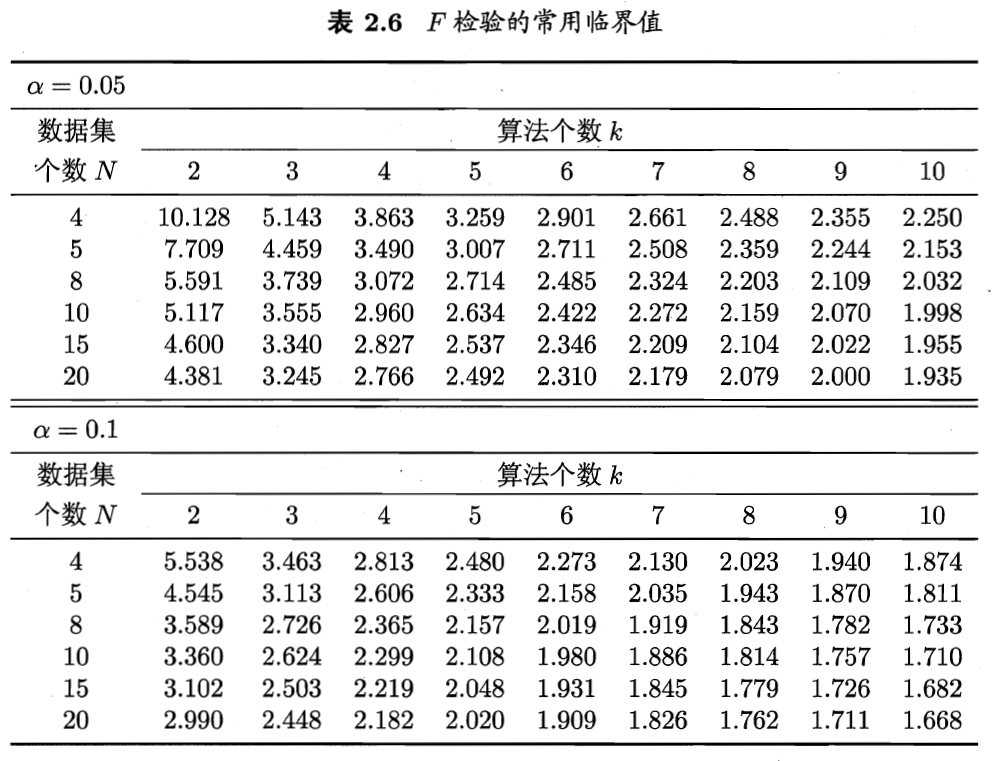
McNemar检验考虑变量服从自由度为1的分布。给定显著度，当以上变量值小于临界值时，不能拒绝假设。

**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*ri*表示第*i*个算法的平均序值，则*ri*的均值和方差分别为(*k*+1)/2，(*k*2-1)/12*N*。变量

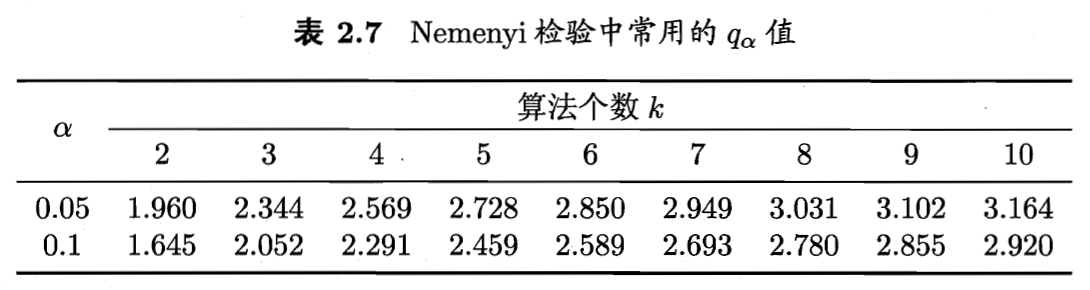
在*k*和*N*都比较大时，服从自由度为*k*-1的分布(卡方分布)。上述的“原始Friedman检验”过于保守，现在通常使用变量如下，其中用上式求得。服从自由度为*k*-1和(*k*-1)(*N*-1)的F分布。下表给出了一些常用的临界值



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

下表给出了=0.05和0.1时常用的值。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。



**4 偏差鱼方差**

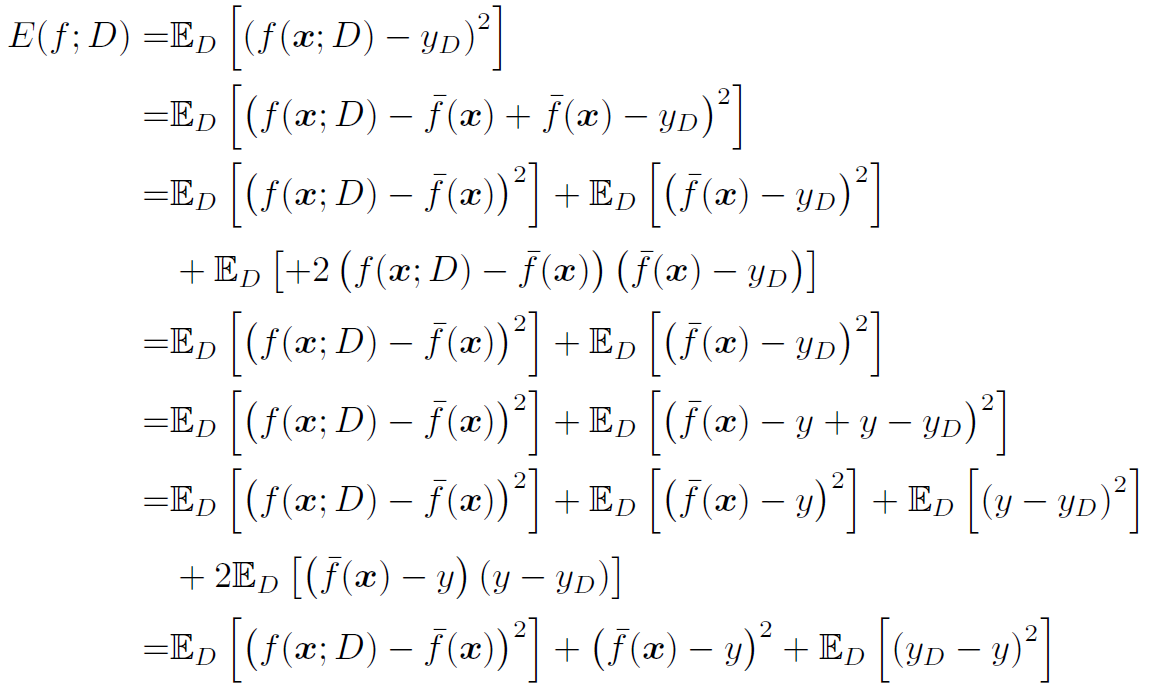
对测试样本*x*，令*yD*为*x*在数据集中的标记，*y*为*x*的真实标记，*f*(*x*;*D*)为训练集*D*上学得模型*f*在*x*上的预测输出。以回归任务为例，学习算法的期望预测为

使用样本数相同的不同训练集产生的方差为

噪声为

期望输出与真实标记的偏差为

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



于是得到，泛化误差可分解为偏差、方差与噪声之和。

+

**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。