# 2.模型的评估与选择

## 2.1.经验误差与过拟合

**训练误差：**学习器在训练集上的误差。

**泛化误差：**学习器在新样本上的误差。

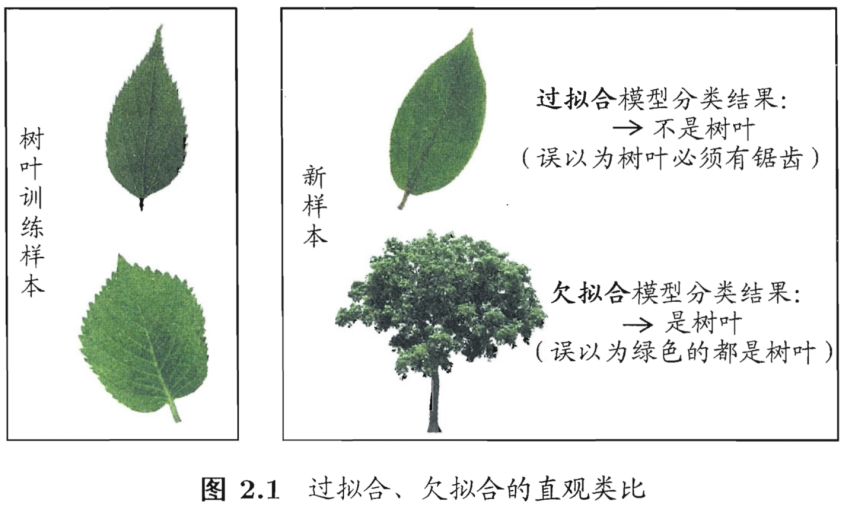
我们希望得到泛化误差小的学习器，但能做的是让训练误差尽量小。

为了达到这个目的，应该从训练样本中尽可能学习出适用于所有潜在样本的普遍规律，这样才可遇到新样本正确判断。

**过拟合：**当学习器把训练样本学得太好了的时候，很可能已经把训练样本的一些特点，当成了所有潜在样本都具有的一般性质，导致泛化能力下降。

**欠拟合：**对训练样本的一般性质尚未学好。

可以得知：在过拟合问题中，训练误差十分小，但测试误差教大；在欠拟合问题中，训练误差和测试误差都比较大。目前，欠拟合问题比较容易克服，例如增加迭代次数等，但过拟合问题还没有十分好的解决方案，过拟合是机器学习面临的关键障碍。



## 2.2.评估方法

在现实任务中，我们往往有多种算法可供选择，那么我们应该选择哪一个算法才是最适合的呢？如上所述，我们希望得到的是泛化误差小的学习器，理想的解决方案是对模型的泛化误差进行评估，然后选择泛化误差最小的那个学习器。但是，泛化误差指的是模型在所有新样本上的适用能力，我们无法直接获得泛化误差。

因此，通常我们采用一个“测试集”来测试学习器对新样本的判别能力，然后以“测试集”上的“测试误差”作为“泛化误差”的近似。显然：我们选取的测试集应尽可能与训练集互斥，因此我们需要对初始数据集进行有效划分，划分出互斥的“训练集”和“测试集”。下面介绍几种常用的划分方法：

**1.留出法**

“留出法”（hold-out）直接将数据集D划分为两个互斥的集合，一个作为训练集S，一个作为测试集T，满足D=S∪T且S∩T=∅，常见的划分为：大约2/3-4/5的样本用作训练，剩下的用作测试。

需要注意的是：训练/测试集的划分要尽可能保持数据分布的一致性，以避免由于分布的差异引入额外的偏差，常见的做法是采取分层抽样。

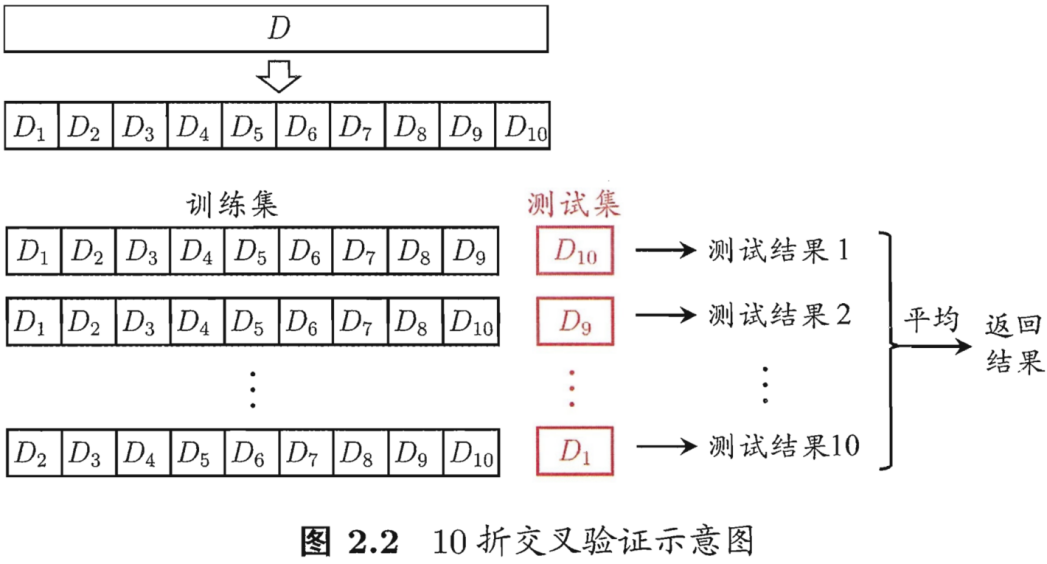
同时，由于划分的随机性，单次的留出法结果往往不够稳定，一般要采用若干次随机划分，重复进行实验评估后取均值作为评估结果。

**2.交叉验证法**

“交叉验证法”（cross validation）先将数据集D划分为k个大小相同的互斥子集，满足，同样地尽可能保持数据分布的一致性，即采用分层抽样的方法获得这些子集。

交叉验证法的思想是：每次用k-1个子集的并集作为训练集，余下的那个子集作为测试集，这样就有K种训练集/测试集划分的情况，从而可进行k次训练和测试，最终返回k次测试结果的均值。

交叉验证法也称“k折交叉验证”，k常用的取值有5、10、20等，下图给出了10折交叉验证的示意图。



与留出法类似，将数据集D划分为K个子集的过程具有随机性，因此K折交叉验证通常也要重复p次，称为p次k折交叉验证，

例如10次10折交叉验证，即进行了100次训练/测试。

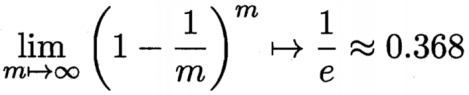
特殊地当划分的k个子集的每个子集中只有一个样本时，称为“留一法”，显然，留一法的评估结果比较准确，但对计算机的消耗也是巨大的。

**3.自助法**

我们希望评估的是用整个D训练出的模型。但在留出法和交叉验证法中，由于保留了一部分样本用于测试，因此实际评估的模型所使用的训练集比D小，这必然会引入一些因训练样本规模不同而导致的估计偏差。留一法受训练样本规模变化的影响较小，但计算复杂度又太高了。“自助法”正是解决了这样的问题。

“自助法”（bootstrapping）：以自助采样/可重复采样/有放回采样为基础。给定D，每次随机挑选一个样本放入D\*，将该样本放回D。重复执行m次，则得到D\*：含m个样本的集。此即自助采样的结果。

可以得知在m次采样中，样本始终不被采到的概率取极限为：



即D中大约有36.8%的样本不出现在D\*中。将D\*作为训练集，D-D\*作为测试集。这样的测试结果，亦称“包外估计”（out-of-bag estimate）。

适用场景：D较小，难以有效划分。且可从D中产生不同的训练集，对集成学习有好处。

缺点：改变了D的分布，引入估计偏差。在数据足够时，前两种更常用。

**4.参数调节**

大多数学习算法都有些参数(parameter) 需要设定，参数配置不同，学得模型的性能往往有显著差别，这就是通常所说的"参数调节"或简称"调参" (parameter tuning)。

学习算法的很多参数是在实数范围内取值，因此，对每种参数取值都训练出模型来是不可行的。

常用grid search（网格搜索法）：对每个参数设定范围和步长，从候选值中遍历。即便如此，调参仍很困难，工程量仍很大。以至于不少任务中，调参影响模型的性能。

ML中涉及两类参数：

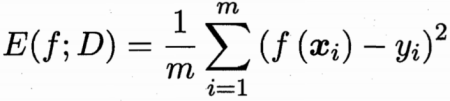
算法参数：亦称超参数，人为指定候选值，数目较少。

模型参数：数目可能很多，通过学习过程中产生候选值。

## 2.3.性能度量

性能度量（performance measure）是衡量模型泛化能力的评价标准，在对比不同模型的能力时，使用不同的性能度量往往会导致不同的评判结果。

回归任务中，最常用的性能度量是“均方误差”（mean squared error）,很多的经典算法都是采用了MSE作为评价函数。



下面介绍分类任务中常用的性能度量

**1.错误率与精度**

错误率：分类错误的样本占样本总数的比例。

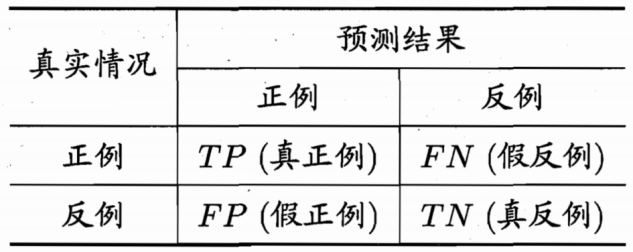
精度：分类正确的样本占样本总数的比例。

**2.查准率，查全率，F1**

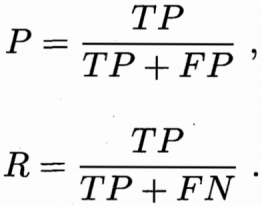
错误率和精度虽然常用，但不能满足所有的需求

例如：在推荐系统中，我们只关心推送给用户的内容用户是否感兴趣（即查准率），或者说所有用户感兴趣的内容我们推送出来了多少（即查全率）。因此，使用查准/查全率更适合描述这类问题。

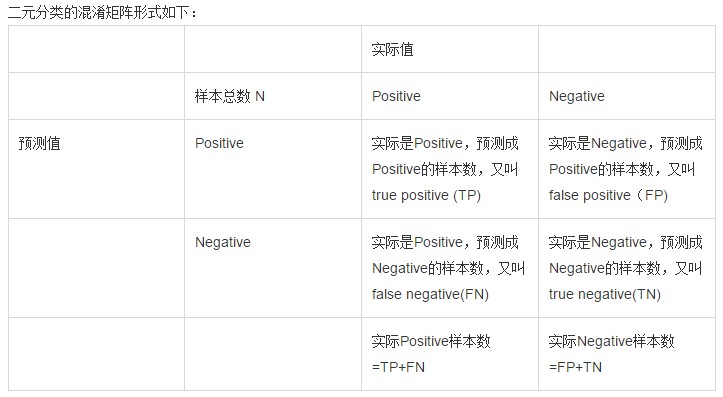
对于二分类问题，分类结果混淆矩阵与查准/查全率定义如下：



查准率P与查全率R分别定义为：



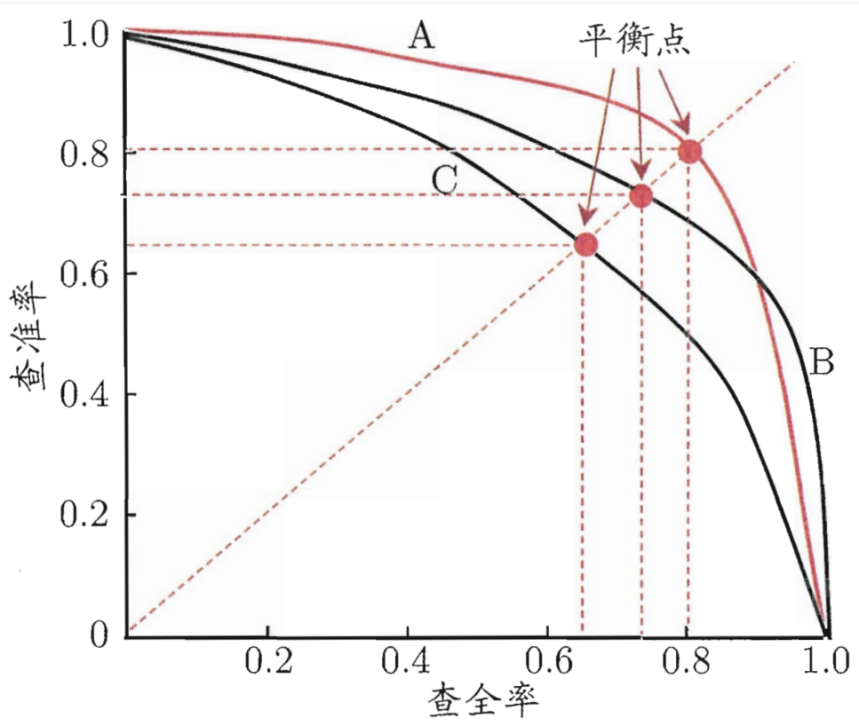
初次接触时，FN与FP很难正确的理解，为方便理解，进行详细描述：



正如天下没有免费的午餐，查准率和查全率是一对矛盾的度量。

例如我们想让推送的内容尽可能用户全都感兴趣，那只能推送我们把握高的内容，这样就漏掉了一些用户感兴趣的内容，查全率就低了；如果想让用户感兴趣的内容都被推送，那只有将所有内容都推送上，宁可错杀一千，不可放过一个，这样查准率就很低了。

“P-R曲线”正是描述查准/查全率变化的曲线，P-R曲线定义如下：根据学习器的预测结果（一般为一个实值或概率）对测试样本进行排序，将最可能是“正例”的样本排在前面，最不可能是“正例”的排在后面，按此顺序逐个把样本作为“正例”进行预测，每次计算出当前的P值和R值，如下图所示：



P-R曲线如何评估呢？

若一个学习器A的P-R曲线被另一个学习器B的P-R曲线完全包住，则称：B的性能优于A。若A和B的曲线发生了交叉，则谁的曲线下的面积大，谁的性能更优。

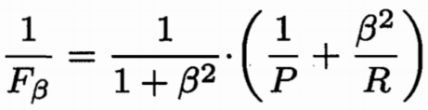
但一般来说，曲线下的面积是很难进行估算的，所以衍生出了“平衡点”（Break-Event Point，简称BEP），即当P=R时的取值，平衡点的取值越高，性能更优。

P和R指标有时会出现矛盾的情况，这样就需要综合考虑他们，最常见的方法就是F-Measure，又称F-Score。F-Measure是P和R的加权调和平均，即：



不同应用对P-R的重视程度不同。商品推荐中查准率高点好；逃犯查询中查全率高点好。

F1度量的推广形式Fβ可以对二者重要性调和：



其中β度量了查全率对查准率的相对重要性，β=1时退化为标准的F1；β>1时查全率有更大影响；β<1时查准率有更大影响.

## 2.4.偏差与方差

“偏差-方差分解”是解释学习器泛化性能的重要工具。

在学习算法中，偏差指的是预测的期望值与真实值的偏差，方差则是每一次预测值与预测值得期望之间的差均方。实际上，偏差体现了学习器预测的准确度，而方差体现了学习器预测的稳定性。

通常，在欠拟合时，偏差主导泛化误差，而训练到一定程度后，偏差越来越小，方差主导了泛化误差。