第二章：模型评估与选择

主讲：连德富 特任教授 | 博士生导师

邮箱：liandefu@ustc.edu.cn

手机：13739227137

主页：http://staff.ustc.edu.cn/~liandefu

模型评估与选择

- 模型评估
  - 给定一个数据集，如何估计一个模型的“泛化” 能力？

- 模型选择
  - 给定一个数据集，如何根据“泛化” 能力，选出最好的模型或选出最好的参数配置

模型评估—经验误差与过拟合

误差：样本真实输出与预测输出之间的差异，可以是错误率

训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是	否
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是	否
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否	是
16	浅白	蜷缩	浊响	模糊	平坦	硬滑	否	否
17	青绿	蜷缩	沉闷	稍糊	稍凹	硬滑	否	否

训练误差
经验误差

测试集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是	否
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否	是
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否	是
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否	否

测试误差

模型评估—经验误差与过拟合

训练集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
1	青绿	蜷缩	浊响	清晰	凹陷	硬滑	是	是
2	乌黑	蜷缩	沉闷	清晰	凹陷	硬滑	是	否
3	乌黑	蜷缩	浊响	清晰	凹陷	硬滑	是	是
6	青绿	稍蜷	浊响	清晰	稍凹	软粘	是	否
7	乌黑	稍蜷	浊响	稍糊	稍凹	软粘	是	是
10	青绿	硬挺	清脆	清晰	平坦	软粘	否	否
14	浅白	稍蜷	沉闷	稍糊	凹陷	硬滑	否	否
15	乌黑	稍蜷	浊响	清晰	稍凹	软粘	否	是

训练误差
经验误差

泛化误差：除训练集外的所有样本上的误差

测试集

编号	色泽	根蒂	敲声	纹理	脐部	触感	好瓜	预测
4	青绿	蜷缩	沉闷	清晰	凹陷	硬滑	是	是
5	浅白	蜷缩	浊响	清晰	凹陷	硬滑	是	是
8	乌黑	稍蜷	浊响	清晰	稍凹	硬滑	是	否
9	乌黑	稍蜷	沉闷	稍糊	稍凹	硬滑	否	是
11	浅白	硬挺	清脆	模糊	平坦	硬滑	否	否
12	浅白	蜷缩	浊响	模糊	平坦	软粘	否	是
13	青绿	稍蜷	浊响	稍糊	凹陷	硬滑	否	否

测试误差

模型评估—经验误差与过拟合

- 事先可能不知道新样本的特征，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

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

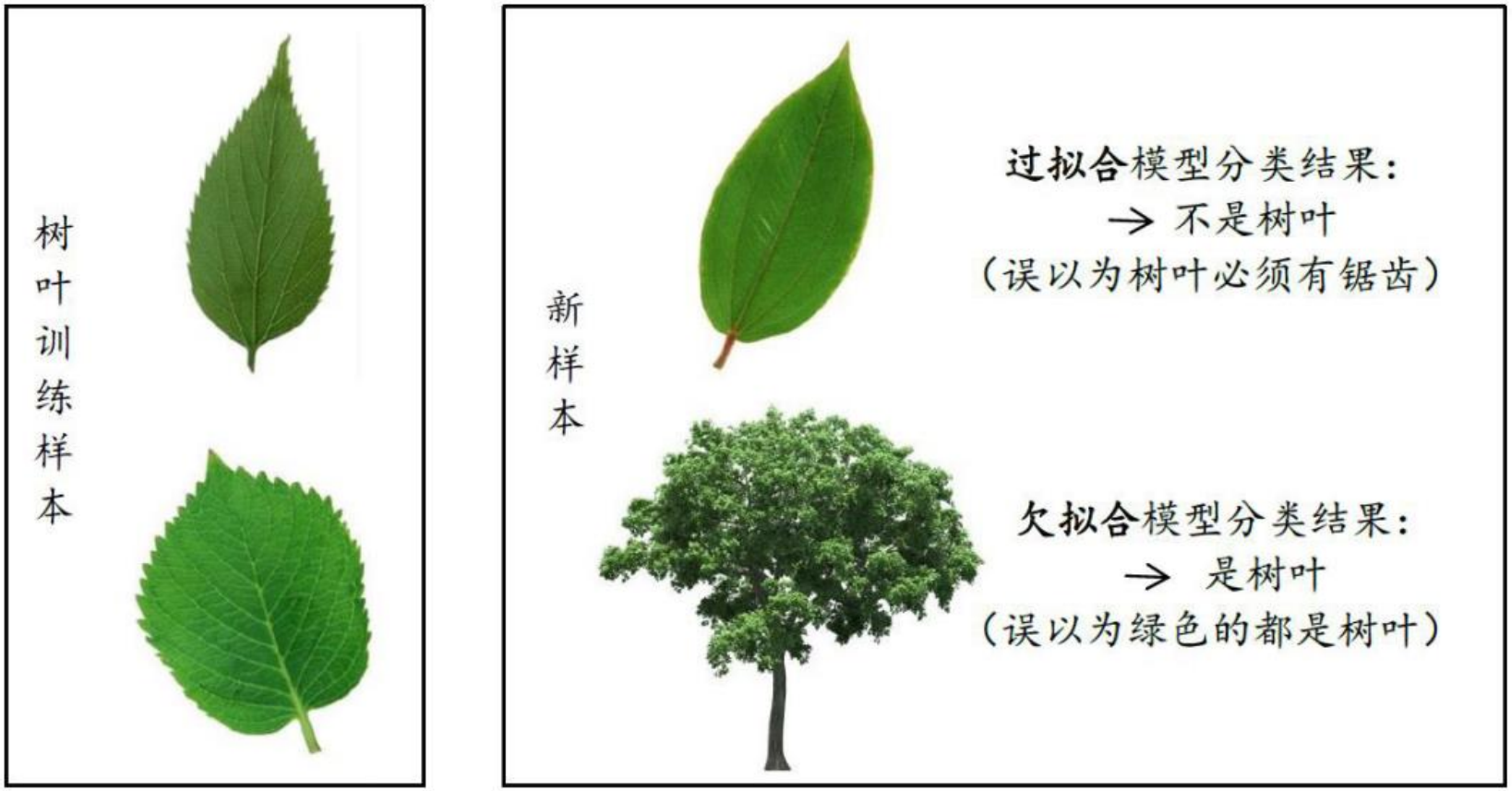
解决办法：优化目标加正则项；early stop

欠拟合

对训练样本的一般性质尚未学好

解决办法：决策树:拓展分支；神经网络：增加训练轮数

模型评估—过拟合与欠拟合



过拟合、欠拟合的直观类比

模型评估—评估方法

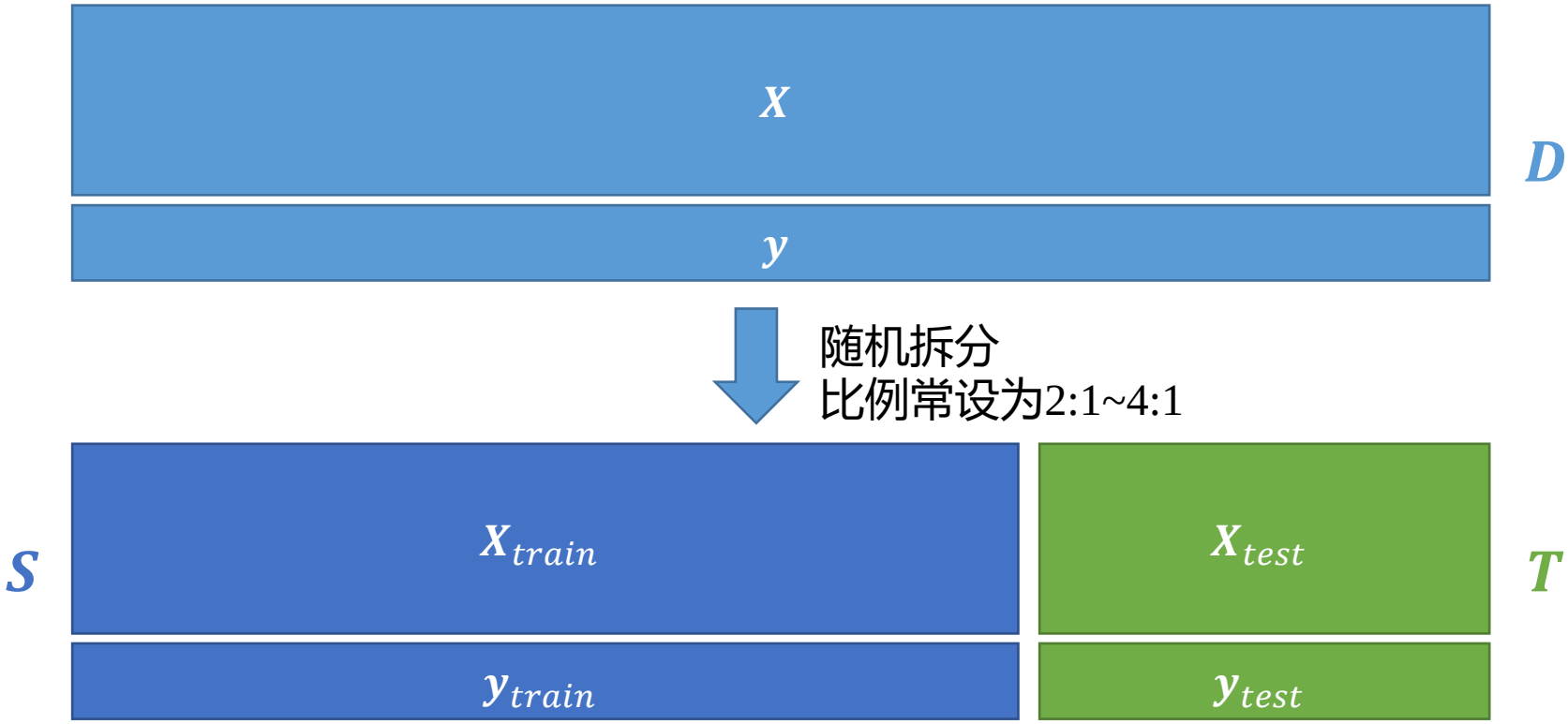
- 需要一个测试集来测试学习器对新样本的判别能力

- 假设测试集是从样本真实分布中独立采样获得，以测试集上的测试误差作为泛化误差的近似

测试集要和训练集中的样本尽量互斥，即测试样本尽量不在训练集中出现、未在训练集中使用过

评估方法—留出法 (hold-out)

- 假设数据集合为$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$



满足$D = S \cup T$且$S \cap T = \emptyset$

评估方法—留出法 (hold-out)

2:1随机拆分

训练集S

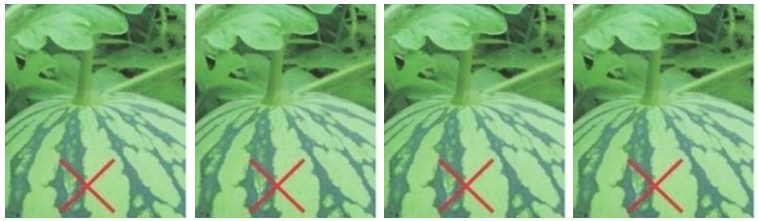
正负样本在训练集、测试集中的分布与数据集的不一致，**误差估计在可能会产生偏差**

测试集T

评估方法—留出法 (hold-out)

2:1随机拆分

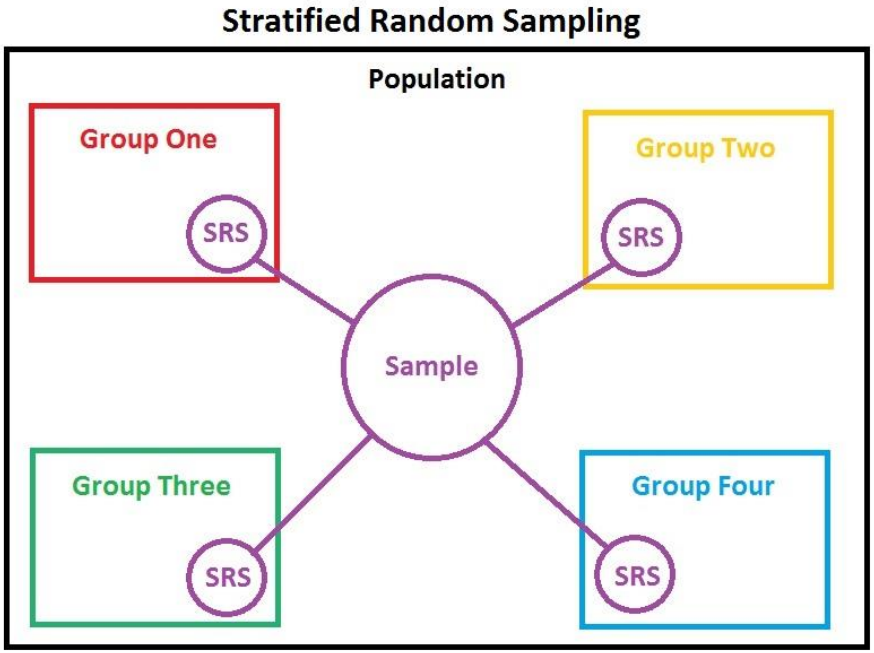
2:1随机拆分

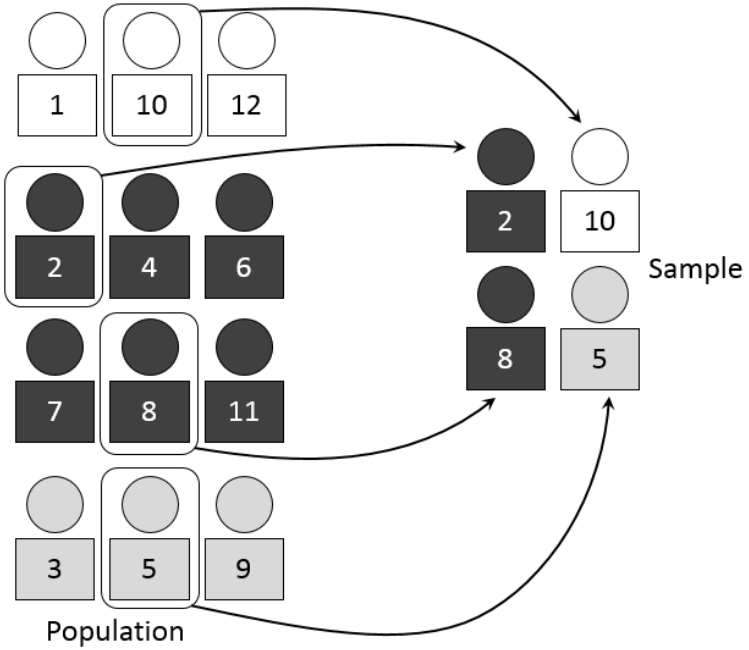


训练集S

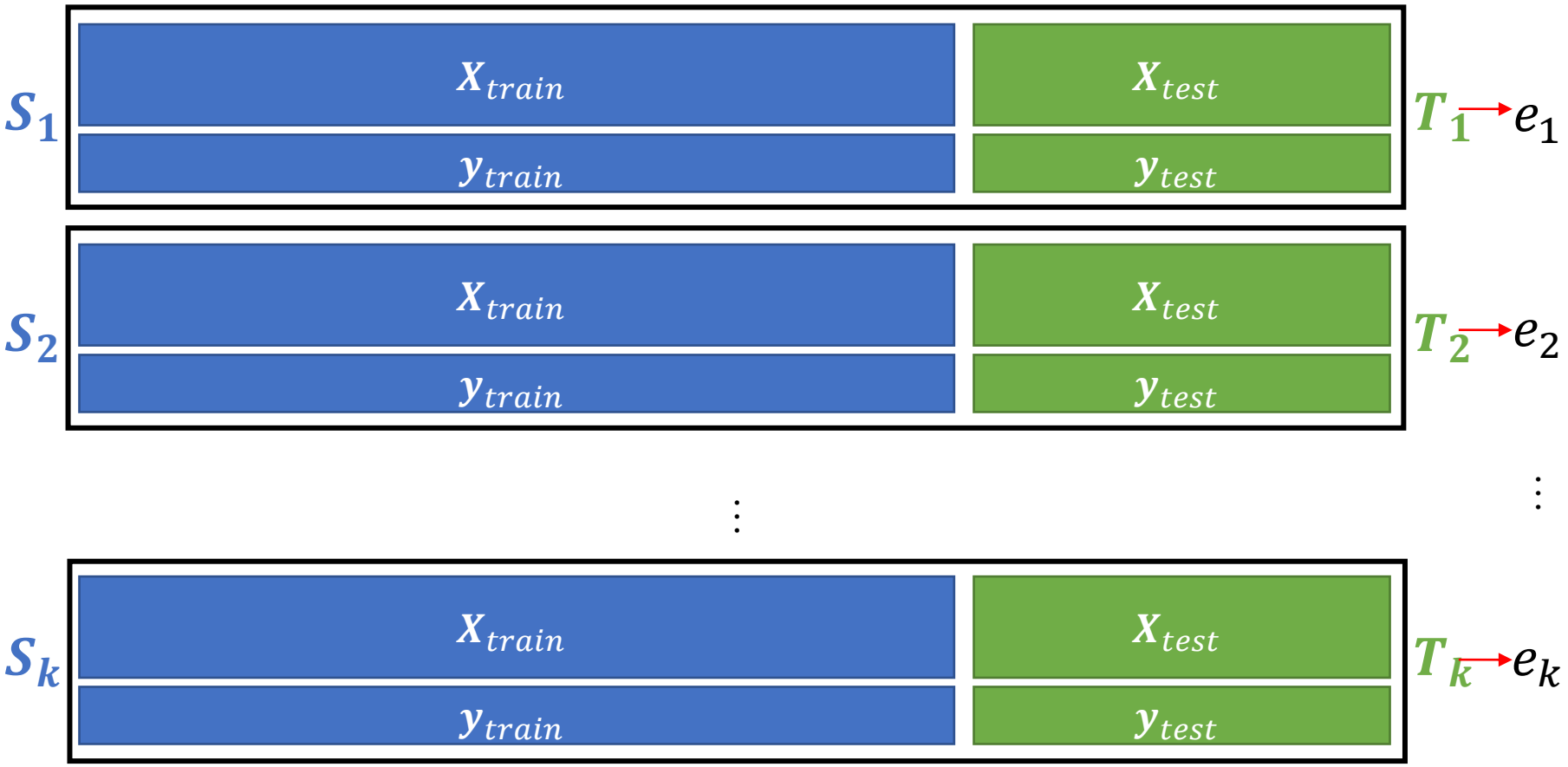
测试集T

评估方法—留出法 (hold-out)





评估方法—留出法 (hold-out)

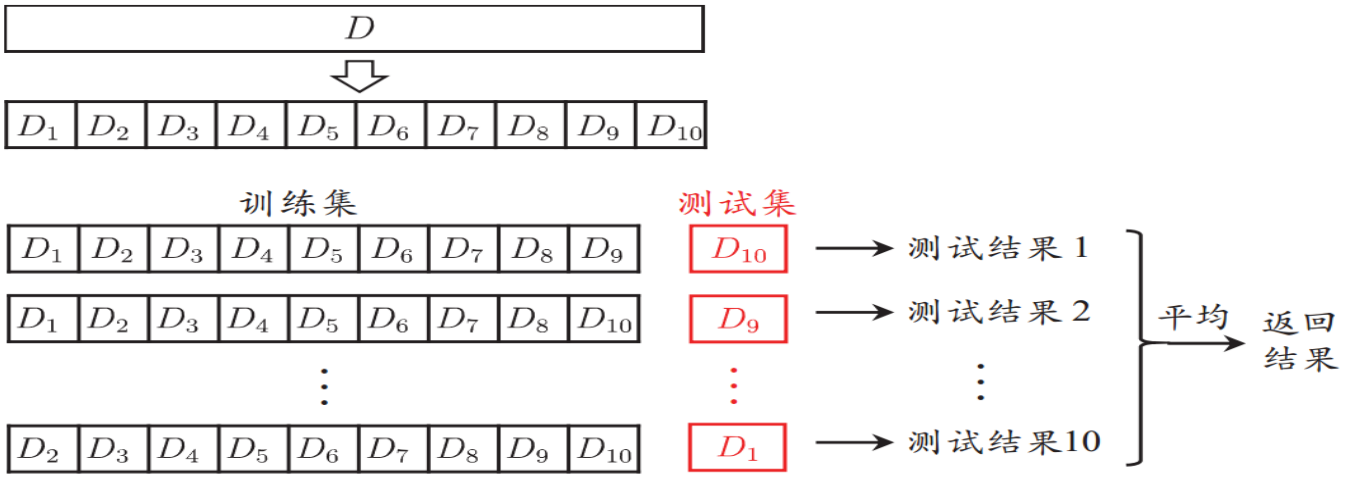


$$e = \frac{1}{K}(e_1 + e_2 + \cdots + e_k)$$

评估方法—交叉验证法

1. 将数据集分层采样划分为K个大小相似的互斥子集
2. 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
3. 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

评估方法—交叉验证法

- 与留出法类似，将数据集D划分为k个子集同样存在多种划分方式
- 为了减小因样本划分不同而引入的差别，k折交叉验证通常随机使用不同的划分重复p次，最终的评估结果是这p次k折交叉验证结果的均值
- 例如常见的“10次10折交叉验证”

评估方法—交叉验证法

- 当$k = m$，（每个样本一个集合），得到留一法

不受随机样本划分方式的影响
结果往往比较准确
当数据集比较大时，计算开销难以忍受

评估方法—自助法

- 希望评估D训练出的模型，但是实际评估模型使用了更小数据集

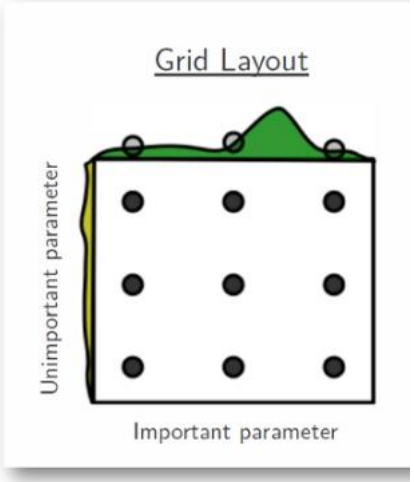
- 以自助采样法为基础，对数据集 D 有放回采样 m 次得到训练集 D'，$D \setminus D'$ 用做测试集。

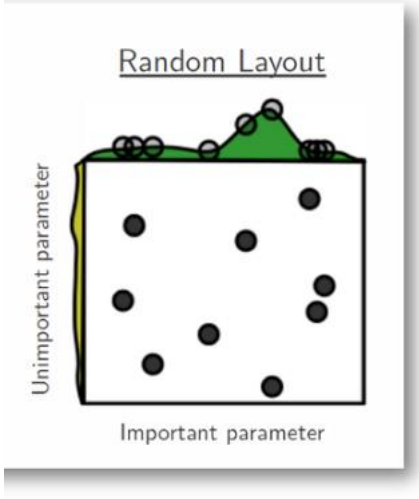
样本在m次采样中始终不被采样到的概率 $= \lim_{m\to\infty}\left(1-\frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$

- 实际模型与预期模型都使用 m 个训练样本
- 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。

模型评估—调参和最终模型

- 大多数学习算法都有些参数需要设定，不同参数设置，导致学得模型的性能有显著差别
- 模型选择，包括学习算法选择和参数配置的设定，后者称为调参
- 调参的一般过程：
  - 将训练集划分为训练集和验证集
  - 通过网格法或随机法进行参数搜索，计算出验证集上的误差
  - 选出最佳的参数配置，在训练集上重新训练





模型评估—性能度量

- 性能度量是衡量模型泛化能力的评价标准
- 反映任务需求，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$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_m, y_m)\}$，评估学习器的性能$f$也即把预测结果$f(x)$和真实标记比较

回归任务最常用的性能度量是“均方误差”

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

假设知道数据的分布，那么均方误差表达为

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

模型评估—性能度量

- 对于分类任务,错误率和精度是最常用的两种性能度量:

错误率：分错样本占样本总数的比例

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

精度：分对样本占样本总数的比率

$$acc(f,D)=\frac{1}{m}\sum_{i}\mathbb{I}(f(\boldsymbol{x}_i)=y_i)=1-E(f,D)$$

模型评估—性能度量

- 错误率和精度虽然常用，但不能满足所有任务需求
  - 比如，挑出的西瓜中有多少比例是好瓜，有多少比例的好瓜被挑选出来
  - 信息检索等场景经常需要衡量正例被预测出来的比率或者预测出来的正例中正确的比率

- 查准率和查全率比错误率和精度更适合

混淆矩阵 (confusion matrix)

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)

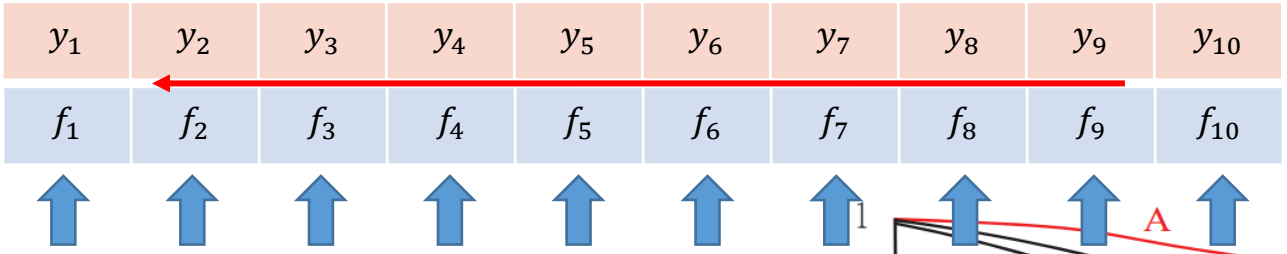
查全率 (Recall) $R = \frac{TP}{TP+FN}$

查准率(Precision) $P = \frac{TP}{TP+FP}$

模型评估—性能度量

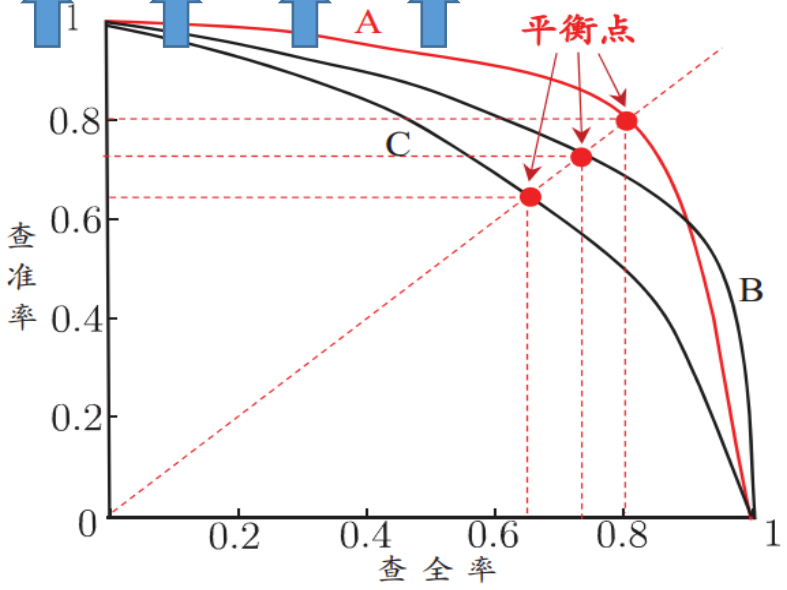
- 查准率和查全率是一堆矛盾的度量
  - 查准率高时，查全率低；查全率高时，查准率低

如何权衡这两个指标呢？



根据学习器的预测结果按正例可能性大小对样例进行排序，并逐个把样本作为正例进行预测

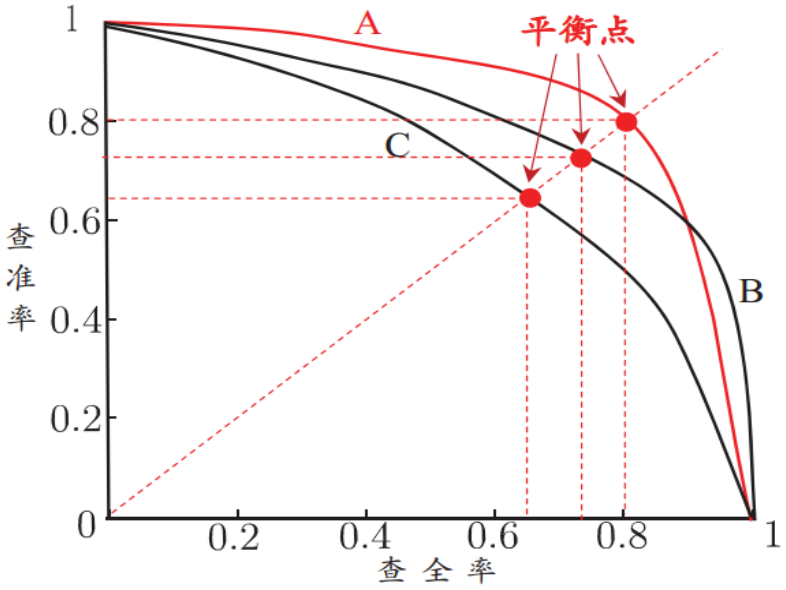
得到查准率-查全率曲线，简称“P-R曲线”



P-R曲线与平衡点示意图

模型评估—性能度量

- 如果一个学习器的P-R曲线被另一个学习器的曲线完全包住，那么后者性能更优
- 如果发生了交叉，则难以判断孰优孰劣
- 可以估算P-R曲线下的面积，但是估算比较困难
- 通过平衡点来权衡这两者指标



P-R曲线与平衡点示意图

平衡点是曲线上“查准率=查全率”时的取值，可用来用于度量P-R曲线有交叉的分类器性能高低

模型评估—性能度量

- 比P-R曲线平衡点更用常用的是F1度量：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{1}{\frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)}$$

- 比F1更一般的形式 F_β ，

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

$\beta = 1$：标准的F1
$\beta > 1$：偏重查全率
$\beta < 1$：偏重查准率

模型评估—性能度量

- 如果有多个二分类混淆矩阵
  - 多次训练/测试、多个数据集训练/测试、多分类中每两两类别的组合

先在各个混淆矩阵上分别计算出查准率和查全率，记为$(P_1, R_1), (P_2, R_2), \cdots, (P_n, R_n)$。再计算均值，得到宏查准率（macro-P）、宏查全率（macro-R）和相应的宏F1（macro-F1）。

$$\text{macro-P} = \frac{1}{n}\sum_i P_i$$

$$\text{micro-R} = \frac{1}{n}\sum_i R_i$$

先在各个混淆矩阵对应元素平均，得到TP、FP、TN、FN的平均值，在基于平均值计算微查准率（micro-P）、微查全率（micro-R）和相应的微F1（micro-F1）

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

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

模型评估—性能度量

- 类似P-R曲线，根据学习器的预测结果对样例排序，并逐个作为正例进行预测，以“假正例率”为横轴，“真正例率”为纵轴可得到ROC曲线，全称“受试者工作特征（Receiver Operating Characteristics）”.

混淆矩阵 (confusion matrix)

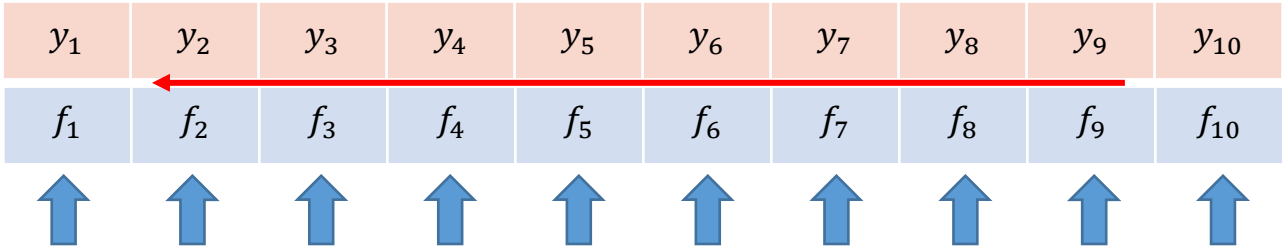
真实情况	预测结果	
	正例	反例
正例	TP (真正例)	FN (假反例)
反例	FP (假正例)	TN (真反例)

$$TPR = \frac{TP}{TP + FN}$$ 真正例率

$$FPR = \frac{FP}{FP + FN}$$ 假正例率

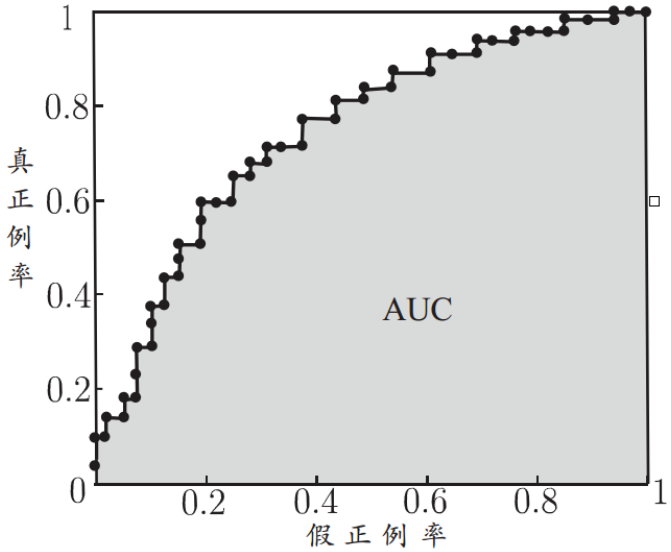
模型评估—性能度量

ROC图的绘制：



给定m^+个正例和m^-个负例，根据学习器预测结果对样例进行排序。

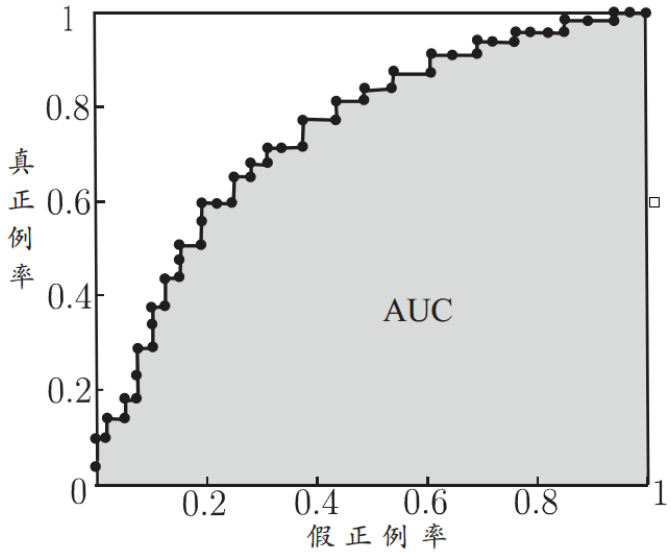
将分类阈值设为每个样例的预测值，当前标记点坐标为(x, y)，当前若为真正例，则对应标记点的坐标为$\left(x, y+\frac{1}{m^+}\right)$；当前若为假正例，则对应标记点的坐标为$\left(x+\frac{1}{m^{-1}}, y\right)$。然后用线段连接相邻点



基于有限样例绘制的 ROC 曲线与 AUC

模型评估—性能度量

若某个学习器的ROC曲线被另一个学习器的曲线“包住”，则后者性能优于前者；否则如果曲线交叉，可以根据ROC曲线下面积大小进行比较，也即AUC值



基于有限样例绘制的 ROC 曲线与 AUC

假设ROC曲线由$\{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$的点按序连接而形成 $(x_1 = 0, x_m = 1)$，则AUC可估算为

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

AUC衡量了样本预测的排序质量

性能度量—代价敏感错误率

- 现实任务中不同类型的错误所造成的后果很可能不同，为了权衡不同类型错误所造成的不同损失，可为错误赋予“非均等代价”。

- 以二分类为例，根据领域知识设定“代价矩阵”，如下表所示，$cost_{ij}$表示将第i类样本预测为第j类样本的代价。一般，$cost_{ii}=0$

真实类别	预测类别	
	第0类	第1类
第0类	0	$cost_{01}$
第1类	$cost_{10}$	0

- 损失程度相差越大，$cost_{01}$与 $cost_{10}$值的差别越大。

性能度量—代价敏感错误率

- 在非均等代价下，不再最小化错误次数，而是最小化“总体代价”，则“代价敏感”错误率相应的为：

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

性能度量—代价曲线

- 在非均等代价下，ROC曲线不能直接反映出学习器的期望总体代价，而“代价曲线”可以

- 代价曲线的横轴是取值为[0,1]的正例概率代价

$$P(+)cost = \frac{p \times cost_{01}}{p \times cost_{01} + (1-p) \times cost_{10}}$$

- 纵轴是取值为[0,1]的归一化代价

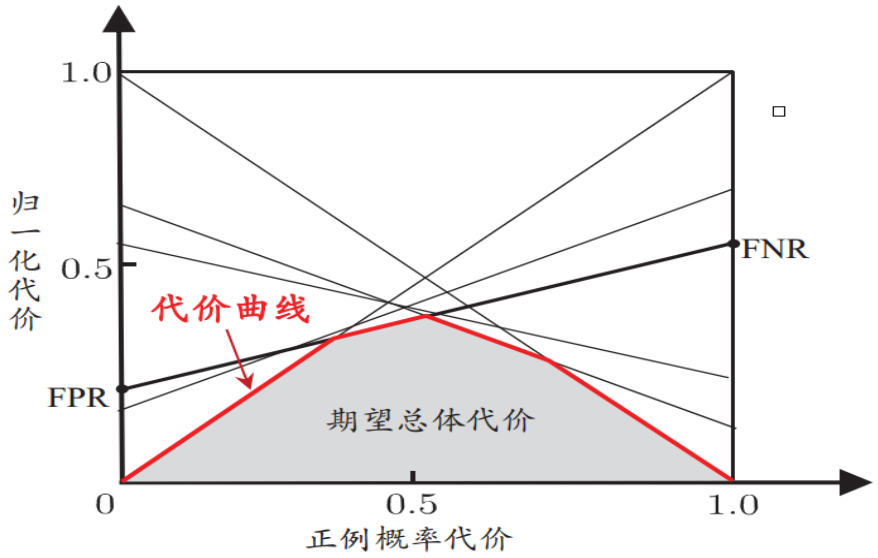
$$\mathrm{cost}_{\mathrm{norm}} = \mathrm{FNR} \times P(+)cost + FPR \times (1 - P(+)cost)$$

$FNR = 1 - TPR$ 假负例率

性能度量—代价曲线

代价曲线图的绘制：

- ROC曲线上每个点对应了代价曲线上的一条线段，设ROC曲线上点的坐标为(TPR,FPR),则可相应计算出FNR,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，围成的面积即为所有条件下学习器的期望总体代价。



代价曲线与期望总体代价

模型评估—比较检验

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，基于其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。

模型评估—二项检验

设泛化错误率为 ϵ，若测试错误率为 $\hat{\epsilon}$，对$\epsilon \leq \epsilon_0$进行假设检验

- 假定测试样本从样本总体分布中独立采样而来，可以使用“二项检验”对 $\epsilon \leq \epsilon_0$ 进行假设检验

- 求解$1-\alpha$概率内能看到的最大错误概率

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

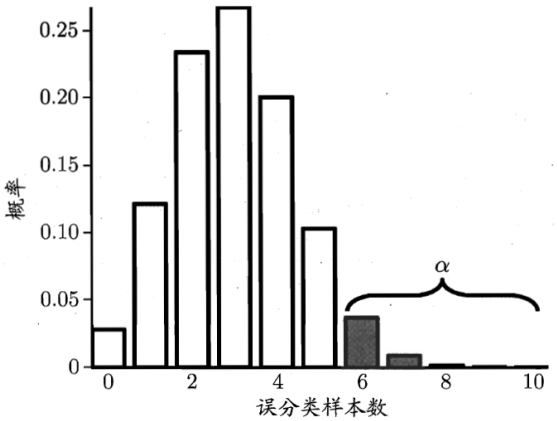


图 2.6 二项分布示意图($m = 10, \epsilon = 0.3$)

- 若$\hat{\epsilon} < \bar{\epsilon}$，则在$\alpha$显著度下，假设$\epsilon \leq \epsilon_0$不能被拒绝
- 否则，该假设被拒绝，即在α显著度下认为$\epsilon > \epsilon_0$

模型评估—t检验

设泛化错误率为 ϵ，若k个测试错误率为$\hat{\epsilon}_1, \hat{\epsilon}_2, \cdots, \hat{\epsilon}_k$，对$\epsilon = \epsilon_0$进行假设检验

- 多次留出法或交叉验证法进行训练/测试时可使用“t检验”
- 平均测试错误率μ和方差σ^2

$$\mu = \frac{1}{k}\sum_i \hat{\epsilon}_i$$

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

- 考虑到这k个测试错误率是泛化错误率ϵ_0的独立采样，则

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

- 服从自由度为k-1的t分布

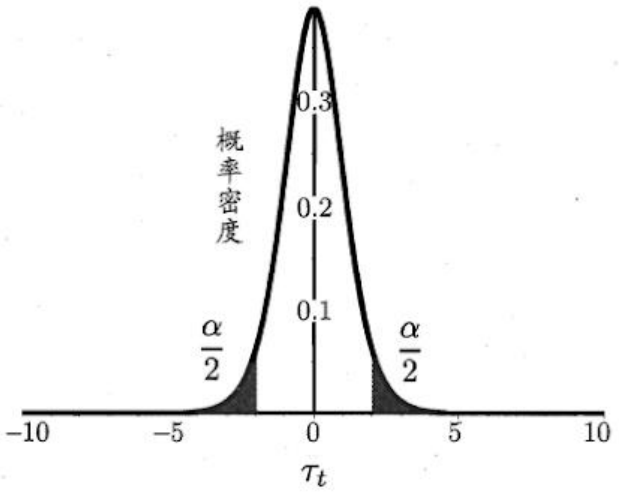
模型评估—t检验

设泛化错误率为 ϵ，若k个测试错误率为$\hat{\epsilon}_1, \hat{\epsilon}_2, \cdots, \hat{\epsilon}_k$，对$\epsilon = \epsilon_0$进行假设检验

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

服从自由度为k-1的t分布

- 考虑双边假设前提下，求解$1 - \alpha$概率内能看到的最大错误概率，对应右图阴影部分



•若$\tau_t \in \left[t_{-\alpha/2}, t_{\alpha/2}\right]$，则在$\alpha$显著度下，假设$\mu = \epsilon_0$不能被拒绝，即泛化误差率为$\epsilon_0$

•否则，该假设被拒绝，即泛化误差率与ϵ_0有显著不同

模型评估—交叉验证t检验

现实任务中，更多时候需要对不同学习器性能进行比较

- 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$\epsilon_1^A, \cdots, \epsilon_k^A$和$\epsilon_1^B, \cdots, \epsilon_k^B$，可用k折交叉验证“成对t检验”进行检验
- 先对每个结果求差$\Delta_i = \epsilon_i^A - \epsilon_i^B$
- 计算差值的均值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$t_{\alpha/2,k-1}$，则假设不能被拒绝，即两个学习器没有显著差别。

否则认为两个学习器有显著差别，平均错误率小的那个学习器性能更优。

模型评估—交叉验证t检验

假设检验的前提是测试错误率为泛化错误率的独立采样

然而由于样本有限，使用交叉验证导致训练集重叠，测试错误率并不独立，从而过高估计假设成立的概率

为缓解这一问题，可采用“5*2交叉验证”法.

模型评估—5*2交叉验证t检验

- 5*2折交叉验证就是做5次二折交叉验证
- 每次二折交叉验证之前将数据打乱，使得5次交叉验证中的数据划分不重复。
- 为缓解测试数据错误率的非独立性，仅计算第一次2折交叉验证结果的平均值$\mu = 0.5(\Delta_1^1 + \Delta_1^2)$和每次二折实验计算得到的方差 $\sigma_i^2 = \left(\Delta_i^1 - \frac{\Delta_i^1+\Delta_i^2}{2}\right)^2 + \left(\Delta_i^2 - \frac{\Delta_i^1+\Delta_i^2}{2}\right)^2$
- 变量$\tau_t = \frac{\mu}{\sqrt{0.2\sum_{i=1}^{5}\sigma_i^2}}$服从自由度为5的t分布

模型评估—McNemar检验

- 对于二分类问题，留出法不仅可以估计出学习器A和B的测试错误率，还能获得两学习器分类结果的差别，如下表所示

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}

对两学习器性能相同进行假设 等价于对$e_{01} = e_{10}$进行假设检验

McNemar 检验考虑变量

$$\tau_{\chi^2} = \frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}}$$

该变量服从自由度为1的卡方分布

模型评估—Friedman检验

- 交叉验证t检验和McNemar检验都是在一个数据集上比较两个算法的性能
- Friedman检验在一组数据集上对多个算法进行比较

- 假设用D_1, D_2, D_3, D_4四个数据集对算法 A, B, C进行比较
- 使用留出法或交叉验证法得到每个算法在每个数据集的测试结果
- 然后在每个数据集上根据性能好坏排序，并赋序值1,2,…
- 若算法性能相同则平分序值,继而得到每个算法的平均序值

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

模型评估—Friedman检验

- 由平均序值进行Friedman检验来判断这些算法是否性能都相同

算法比较序值表

数据集	算法 A	算法 B	算法 C
D_1	1	2	3
D_2	1	2.5	2.5
D_3	1	2	3
D_4	1	2	3
平均序值	1	2.125	2.875

假设在N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。**若这些算法性能都相同，则它们的平均序值应当相同**

若不考虑平分序值情况，那么

$$\mathrm{E}[r_i] = \frac{k+1}{2}, \mathrm{var}(r_i) = \frac{(k^2-1)}{12N}$$

当k和N都较大时，变量

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

服从自由度为k-1的卡方分布

模型评估—Nemenyi后续检验

- 若“所有算法的性能相同”这个假设被拒绝，说明算法的性能显著不同，此时可用Nemenyi后续检验进一步区分算法。

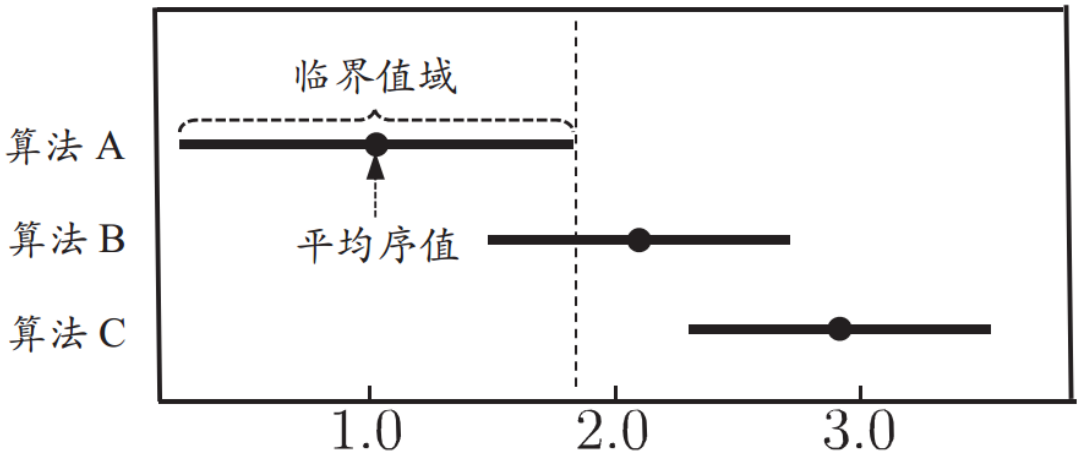
- Nemenyi检验计算平均序值差别的临界阈值

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

- 如果两个算法的平均序值之差超出了临界阈值CD，则以相应的置信度拒绝“两个算法性能相同”这一假设。

模型评估—Nemenyi后续检验

- 根据上例的序值结果可绘制如下Friedman检验图，横轴为平均序值，每个算法圆点为其平均序值，线段为临界阈值的大小。



- 若两个算法有交叠(A和B)，则说明没有显著差别
- 否则有显著差别(A和C),算法A明显优于算法C

模型评估—泛化性能解释

- 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华错误率进行拆解。

- 对测试样本$\boldsymbol{x}$，令y_D 为 $\boldsymbol{x}$ 在数据集中的标记，y为 $\boldsymbol{x}$ 的真实标记，$f(x;D)$为训练集 D 上学得模型 f 在 $\boldsymbol{x}$ 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：$\mathrm{var}(\boldsymbol{x}) = \mathbb{E}_D\left[\left(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x})\right)^2\right]$

输出噪声为：$\varepsilon^2 = \mathbb{E}_D[(y_D - y)^2]$

期望输出与真实标记的差别称为偏差为：$bias^2(\boldsymbol{x}) = \left(\bar{f}(\boldsymbol{x}) - y\right)^2$

模型评估—泛化性能解释

- 为便与讨论，假定噪声期望为0，也即 $\mathbb{E}_D[y_D - y] = 0$，

对泛化误差分解

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

$var(\boldsymbol{x})$　　$bias^2(\boldsymbol{x})$　　ε^2

泛化误差可分解为偏差、方差与噪声之和

模型评估—泛化性能解释

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

$var(\boldsymbol{x})$ $bias^2(\boldsymbol{x})$ ε^2

- 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

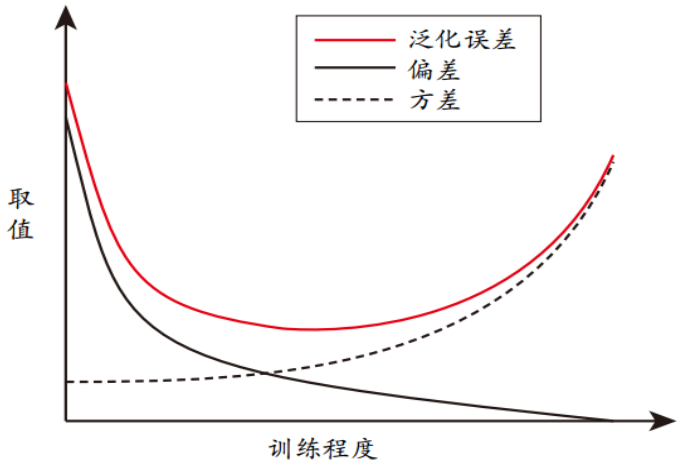
泛化性能是由学习算法的能力、数据的充分性以及学习任务本身的难度所共同决定的。给定学习任务为了取得好的泛化性能，需要使偏差小(充分拟合数据)而且方差较小(减少数据扰动产生的影响)。

模型评估—泛化性能解释

- 一般来说，偏差与方差是有冲突的，称为偏差-方差窘境。

- 如右图所示，假如我们能控制算法的训练程度：

  - 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

作业

- 习题2.2
- 习题2.4
- 习题2.5
- 习题2.9