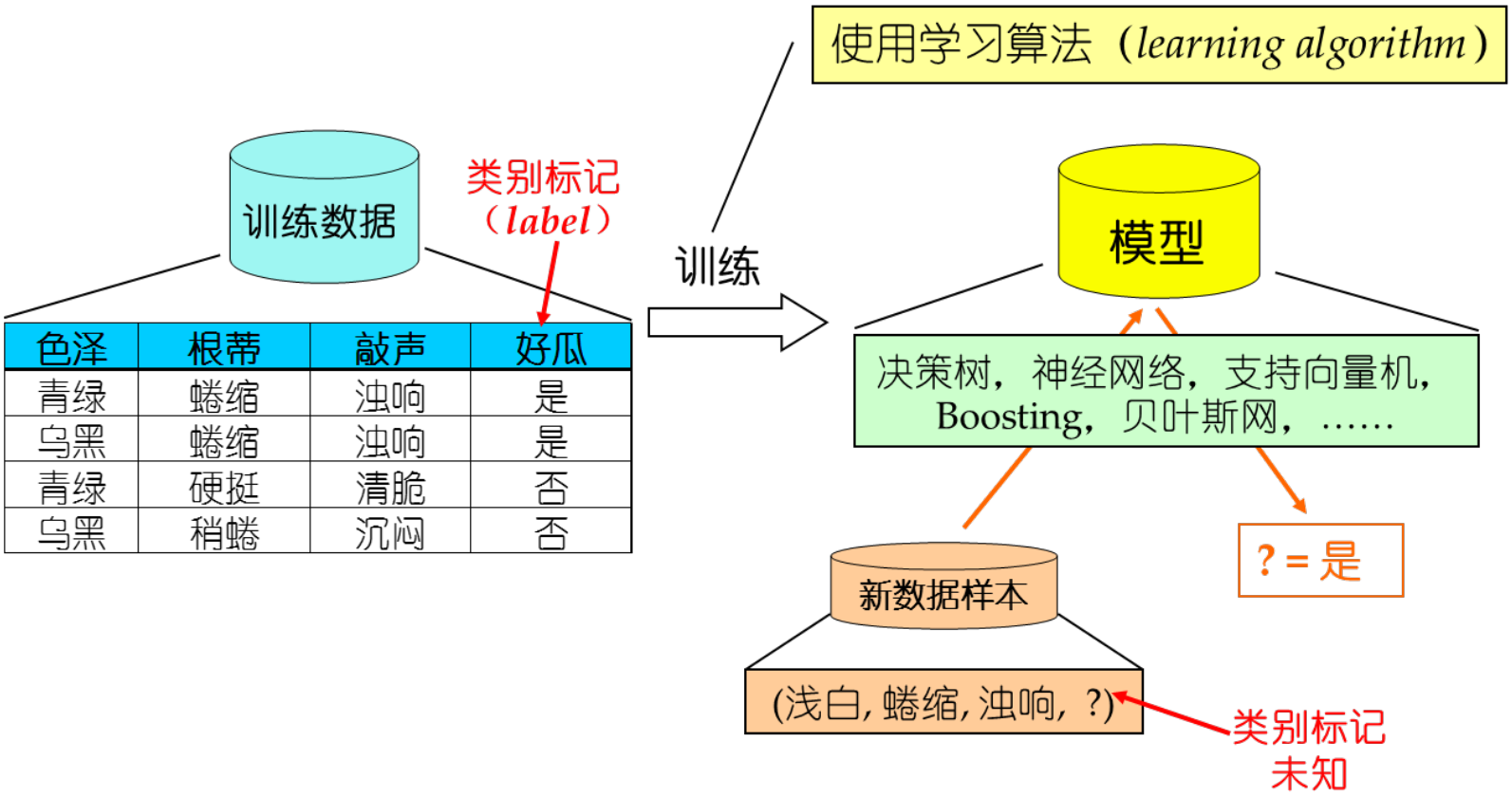
高级机器学习

第二讲 模型评估与模型选择

典型的机器学习过程



色泽	根蒂	敲声	好瓜
青绿	蜷缩	浊响	是
乌黑	蜷缩	浊响	是
青绿	硬挺	清脆	否
乌黑	稍蜷	沉闷	否

机器学习是无所不能的吗？No!

并非“一切皆可学”，例如：

- 特征信息不充分
  - 例如，重要特征信息没有获得
- 样本信息不充分
  - 例如，仅有很少的数据样本

机器学习具有坚实的理论基础

Leslie Valiant
(莱斯利•维利昂特)
(1949-)
2010年图灵奖

计算学习理论

Computational learning theory

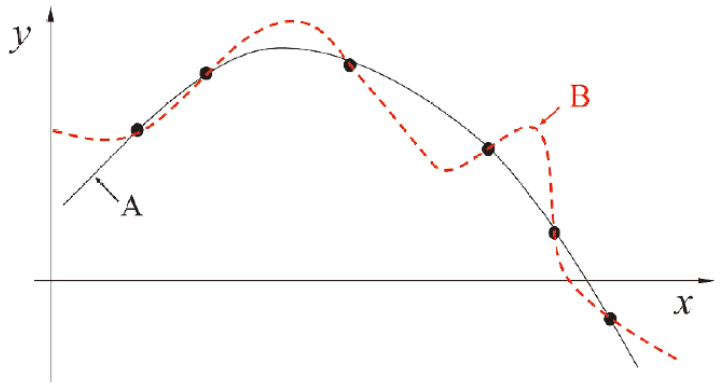
最重要的理论模型：

PAC (Probably Approximately Correct, 概率近似正确) learning model [Valiant, 1984]

$$P(|f(\boldsymbol{x}) - y| \leq \epsilon) \geq 1 - \delta$$

归纳偏好（inductive bias）

机器学习算法在学习过程中对某种类型假设的偏好



A更好？
B更好？

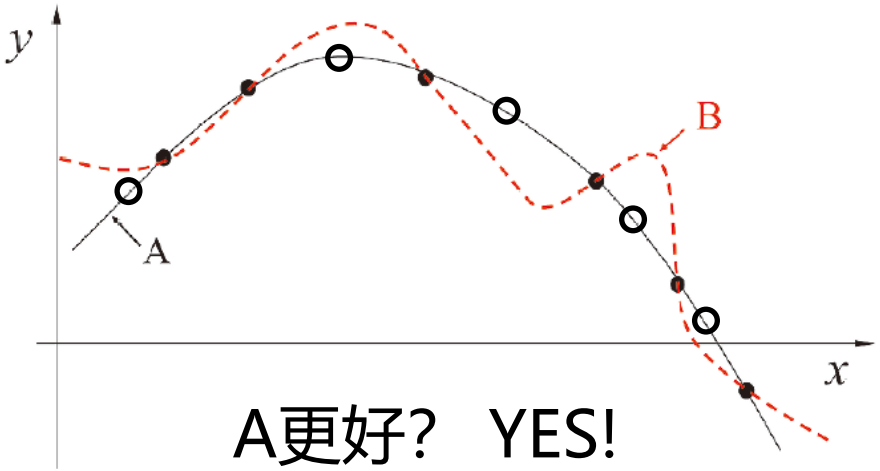
一般原则：
奥卡姆剃刀
(Occam's razor)

任何一个有效的机器学习算法必有其偏好

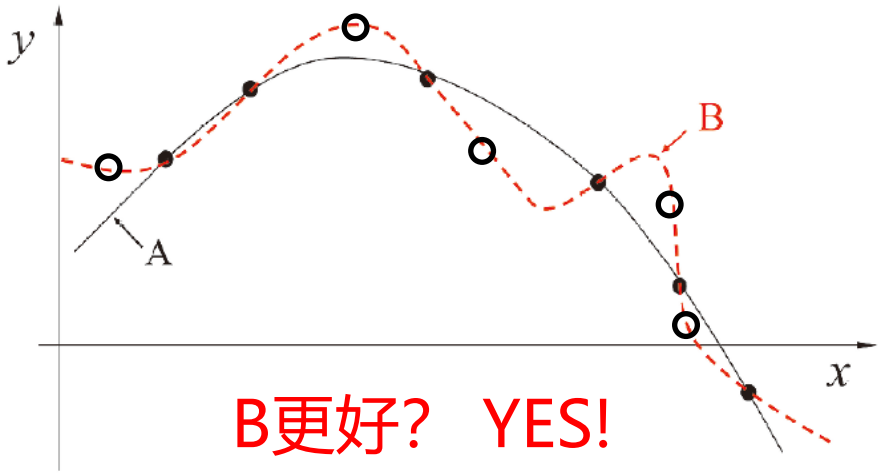
学习算法的归纳偏好是否与问题本身匹配，
大多数时候直接决定了算法能否取得好的性能！

哪个算法更好?

黑点：训练样本；白点：测试样本



A更好？ YES!



B更好？ YES!

没有免费的午餐!

NFL定理：一个算法 $\mathfrak{L}_a$ 若在某些问题上比另一个算法 $\mathfrak{L}_b$ 好，必存在另一些问题，$\mathfrak{L}_b$ 比 $\mathfrak{L}_a$ 好

哪个算法更好?

简单起见，假设样本空间 $\mathcal{X}$ 和假设空间 $\mathcal{H}$ 离散，令 $P(h|X,\mathfrak{L}_a)$ 代表算法 $\mathfrak{L}_a$ 基于训练数据 X 产生假设 h 的概率，f 代表要学的目标函数，$\mathfrak{L}_a$ 在训练集之外所有样本上的总误差为

$$E_{ote}(\mathfrak{L}_a|X,f)=\sum_{h}\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\,\mathbb{I}\left(h\left(\boldsymbol{x}\right)\neq f\left(\boldsymbol{x}\right)\right)P\left(h\mid X,\mathfrak{L}_a\right)$$

考虑二分类问题，目标函数可以为任何函数 $\mathcal{X}\mapsto\{0,1\}$ 函数空间为 $\{0,1\}^{|\mathcal{X}|}$ 对所有可能的 f 按均匀分布对误差求和，有

$$\sum_{f}E_{ote}(\mathfrak{L}_a|X,f)=\sum_{f}\sum_{h}\sum_{\boldsymbol{x}\in\mathcal{X}-X}P(\boldsymbol{x})\,\mathbb{I}(h(\boldsymbol{x})\neq f(\boldsymbol{x}))\;P(h\mid X,\mathfrak{L}_a)$$

哪个算法更好?

考虑二分类问题，目标函数可以为任何函数 $\mathcal{X} \mapsto \{0,1\}$ 函数空间为 $\{0,1\}^{|\mathcal{X}|}$ 对所有可能的 f 按均匀分布对误差求和，有

$$\begin{aligned}
\sum_f E_{ote}(\mathfrak{L}_a|X,f) &= \sum_f \sum_h \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x})\ \mathbb{I}(h(\boldsymbol{x}) \neq f(\boldsymbol{x}))\ P(h \mid X, \mathfrak{L}_a) \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h \mid X, \mathfrak{L}_a) \sum_f \mathbb{I}(h(\boldsymbol{x}) \neq f(\boldsymbol{x})) \\
&= \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h \mid X, \mathfrak{L}_a) \frac{1}{2} 2^{|\mathcal{X}|} \\
&= \frac{1}{2} 2^{|\mathcal{X}|} \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \sum_h P(h \mid X, \mathfrak{L}_a) \\
&= 2^{|\mathcal{X}|-1} \sum_{\boldsymbol{x}\in\mathcal{X}-X} P(\boldsymbol{x}) \cdot 1
\end{aligned}$$

总误差与学习算法无关！ ⇨ **所有算法同样好！**

NFL定理的寓意

NFL定理的重要前提：

所有“问题”出现的机会相同、或所有问题同等重要

实际情形并非如此；我们通常只关注自己正在试图解决的问题

脱离具体问题，空泛地谈论“什么学习算法更好”
毫无意义！

具体问题，具体分析！

现实机器学习应用

把机器学习的“十大算法”“二十大算法”都弄熟，

逐个试一遍，是否就“止于至善”了？

NO！

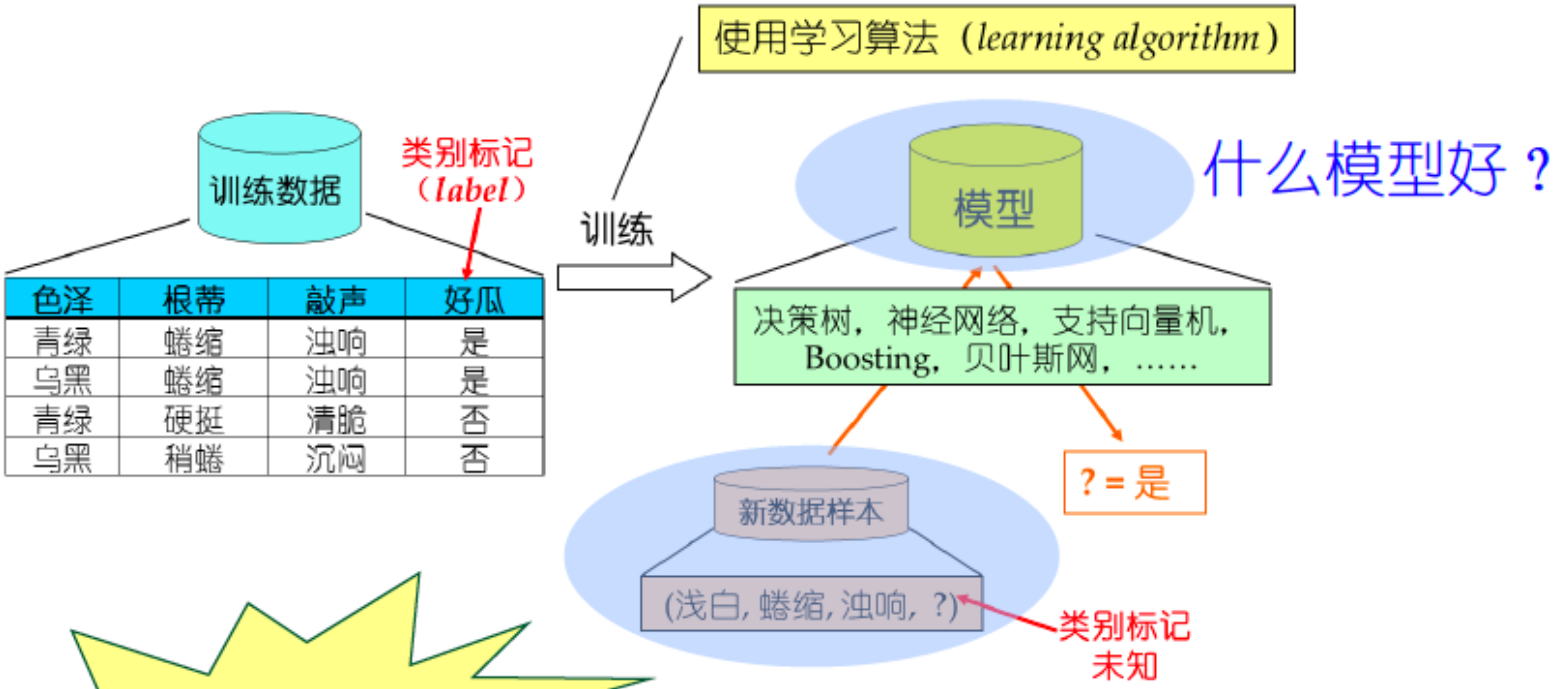
机器学习并非“十大套路”“二十大招数”的简单堆积

现实任务千变万化，

以有限的“套路”应对无限的“问题”，焉有不败？

最优方案往往来自：**按需设计、度身定制**

典型的机器学习过程



泛化能力强！

能很好地适用于 unseen instance

例如，**错误率**低、**精度**高

然而，我们手上没有 unseen instance，……

泛化误差 vs 经验误差

泛化误差：在“未来”样本上的误差

经验误差：在训练集上的误差，亦称“训练误差”

- 泛化误差越小越好
- 经验误差是否越小越好？

NO! 因为会出现“过拟合”(overfitting)

过拟合 （overfitting）vs 欠拟合（underfitting）

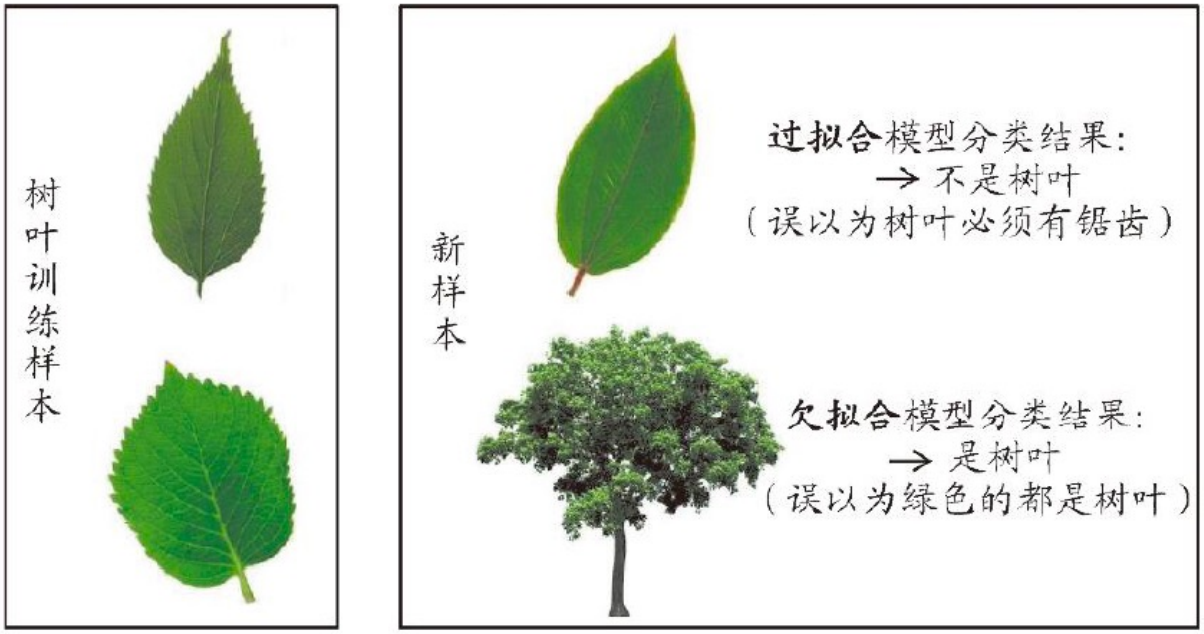


图 2.1 过拟合、欠拟合的直观类比

1. 在不存在“严重”过拟合或欠拟合现象时，经验误差和泛化误差可以进行相互界定；
2. 绝对地消除过拟合或欠拟合不可能；
3. 余下讨论中，假定算法没有严重过拟合现象。

模型选择 - 经验误差

三个关键问题：

- 如何获得测试结果？ ⇨ 评估方法
- 如何评估性能优劣？ ⇨ 性能度量
- 如何判断实质差别？ ⇨ 比较检验

评估方法

可是, 我们只有一个包含 m 个样例的数据集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$, 既要训练, 又要测试, 怎样才能做到呢? 答案是: 通过对 D 进行适当的处理, 从中产生出训练集 S 和测试集 T . 下面介绍几种常见的做法.

- 留出法（hold-out）
- 交叉验证法
- 留一法

越往下或更准确，但更复杂，开销越高

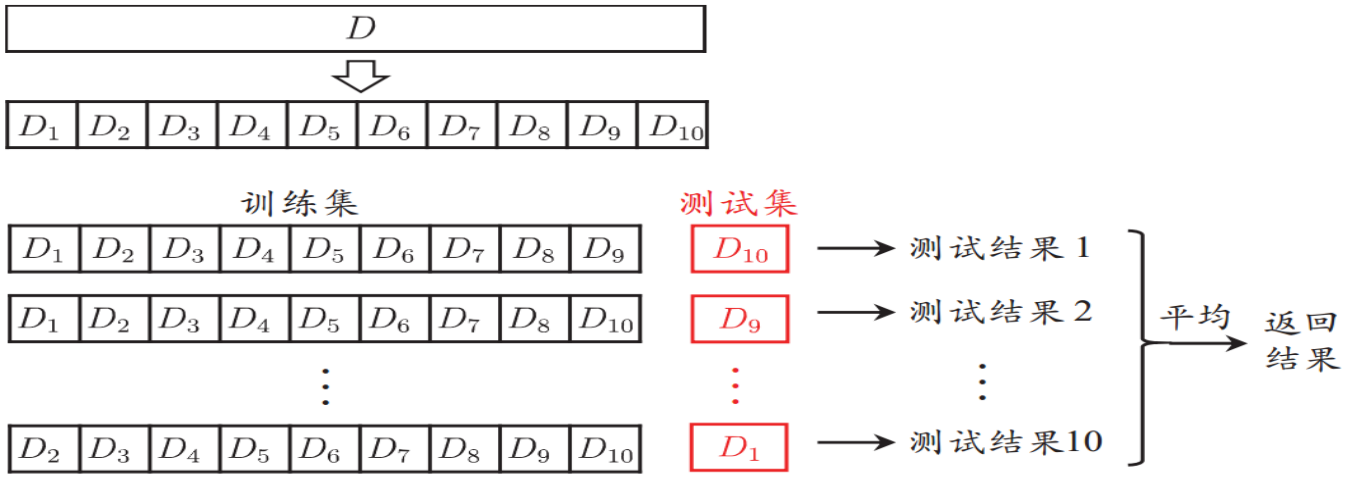
评估方法

- 留出法（hold-out）：
  - 直接将数据集划分为两个互斥集合——训练和测试集
  - 训练/测试集划分要尽可能保持数据分布的一致性
  - 分层采样（stratified sampling）：保持类别比例一致
  - 一般若干次随机划分、重复实验取平均值
  - 通常根据数据集大小划分训练/测试样本比例

评估方法

- 交叉验证法（cross-validation）：

  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

评估方法

P次k折交叉验证：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”

假设数据集D包含m个样本，若令k=m，则得到留一法(LOO)：

- 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，实际中不采用

模型选择

三个关键问题：

- 如何获得测试结果？ ⇨ 评估方法
- 如何评估性能优劣？ ⇨ 性能度量
- 如何判断实质差别？ ⇨ 比较检验

性能度量

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

回归任务最常用的性能度量是“均方误差”：

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

对于分类任务，错误率和精度是最常用的两种性能度量：

分类错误率

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

精度

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

性能度量

信息检索、Web搜索等场景中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，此时**查准率和查全率**比错误率和精度更适合。

统计真实标记和预测结果的组合可以得到“混淆矩阵”

分类结果混淆矩阵

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

查全率 $R = \frac{TP}{TP + FN}$

性能度量

比P-R曲线平衡点更用常用的是**F1**度量：

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

比F1更一般的形式 F_β

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

$\beta = 1$：标准F1

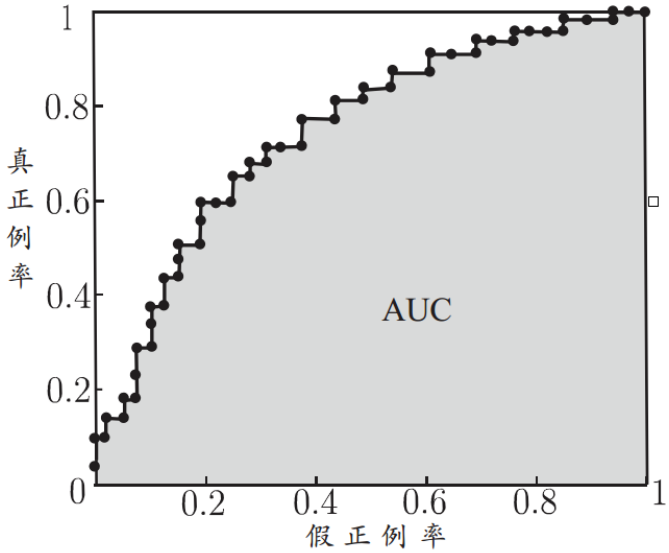
$\beta > 1$：偏重查全率(逃犯信息检索)

$\beta < 1$：偏重查准率(商品推荐系统)

性能度量

ROC曲线：遍历真比例率和假正例率，形成ROC曲线

AUC值：ROC曲线下面积大小



基于有限样例绘制的 ROC 曲线与 AUC

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

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

$$\mathrm{AUC} = 1 - \ell_{rank} .$$

AUC衡量样本预测的排序质量

模型选择

三个关键问题：

- 如何获得测试结果？ ⇨ 评估方法
- 如何评估性能优劣？ ⇨ 性能度量
- 如何判断实质差别？ ⇨ 比较检验

性能评估

- 性能比较的若干事实：
  - 测试性能不等于泛化性能
  - 测试性能随着测试集的变化而变化
  - 机器学习算法本身有一定的随机性

模型A 5次评估性能：91%，92%，93%，92%，92%

模型B 5次评估性能：86%，100%，99%，89%，91%

模型C 5次评估性能：85%，98%，98%，88%，90%

模型A、B、C平均性能分别是92%，93%，92.2%

你会选用哪个模型？为什么？

性能评估

- 性能比较的若干事实：
  - 测试性能不等于泛化性能
  - 测试性能随着测试集的变化而变化
  - 机器学习算法本身有一定的随机性

直接根据性能大小判断模型优劣可能会有错误，需要缓解上述不确定性

性能评估

- 多次实验，计算性能的均值与方差：**Mean+Std**

- **假设检验：**为学习器的性能比较提供了数学理论依据。利用假设检验可推断学习器A的泛化性能是否在统计意义上优于B，以及该结论的把握有多大
  - T检验（最常用）
  - Friedman检验
  - Nemenyi检验
  - 。。。

理论问题：泛化风险和经验风险的差距

“误差”包含了哪些因素？

换言之，从机器学习的角度看，

“误差”从何而来?

偏差-方差分解

对回归任务，泛化误差可通过“偏差-方差分解”拆解为：

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

期望输出与真实输出的差别

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

同样大小的训练集的变动，所导致的性能变化

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

训练样本的标记与真实标记有区别

表达了当前任务上任何学习算法所能达到的期望泛化误差下界

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

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共同决定

小结

- 经验误差与过拟合
  - 过拟合，欠拟合
- 评估方法
  - 留出法，交叉验证
- 性能度量
  - 均方误差，精度，查准率/查全率/F1，AUC
- 比较检验
  - T-检验
- 偏差与方差
  - 了解基本原理