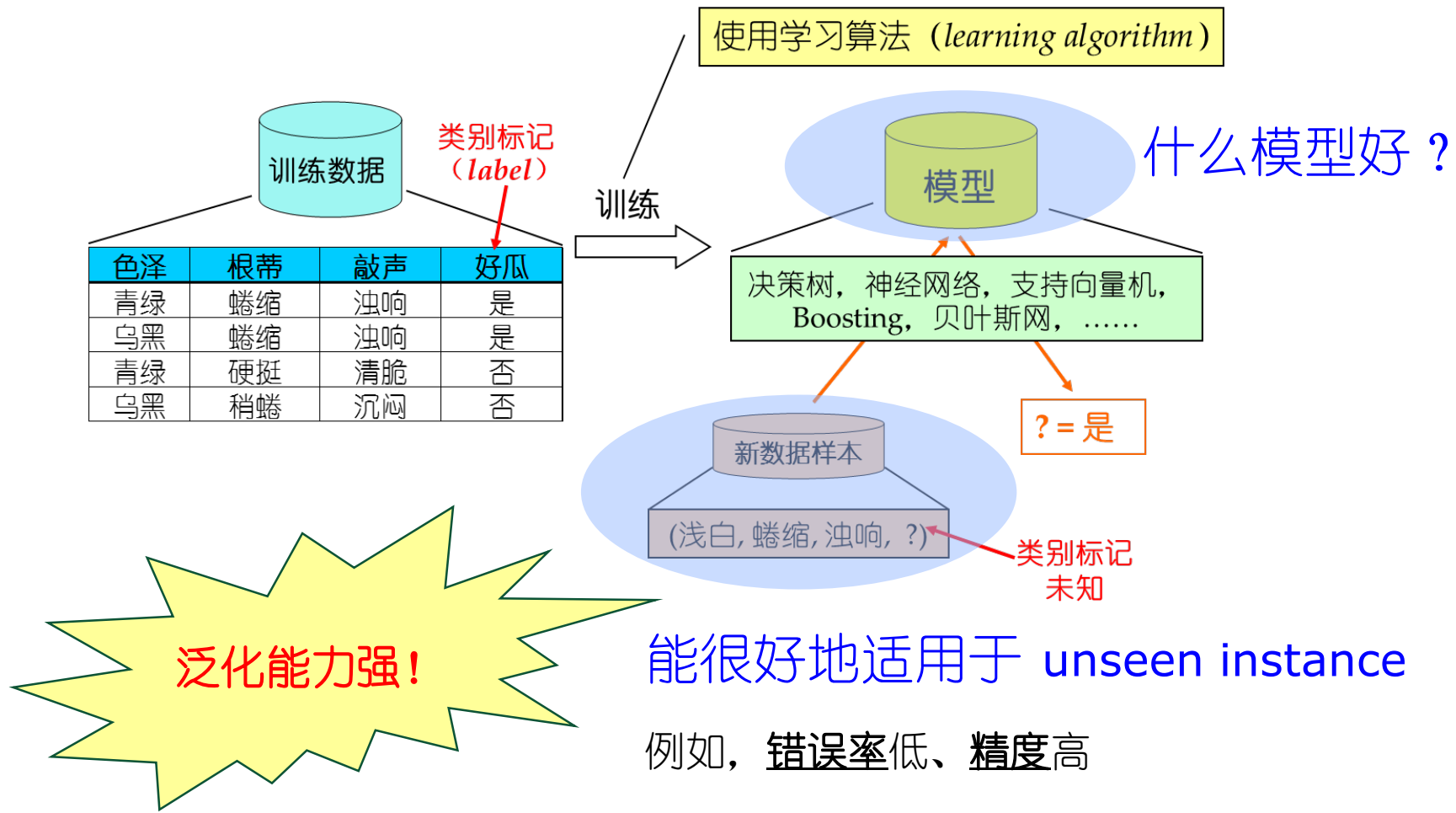


## 二、模型评估与选择

# 典型的机器学习过程



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)

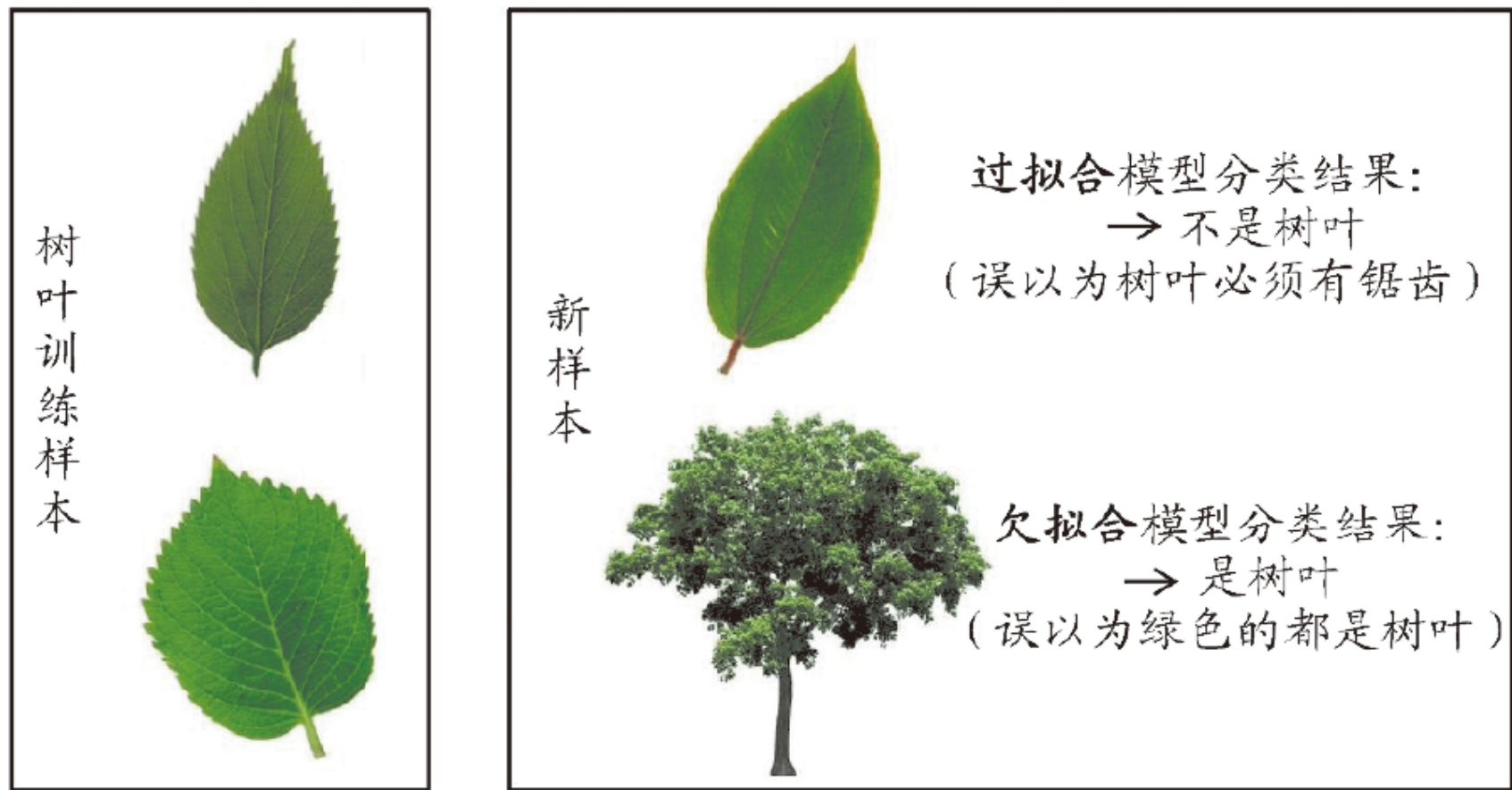
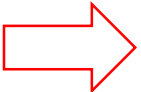
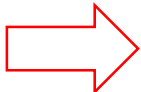
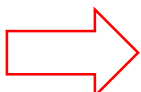


图 2.1 过拟合、欠拟合的直观类比

# 模型选择 (model selection)

---

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

---

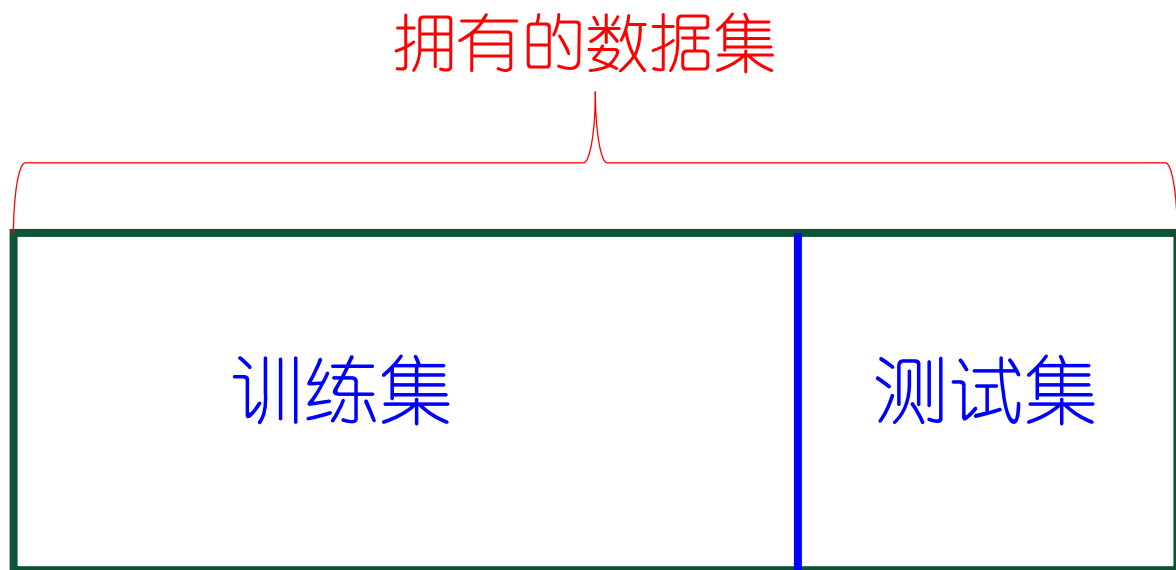
关键：怎么获得“测试集” (test set) ？

测试集应该与训练集“互斥”

常见方法：

- ▣ 留出法 (hold-out)
- ▣ 交叉验证法 (cross validation)
- ▣ 自助法 (bootstrap)

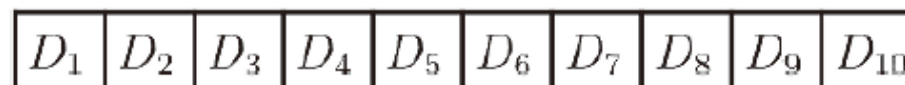
# 留出法



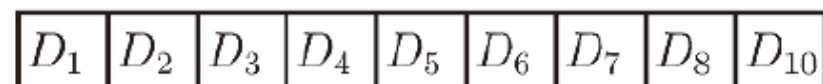
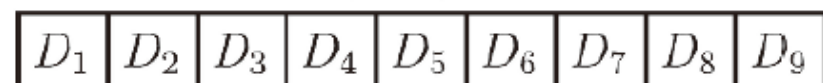
注意：

- 保持数据分布一致性（例如：分层采样）
- 多次重复划分（例如：**100**次随机划分）
- 测试集不能太大、不能太小（例如： **$1/5 \sim 1/3$** ）

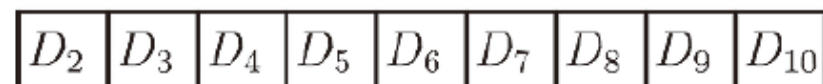
# $k$ -折交叉验证法



训练集



$\vdots$



测试集



→ 测试结果 1



→ 测试结果 2

$\vdots$



→ 测试结果 10

平均  
返回  
结果

若  $k = m$ , 则得到 “留一法”  
(leave-one-out, LOO)

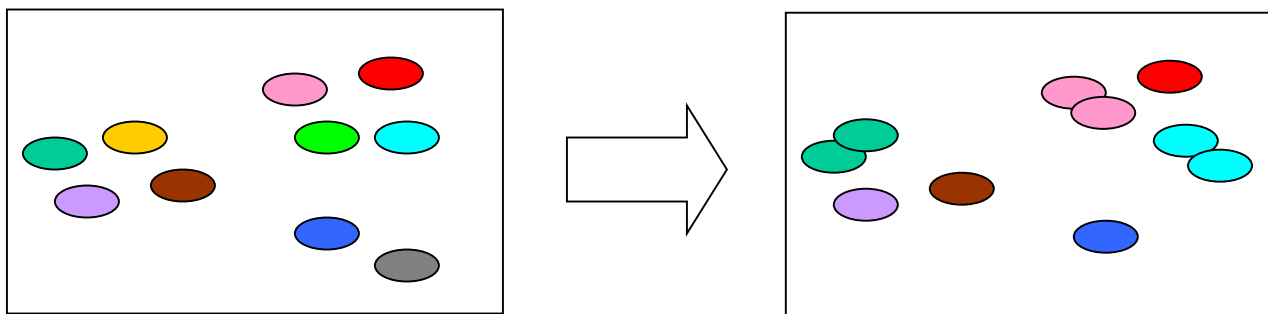
图 2.2 10 折交叉验证示意图



# 自助法

基于“自助采样” (bootstrap sampling)

亦称“有放回采样”、“可重复采样”



约有 36.8% 的样本不出现

$$\downarrow \lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$$

“包外估计” (out-of-bag estimation)

➤ 训练集与原样本集同规模

➤ 数据分布有所改变

# “调参”与最终模型

---

算法的参数：一般由人工设定，亦称“超参数”

模型的参数：一般由学习确定

调参过程相似：先产生若干模型，然后基于某种评估方法进行选择

参数调得好不好对性能往往对最终性能有关键影响

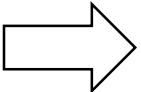
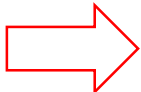
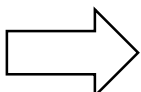
区别：训练集 vs. 测试集 vs. 验证集 (validation set)

算法参数选定后，要用“训练集+验证集”重新训练最终模型

# 模型选择 (model selection)

---

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(performance measure)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

什么样的模型是“好”的，不仅取决于算法和数据，还取决于任务需求

▣ 回归(regression) 任务常用均方误差：

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

# 错误率 vs. 精度

---

□ 错误率：

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

□ 精度：

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

# 查准率 vs. 查全率

表 2.1 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：
$$R = \frac{TP}{TP + FN}$$

# F1

F1 度量:

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

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

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

若对查准率/查全率有不同偏好:

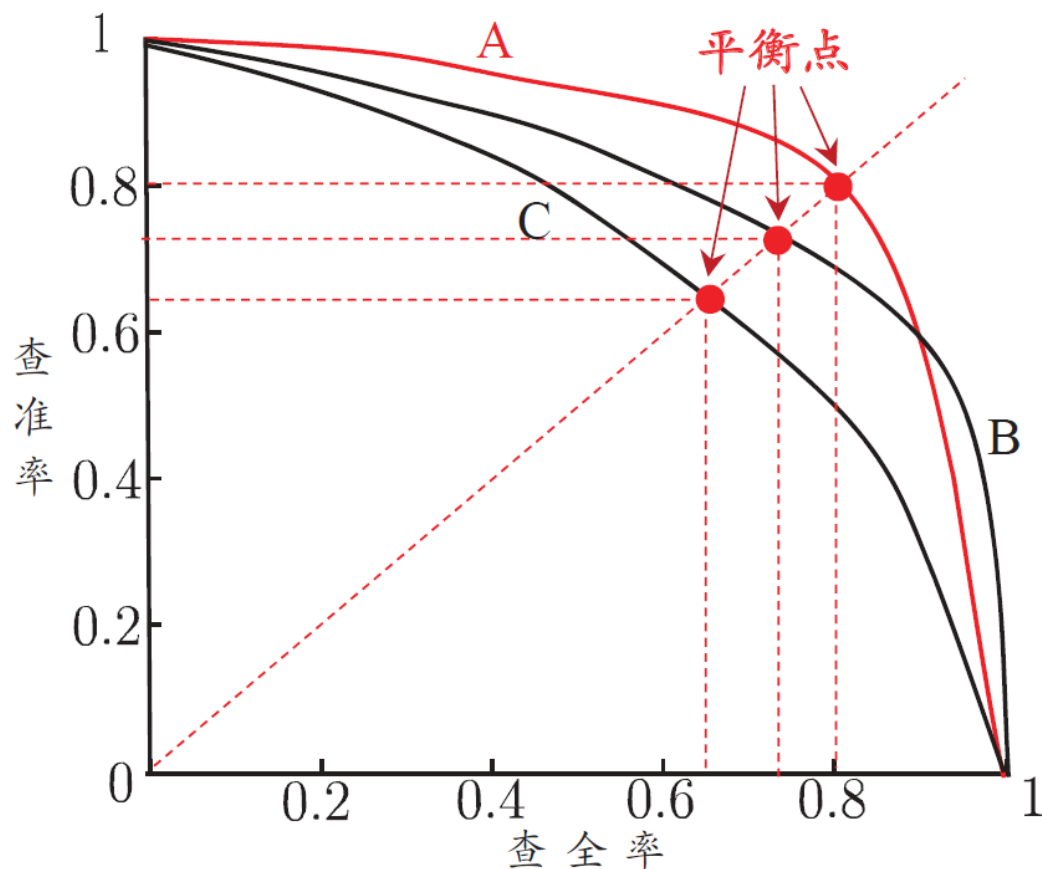
$$F_{\beta} = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

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

$\beta > 1$  时查全率有更大影响;  $\beta < 1$  时查准率有更大影响

# PR图, BEP

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



PR图:

- 学习器 A 优于 学习器 C
- 学习器 B 优于 学习器 C
- 学习器 A ?? 学习器 B

BEP:

- 学习器 A 优于 学习器 B
- 学习器 A 优于 学习器 C
- 学习器 B 优于 学习器 C



# 宏XX vs. 微XX

若能得到多个混淆矩阵：

(例如多次训练/测试的结果，多分类的两两混淆矩阵)

宏(**macro-**)查准率、查全率、F1

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

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

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

微(**micro-**)查准率、查全率、F1

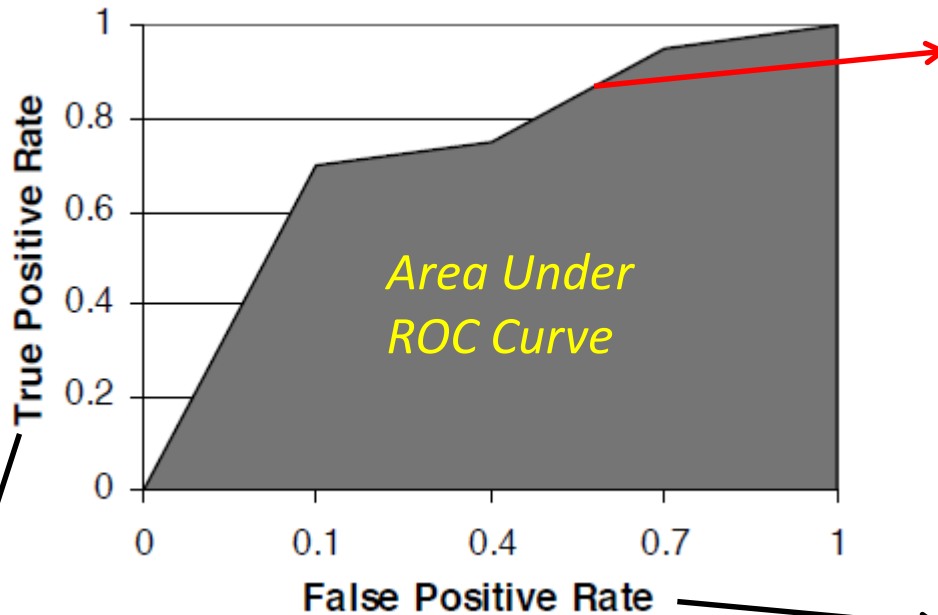
$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} ,$$

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

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

# ROC, AUC

AUC: **A**rea **U**nder the ROC **C**urve



ROC (Receiver Operating Characteristic) Curve [Green & Swets, Book 66; Spackman, IWML'89]

*The bigger, the better*

$$tpr = \frac{TP}{TP + FN} = \frac{TP}{m_+}$$

$$fpr = \frac{FP}{FP + TN} = \frac{FP}{m_-}$$

$$AUC = 1 - \frac{1}{m^+ m^-} \sum_{x^+ \in D^+} \sum_{x^- \in D^-} \left( \mathbb{I}(f(x^+) < f(x^-)) + \frac{1}{2} \mathbb{I}(f(x^+) = f(x^-)) \right)$$

# 非均等代价

犯不同的错误往往会造成不同的损失

此时需考虑“非均等代价”  
(unequal cost)

表 2.2 二分类代价矩阵

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

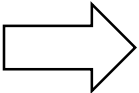
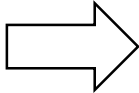
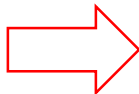
□ 代价敏感(cost-sensitive)错误率：

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

# 模型选择 (model selection)

---

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

# 比较检验

---

在某种度量下取得评估结果后，是否可以直接比较以评判优劣？

- NO ! 因为：**
- 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
  - 很多机器学习算法本身有一定的随机性

机器学习 ..... “概率近似正确”

# 常用方法

统计假设检验 (hypothesis test) 为学习器性能比较提供了重要依据

## □ 两学习器比较

➤ 交叉验证 t 检验 (基于成对 t 检验)

k 折交叉验证; 5x2交叉验证

➤ McNemar 检验 (基于列联表, 卡方检验)

## □ 多学习器比较

➤ Friedman + Nemenyi

- Friedman检验 (基于序值, F检验; 判断“是否都相同”)
- Nemenyi 后续检验 (基于序值, 进一步判断两两差别)



统计显著性

# 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

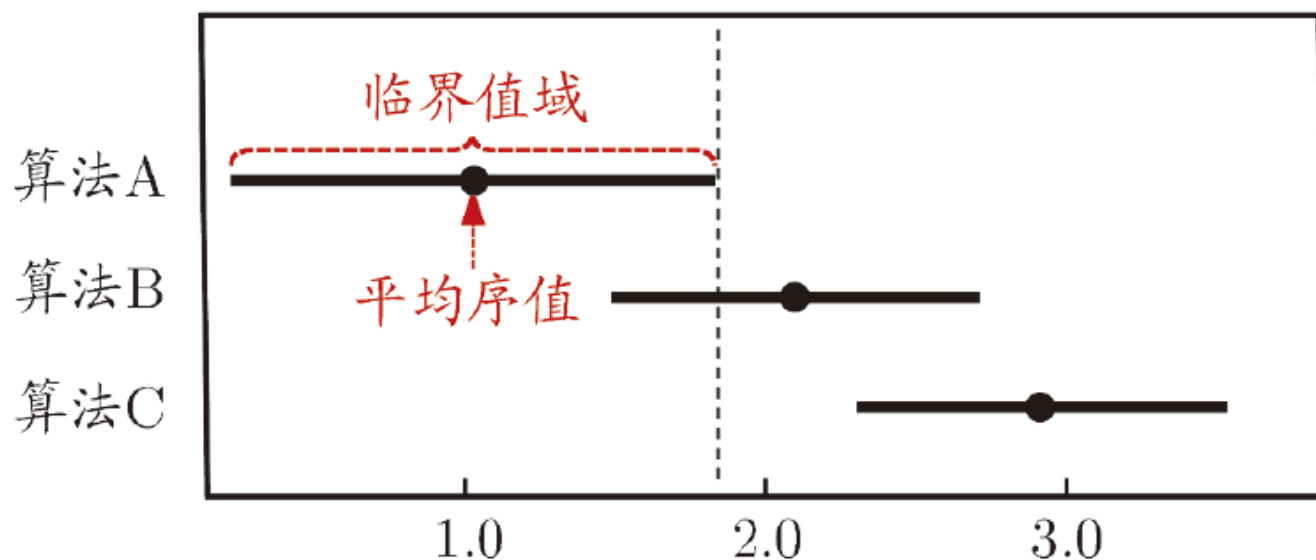


图 2.8 Friedman 检验图

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 显著优于算法 C

---

“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 (bias-variance decomposition)

对回归任务，泛化误差可通过“偏差-方差分解”拆解为：

$$E(f; D) = \underbrace{bias^2(\mathbf{x})}_{\text{red}} + \underbrace{var(\mathbf{x})}_{\text{blue}} + \underbrace{\varepsilon^2}_{\text{green}}$$

期望输出与真实输出的差别

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

同样大小的训练集的变动，所导致的性能变化

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

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

# 偏差-方差窘境 (bias-variance dilemma)

一般而言，偏差与方差存在冲突：

- 训练不足时，学习器拟合能力不强，偏差主导
- 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
- 训练充足后，学习器的拟合能力很强，方差主导

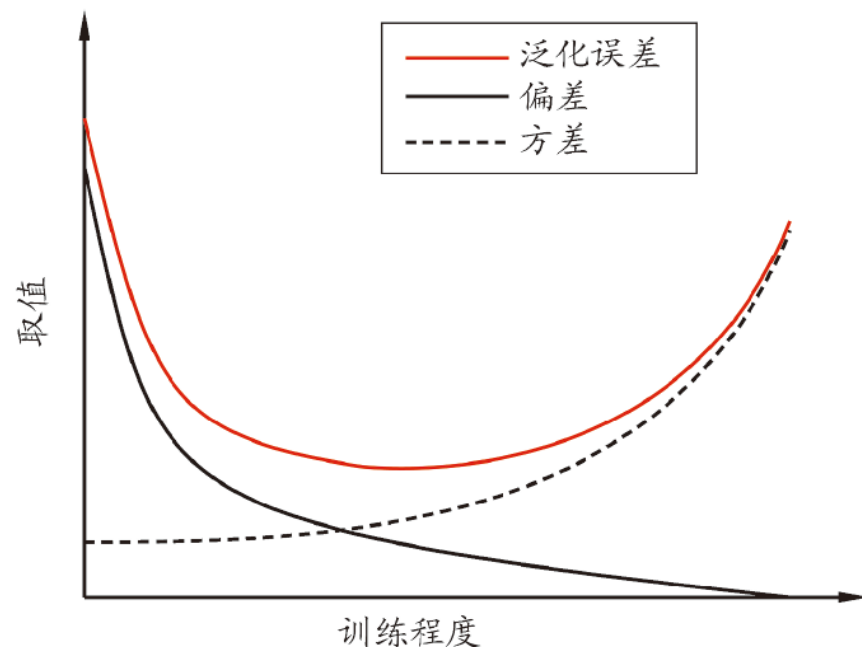


图 2.9 泛化误差与偏差、方差的关系示意图

前往第三站.....

