一、绪 论

主讲教师：周志华

课程主页

https://cs.nju.edu.cn/zhouzh/zhouzh.files/course/ml.htm

机器学习导论

https://cs.nju.edu.cn/zhouzh/zhouzh.files/course/ml.htm

[Home]

机器学习导论

课程代码：	22011430
授课对象：	计算机科学与技术系
学生人数：	150
上课时间：	2022年秋季学期，每周一，10:10 – 12:00
上课地点：	南京大学仙林校区 仙II-122
教学用书：	周志华 著. 机器学习，北京：清华大学出版社，2016年1月. {教材勘误}
讲义作业：	{内部网站}（本班同学校内访问）
主讲老师：	周志华 教授
教学组老师：	叶翰嘉 博士（关于作业、答疑、考试方面的问题，请联系叶老师）

[Home]

课程作业

6 次作业，每2-3周一次

Deadline: 每次作业布置后 一般两周截止，请看作业网站的规定

https://cs.nju.edu.cn/zhouzh/zhouzh.files/course/ml.htm

[Home]

机器学习导论

课程代码:	22011430
授课对象:	计算机科学与技术系
学生人数:	150
上课时间:	2022年秋季学期，每周一，10:10 – 12:00
上课地点:	南京大学仙林校区 仙II-122
教学用书:	周志华 著. 机器学习，北京：清华大学出版社，2016年1月. {教材勘误}
讲义作业:	{内部网站} (本班同学校内访问)
主讲老师:	周志华 教授
教学组老师:	叶翰嘉 博士 (关于作业、答疑、考试方面的问题，请联系叶老师)

[Home]

课程成绩

- **能力测试：20%**

  6次作业中，各人自选**1**次

- **平时成绩：40%**

  其他**5**次作业中，各人自选**4**次之和

- **期末考试：40%**

Deadline之后提交的作业，以此次0分计算

助教团队

叶翰嘉老师带领

博士生助教：

胡文超，高恩豪、施意、张逸凯

负责作业、答疑、考试、评分

上述方面的问题请直接联系叶翰嘉老师

yehj@nju.edu.cn

课堂纪律

从不点名

来去自由

保持安静！！

欢迎旁听

假设空间

表 1.1 西瓜数据集

编号	色泽	根蒂	敲声	好瓜
1	青绿	蜷缩	浊响	是
2	乌黑	蜷缩	浊响	是
3	青绿	硬挺	清脆	否
4	乌黑	稍蜷	沉闷	否

(色泽=?)∧(根蒂=?)∧(敲声=?)↔好瓜

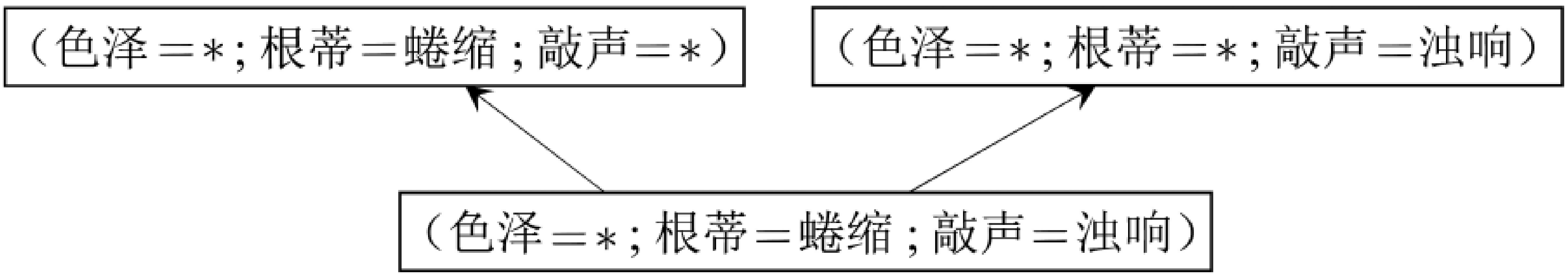
学习过程 → 在所有假设(**hypothesis**)组成的空间中进行**搜索**的过程

目标：找到与训练集“匹配”(fit)的假设

假设空间的大小：(n1+1) x (n2+1) x (n3+1) + 1

版本空间

版本空间(version space)：与训练集一致的假设集合



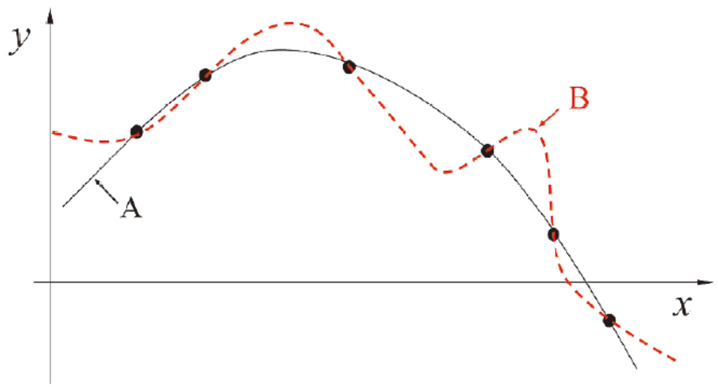
在面临新样本时，会产生不同的输出

例如：(青绿; 蜷缩; 沉闷)

应该采用哪一个模型(假设)?

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？

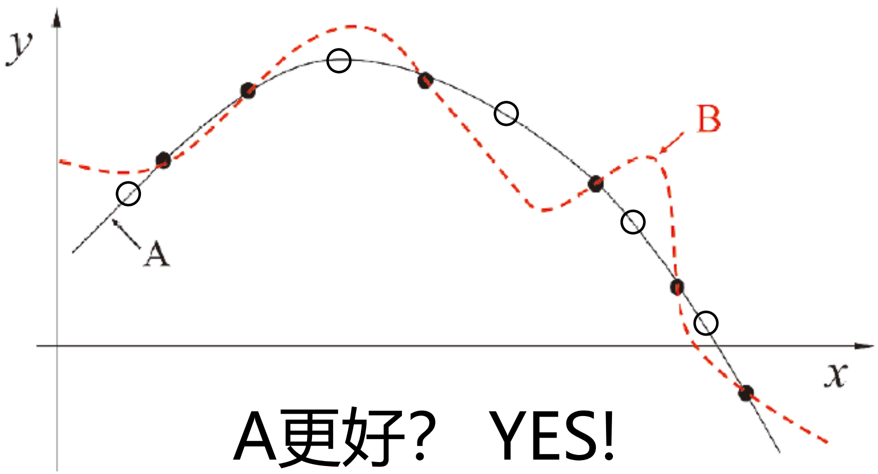
一般原则：
奥卡姆剃刀
(Ocam's razor)

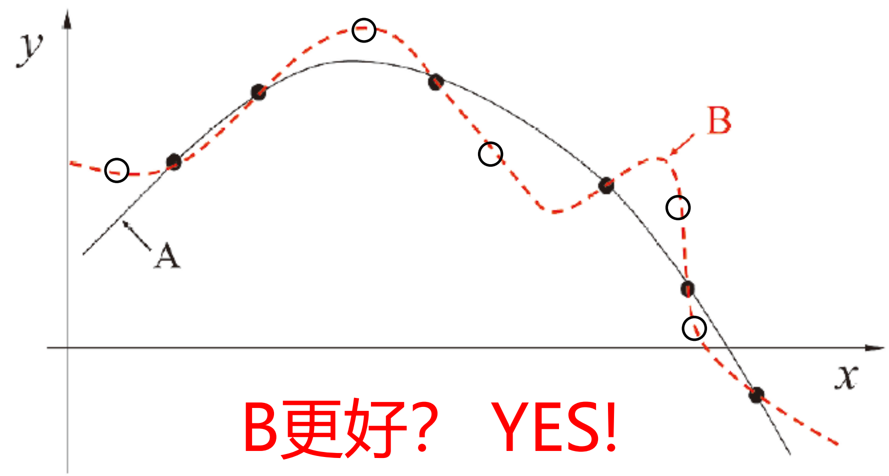
任何一个有效的机器学习算法必有其偏好

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





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

NFL定理

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 $\boldsymbol{X}$ 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

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

NFL定理

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！ $\Rightarrow$ **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！

现实机器学习应用中 ……

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？

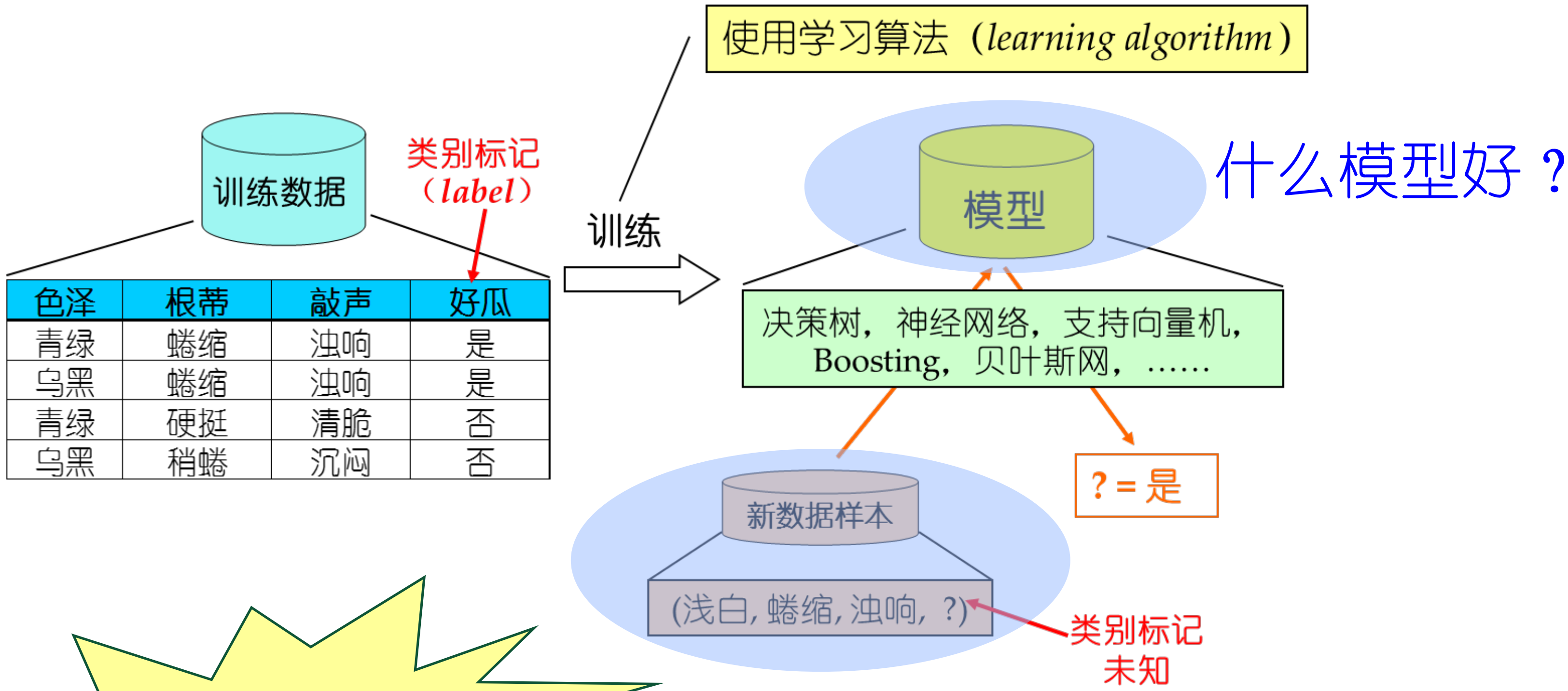
最优方案往往来自：**按需设计、度身定制**

前往第二站……

二、模型评估与选择

主讲教师：周志华

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

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)

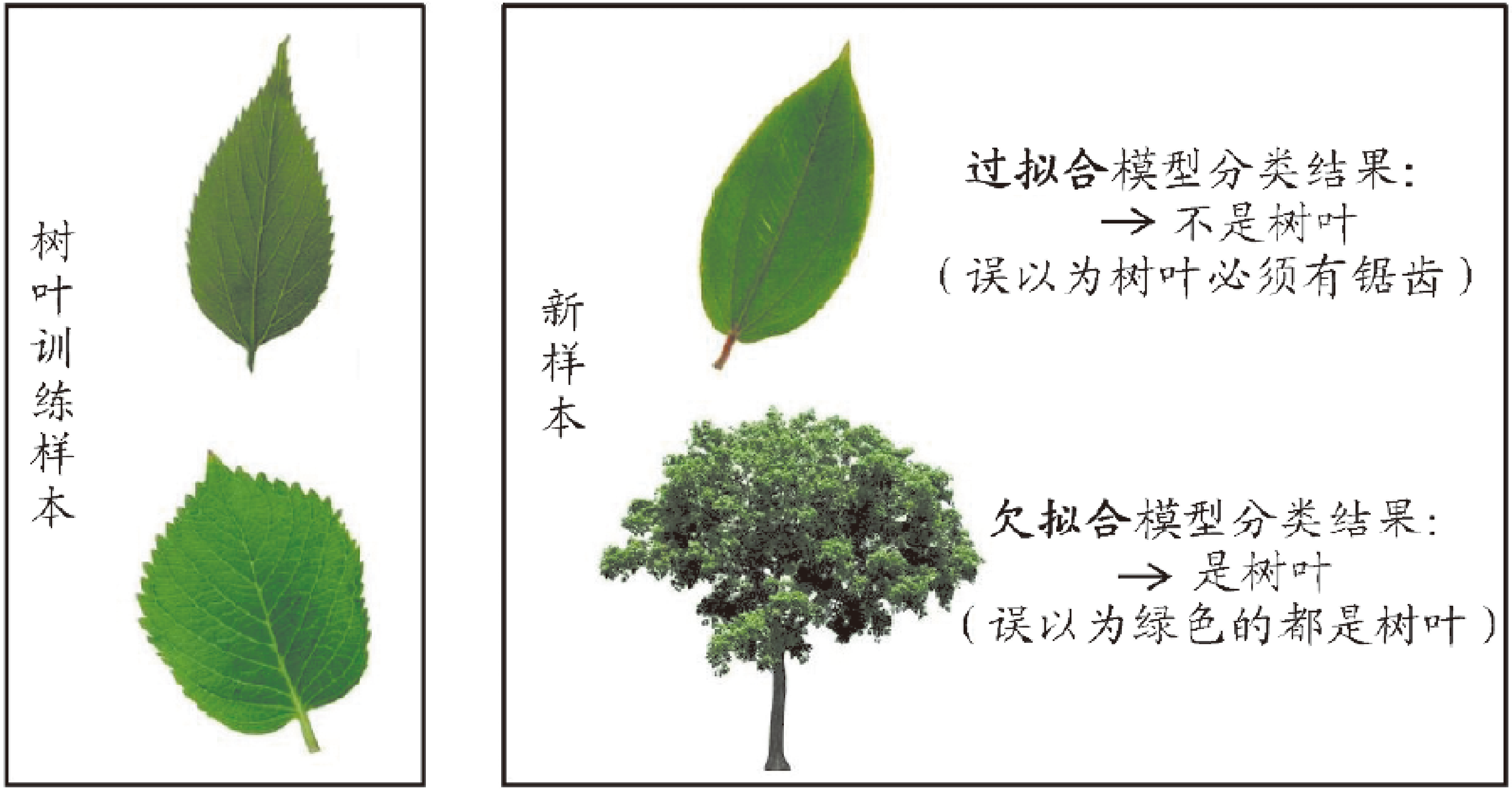


图 2.1 过拟合、欠拟合的直观类比

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

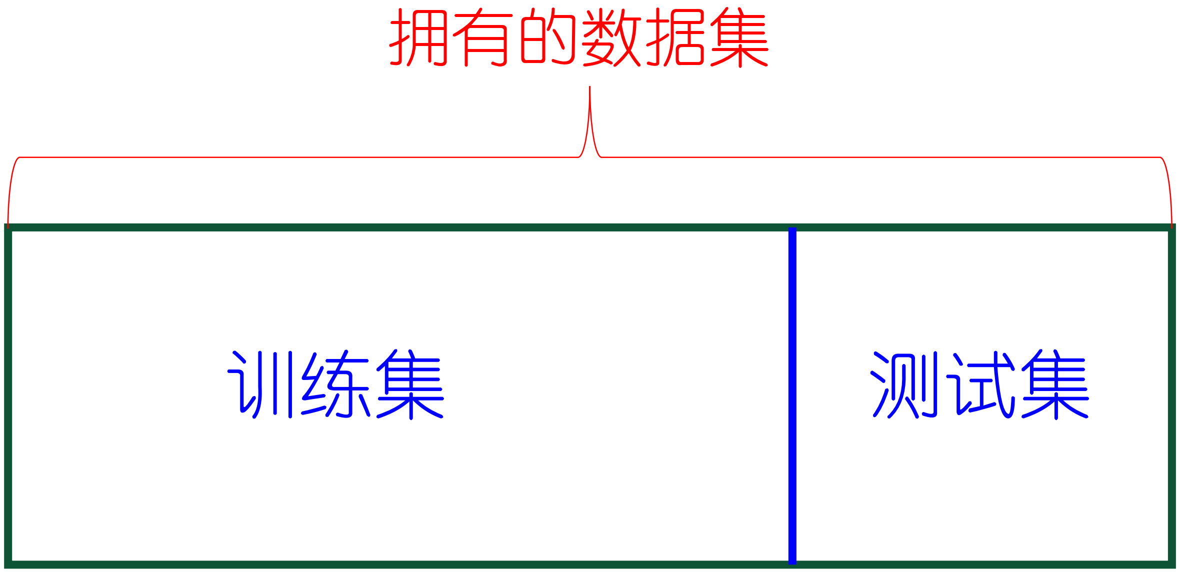
关键：怎么获得“测试集”(test set) ？

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：1/5~1/3)

k-折交叉验证法

若 k = m，则得到“留一法”
(leave-one-out, LOO)

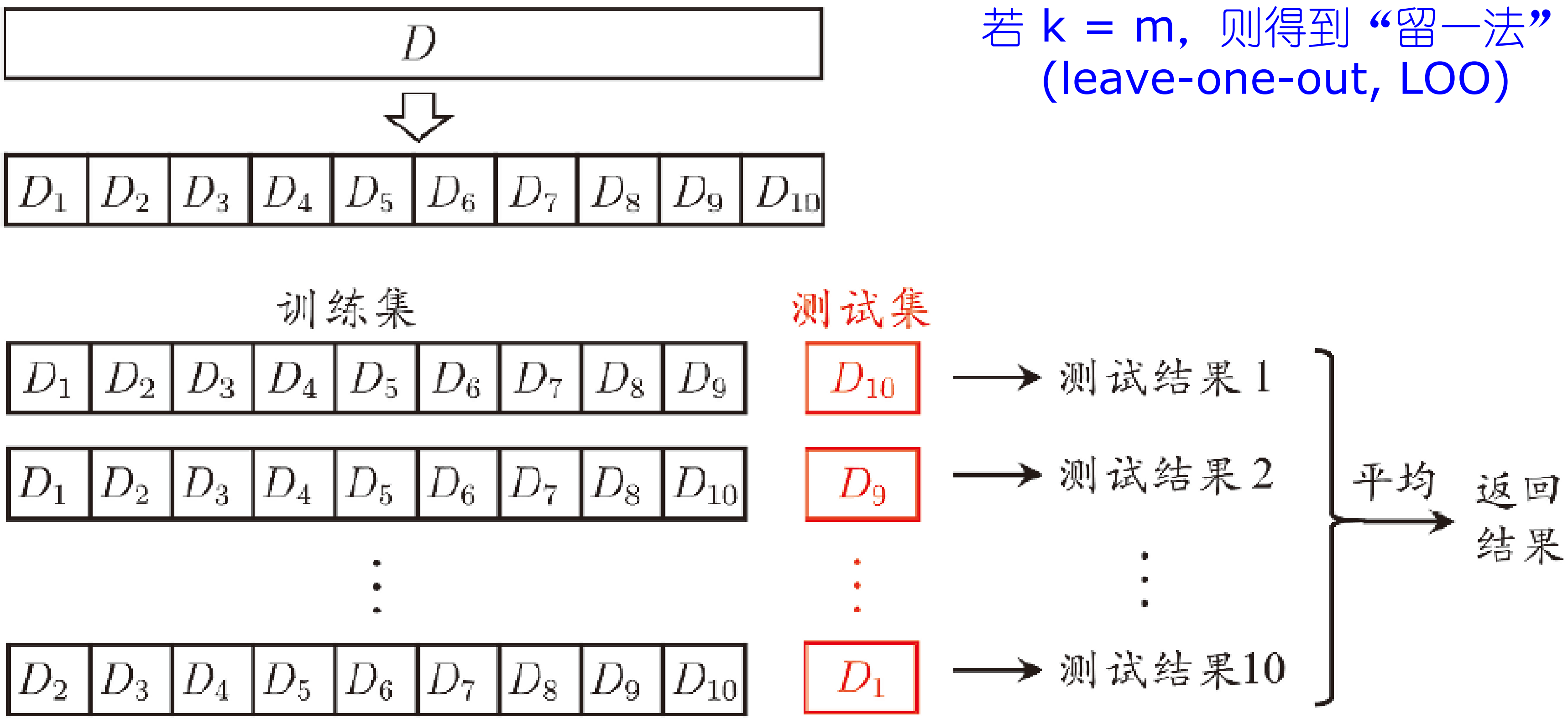
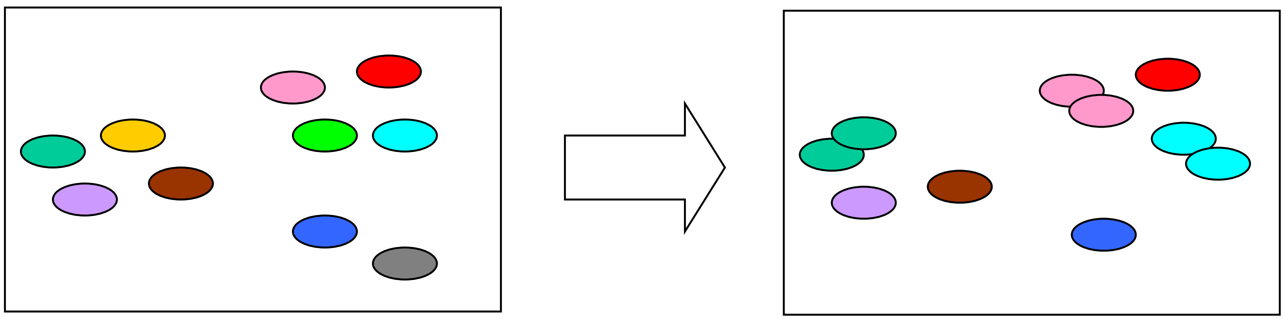


图 2.2 10 折交叉验证示意图

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

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

“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：

$$E(f ; D)=\frac{1}{m} \sum_{i=1}^{m}\left(f\left(\boldsymbol{x}_{i}\right)-y_{i}\right)^{2}$$

错误率 vs. 精度

- 错误率：

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

- 精度：

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

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：

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

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

若对查准率/查全率有不同偏好：

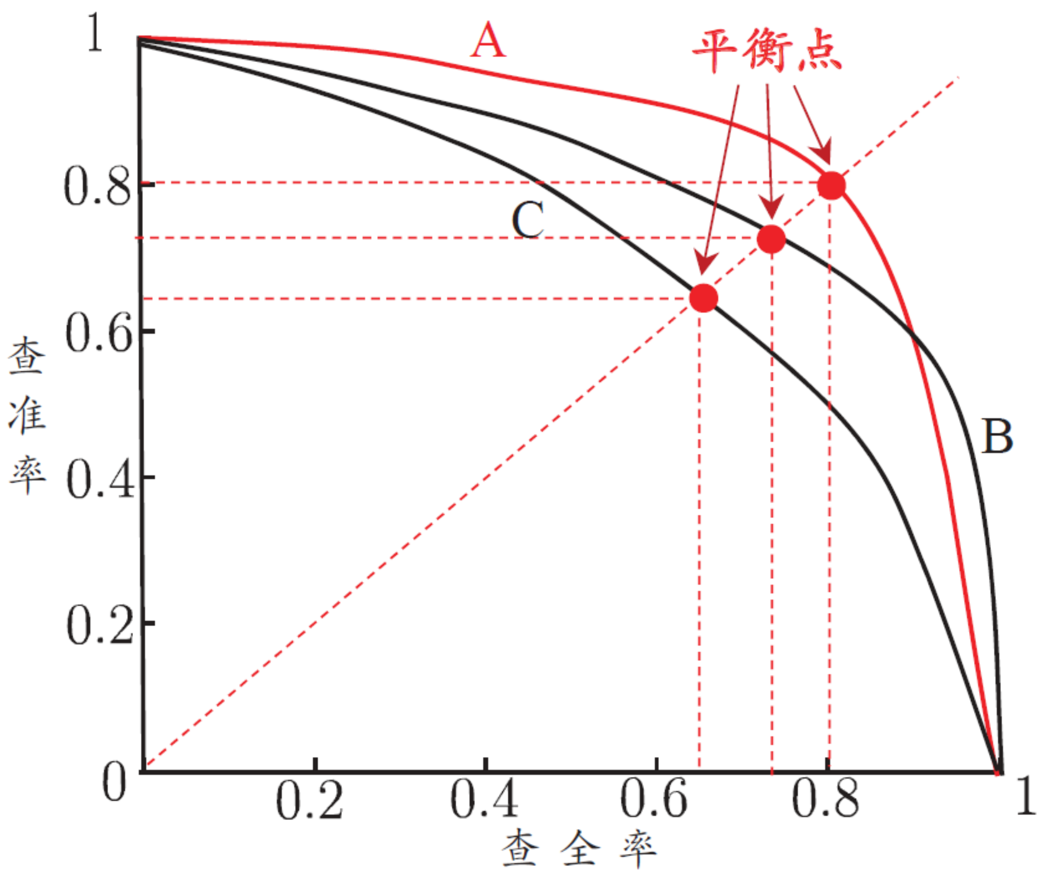
$$F_\beta = \frac{(1+\beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

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

$\beta > 1$ 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

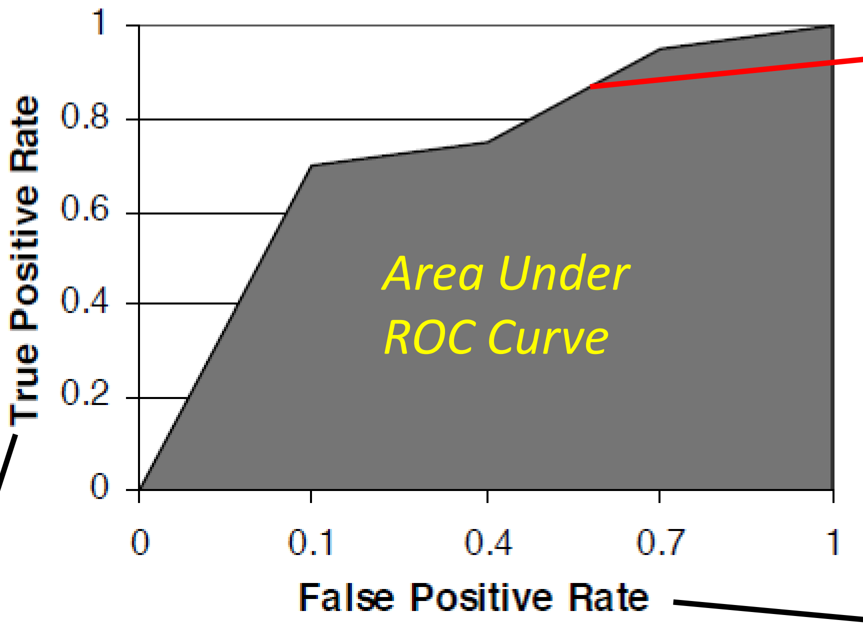
$$\text{micro-}P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} \ ,$$

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

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

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

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

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

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

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

- 代价敏感(cost-sensitive)错误率：

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

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

统计显著性

- 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)

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

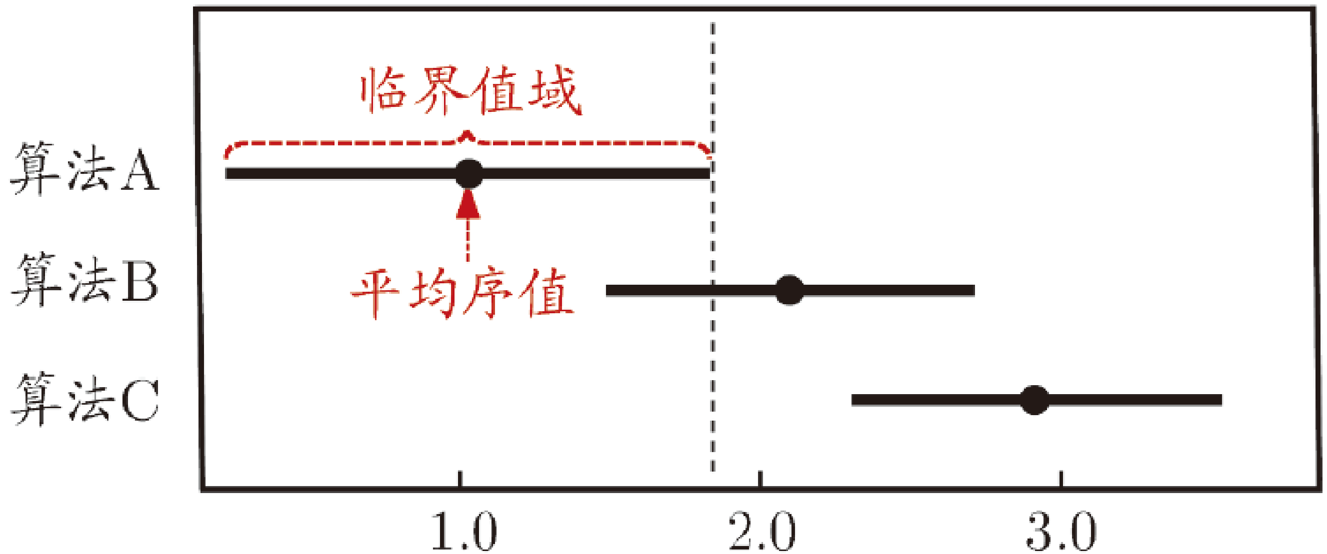


图 2.8 Friedman 检验图

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：

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

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

偏差-方差窘境 (bias-variance dillema)

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

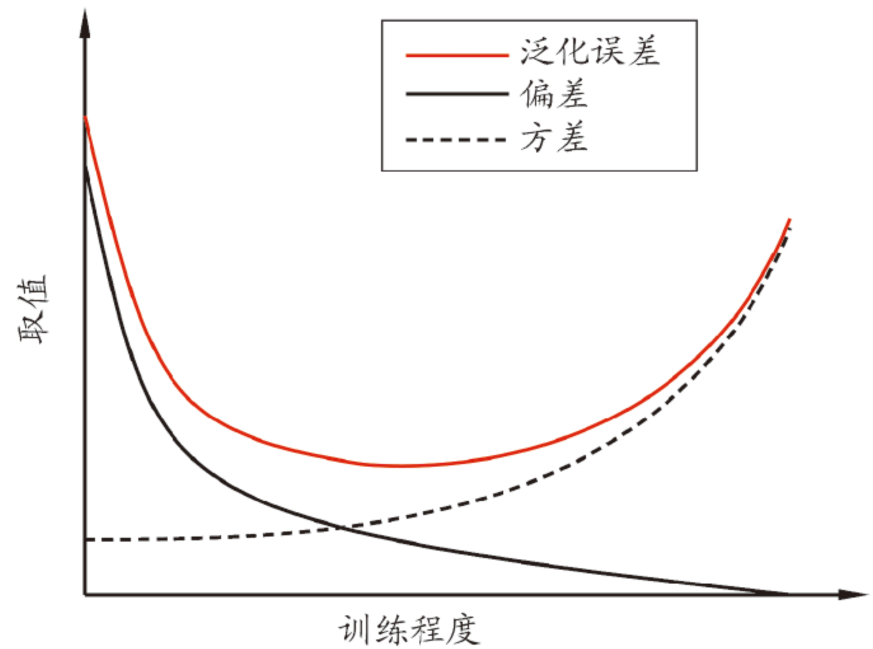


图 2.9 泛化误差与偏差、方差的关系示意图

前往第三站……