南京大学
NANJING UNIVERSITY

人工智能导论

机器学习
(Machine Learning)

郭兰哲

南京大学 智能科学与技术学院

https://www.lamda.nju.edu.cn/guolz

Email: guolz@nju.edu.cn

机器学习（Machine Learning)

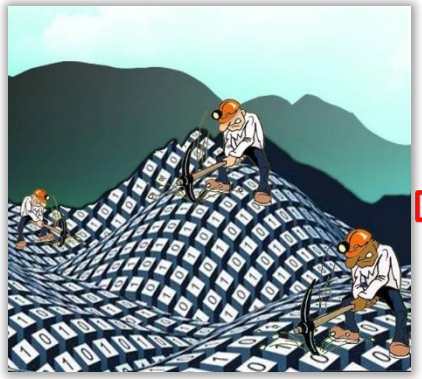
经典定义：利用经验改善系统自身的性能 [T. Mitchell 教科书, 1997]

经验 → 数据

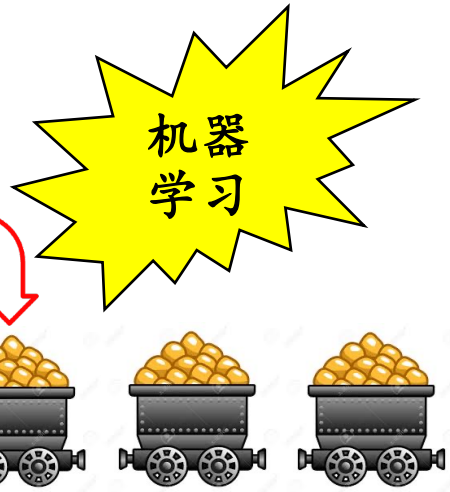
随着该领域的发展，目前主要研究智能数据分析的理论和方法，并已成为智能数据分析技术的源泉之一

大数据时代

大数据≠大价值



智能
数据分析



画作鉴别

画作鉴别(painting authentication)：确定作品的真伪


勃鲁盖尔（1525-1569）的作品？



梵高（1853-1890）的作品？

该工作对专业知识要求极高

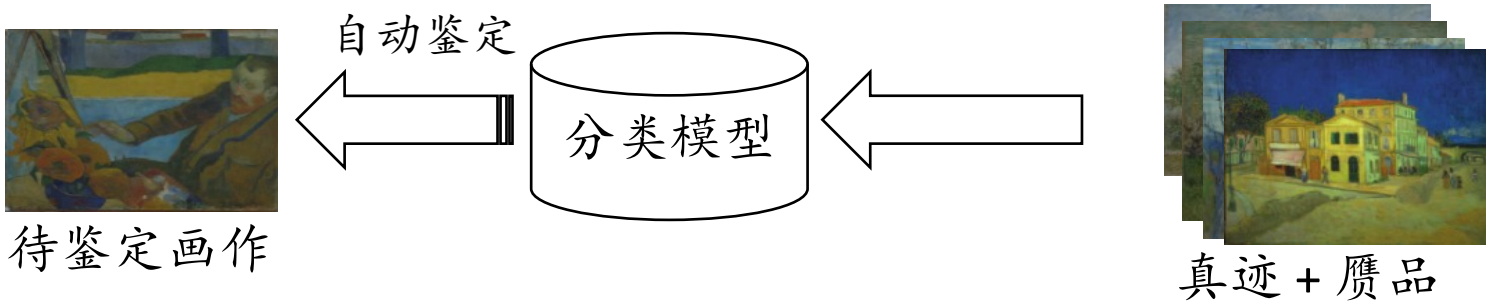
- 具有较高的绘画艺术修养
- 掌握画家的特定绘画习惯

只有少数专家花费很大精力才能完成分析工作！

很难同时掌握不同时期、不同流派多位画家的绘画风格！

画作鉴别

为了降低分析成本，**机器学习**技术被引入



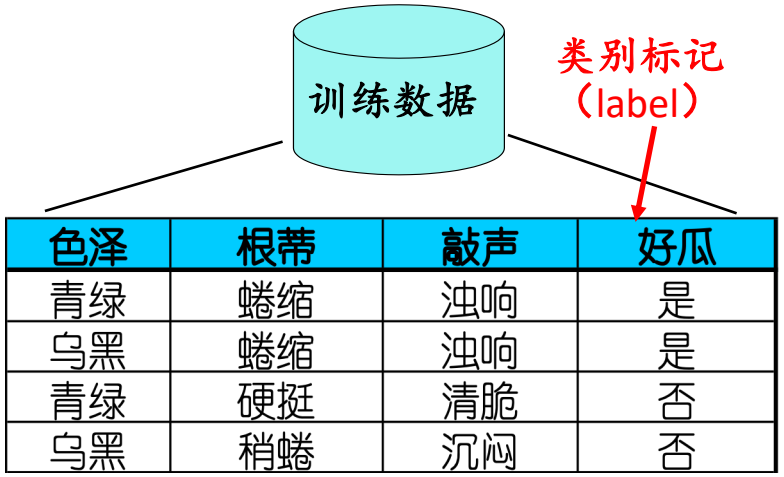
Kröller Müller美术馆与Cornell等大学的学者对82幅梵高真迹和6幅赝品进行分析，自动鉴别精度达 **95%** [C. Johnson et al., 2008]

Dartmouth学院、巴黎高师的学者对8幅勃鲁盖尔真迹和5幅赝品进行分析，自动鉴别精度达 **100%** [J. Hughes et al., 2009][J. Mairal et al., 2012]

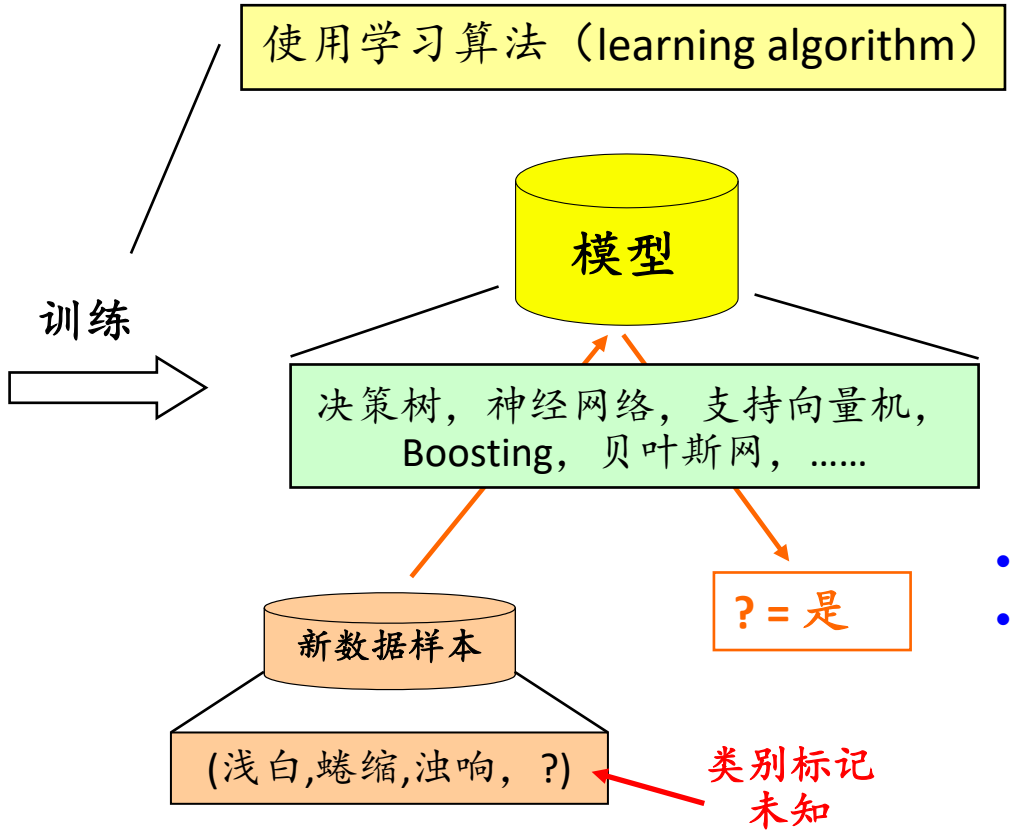
(对用户要求低、准确高效、适用范围广)

典型的机器学习过程

- 监督学习(supervised learning)
- 无监督学习(unsupervised learning)
- 强化学习(reinfocement learning)



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



- 分类, 回归
- 二分类, 多分类

- 数据集：训练集、测试集
- 示例(instance), 样例(example), 样本(sample)
- 属性(attribute), 特征(feature)
- 属性值
- 属性空间, 样本空间, 输入空间
- 特征向量(feature vector)
- 标记空间, 输出空间

- 未见样本(unseen instance)
- 未知“分布”
- 独立同分布(i.i.d.)
- **泛化(generalization)**

假设空间(Hypothesis Space)

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

经验风险最小化（Empirical Risk Minimization）

学习目标：在空间$\mathcal{F}$中寻找能够在整个数据分布上表现最好的模型 f

$$\min_{f\in\mathcal{F}} \mathbb{E}_{(x,y)\sim D}[loss(f(x),y)]$$ 泛化风险

现实任务中，无法得知完整的数据分布，只能获取训练数据

假设所有训练样本都是独立地从这个分布中采样而得

$$\min_{f\in\mathcal{F}} \sum_{i=1}^{n} loss(f(x_i),y_i)$$ 经验风险

泛化风险 vs. 经验风险

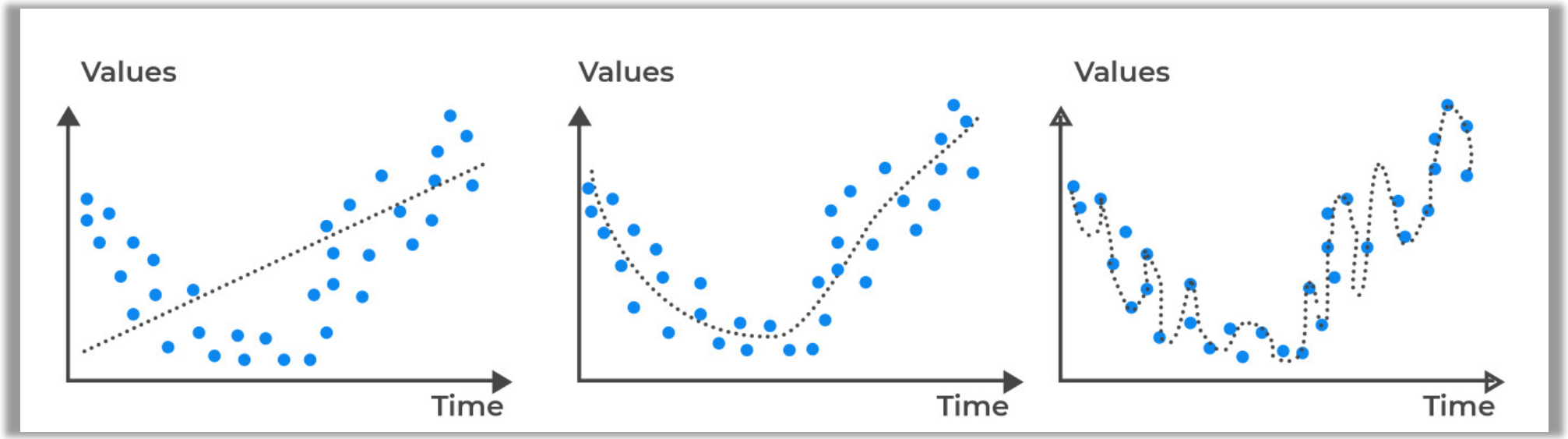
泛化误差：在“未来”样本上的误差

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？

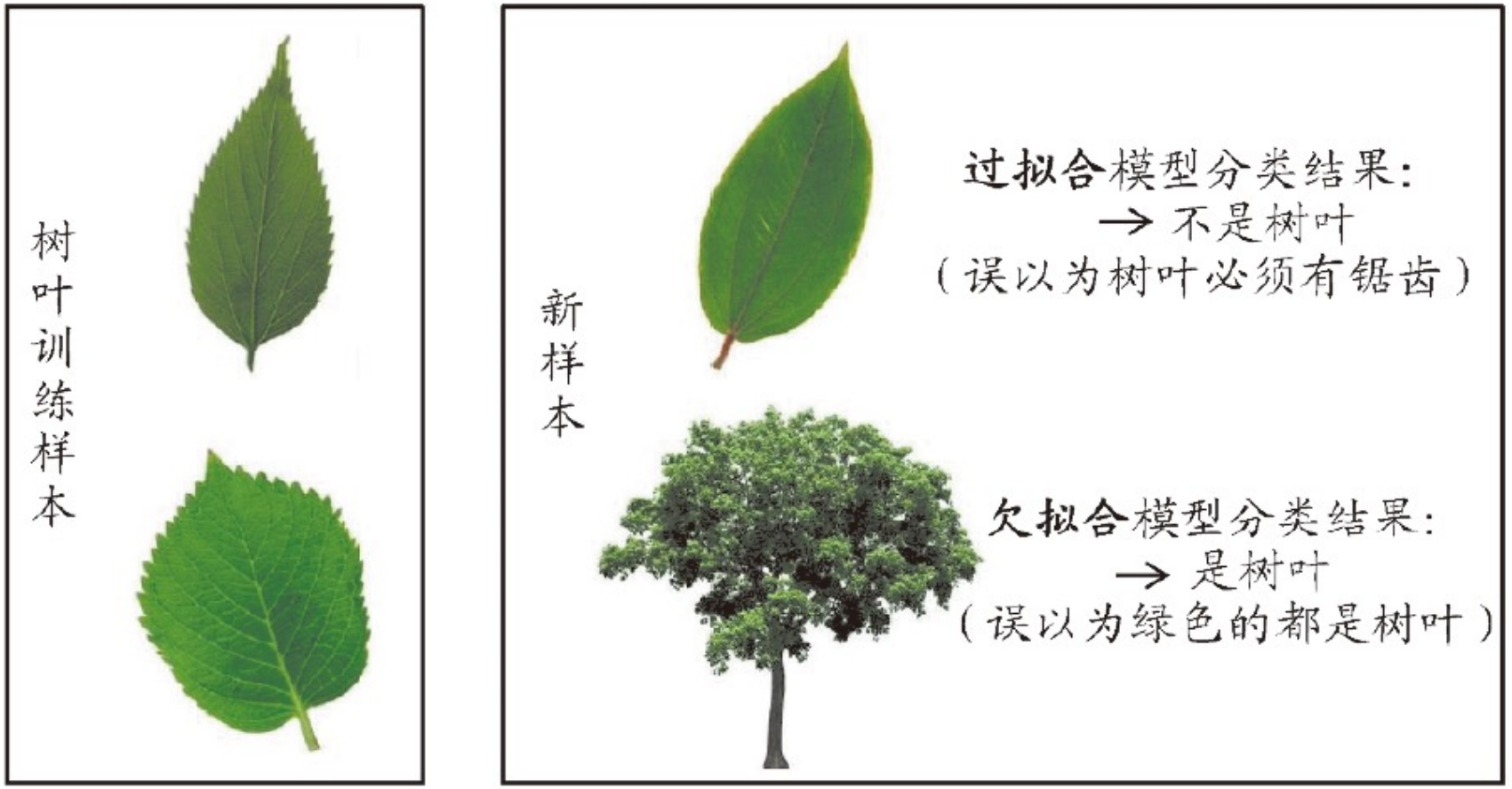
NO！因为会出现“**过拟合**”(overfitting)

过拟合(overfitting) vs.欠拟合(underfitting)



underfitting　　Good fit　　overfitting

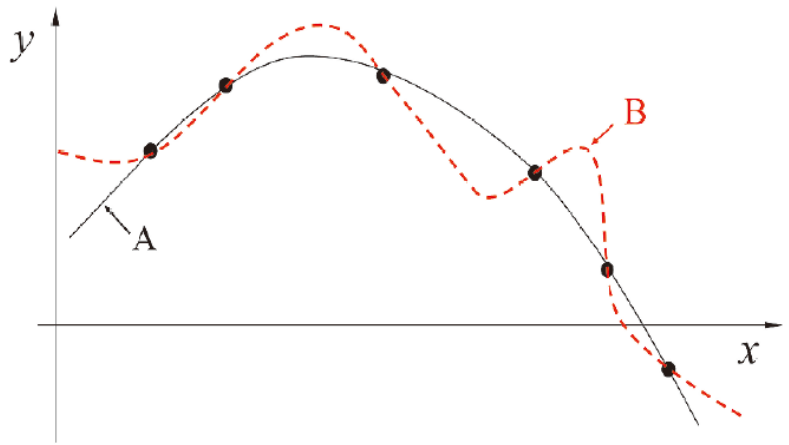
过拟合(overfitting) vs 欠拟合(underfitting)



一般而言，训练样本越少，模型越复杂，越容易过拟合

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？

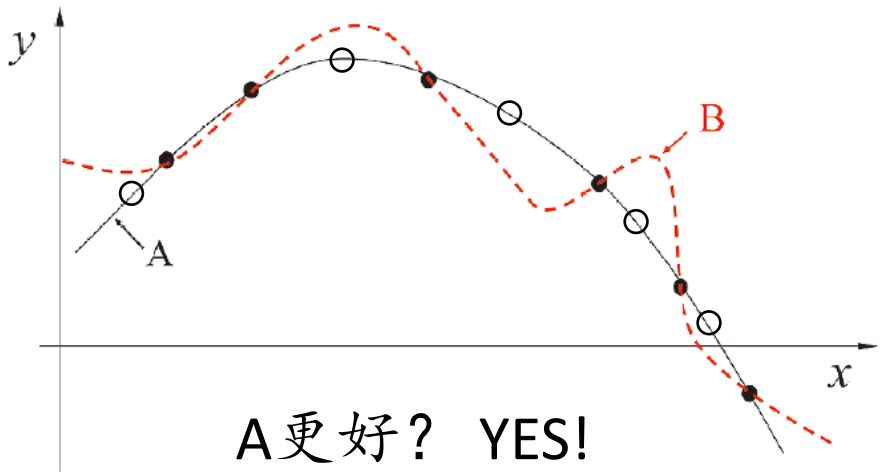
一般原则：
奥卡姆剃刀
(Occam's razor)

任何一个有效的机器学习算法必有其偏好

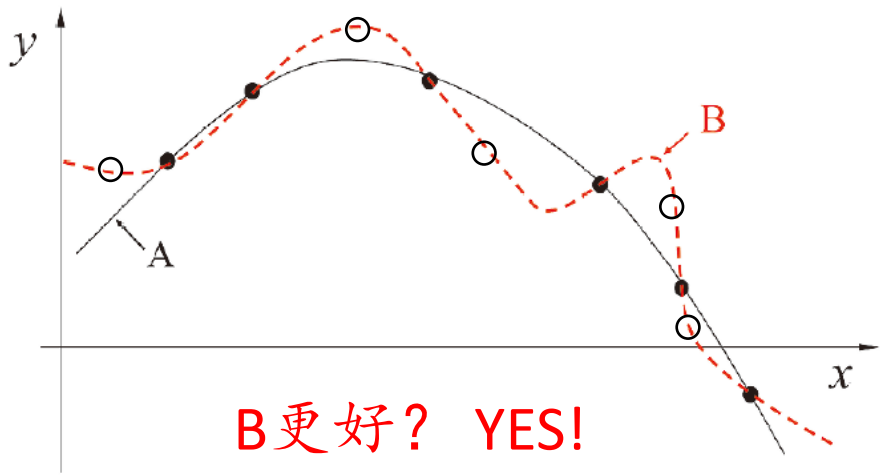
学习算法的归纳偏好是否与问题本身匹配，
大多数时候直接决定了算法能否取得好的性能！

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！

No Free Lunch 定理：一个算法$\mathcal{A}_1$若在某些问题上比另一个算法$\mathcal{A}_2$好，必存在另一些问题，$\mathcal{A}_2$比$\mathcal{A}_1$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**

如何评估与选择模型

先产生若干模型，然后基于某种评估方法进行选择

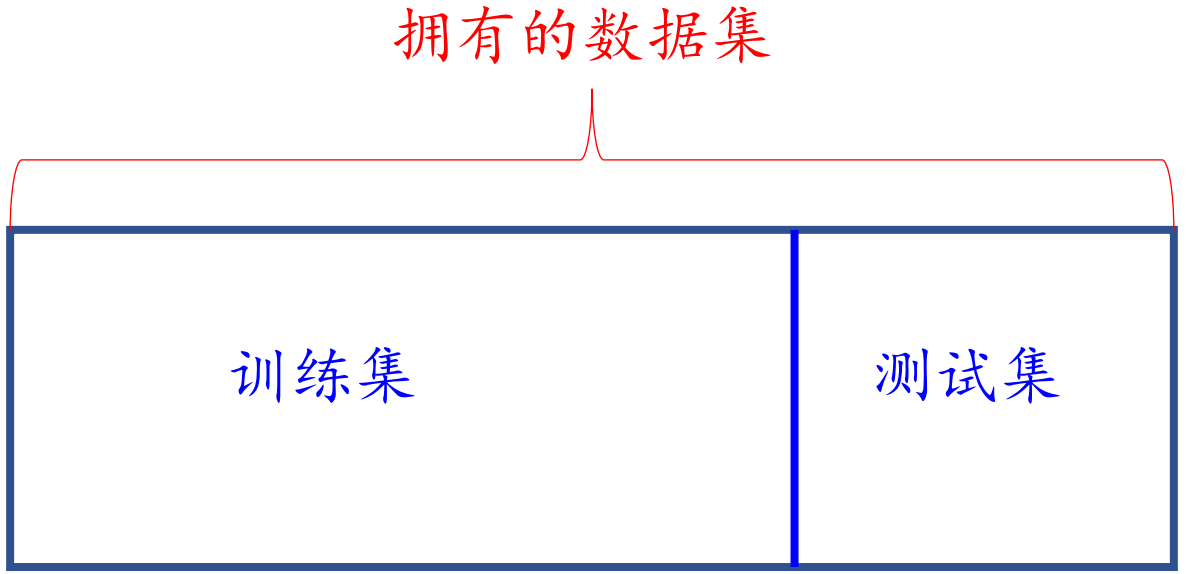
模型选的得好不好对性能往往对最终性能有关键影响

- 如何获得测试结果？

- 如何评估测试结果？

如何获得测试结果

数据划分：
留出法(hold-out)



注意：

- 保持数据分布一致性 （例如: 分层采样)
- 测试集不能太大、不能太小 (例如： 1/5~1/3)
- 多次重复划分，计算均值+方差 (例如: 10次随机划分)

区别:

- 训练集 (training set)
- 测试集 (test set)
- 验证集 (validation set)

如何评估测试结果

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

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：

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

错误率与精度

- 错误率(Error Rate)：

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

- 精度(Accuracy)：

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

查准率与查全率

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)

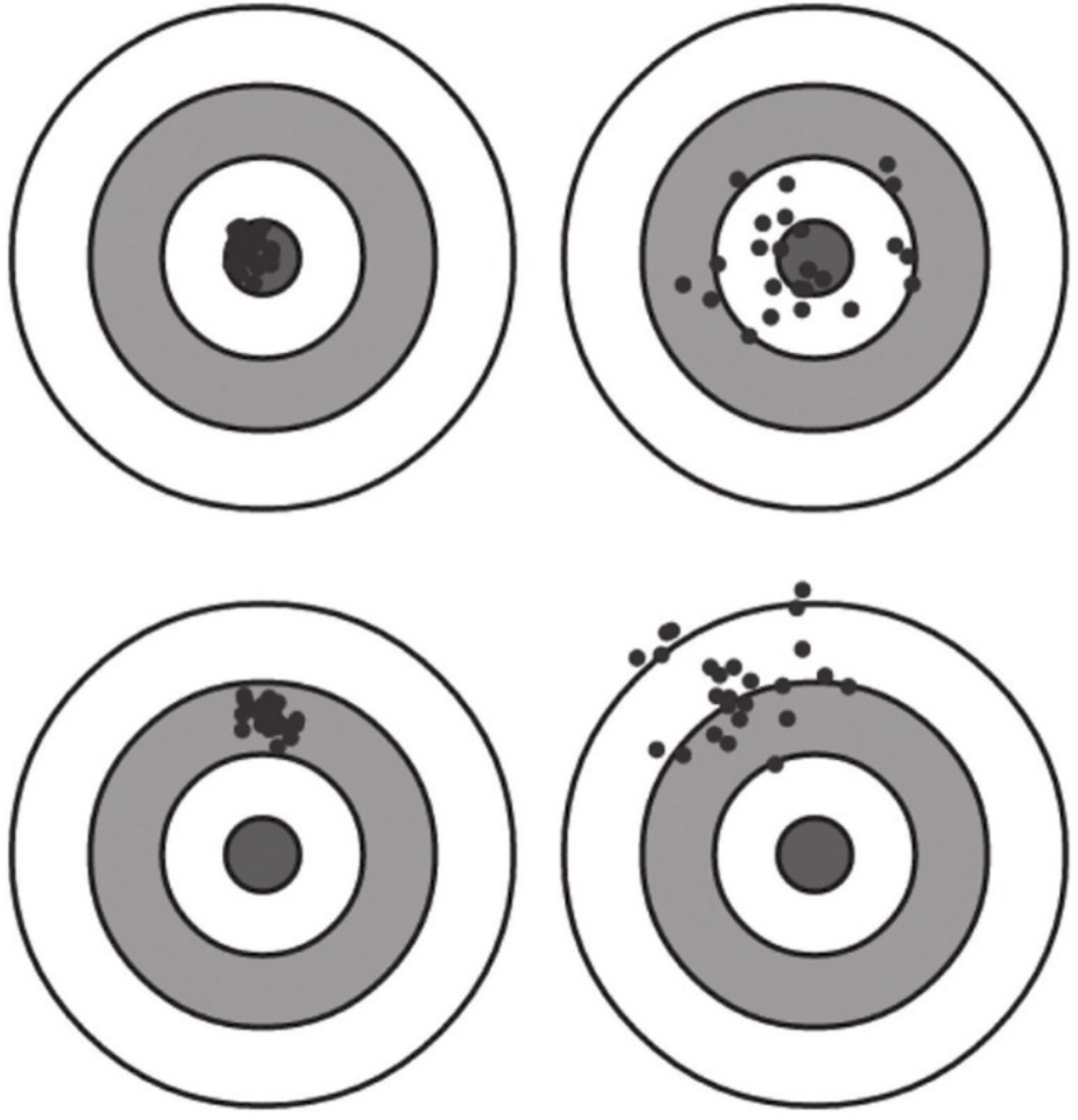
机器学习的误差

“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?

机器学习的误差



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

机器学习是无所不能的吗？

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

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)

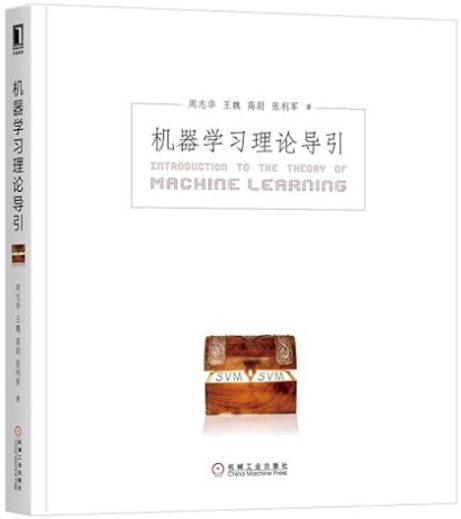
learning model

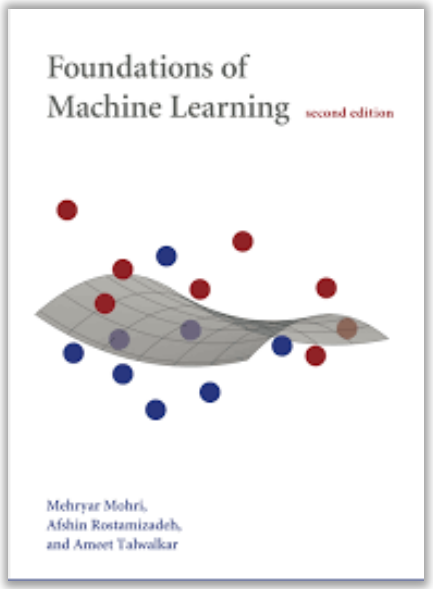
$$P(|f(\boldsymbol{x}) - y| \leq \epsilon) \geq 1 - \delta$$

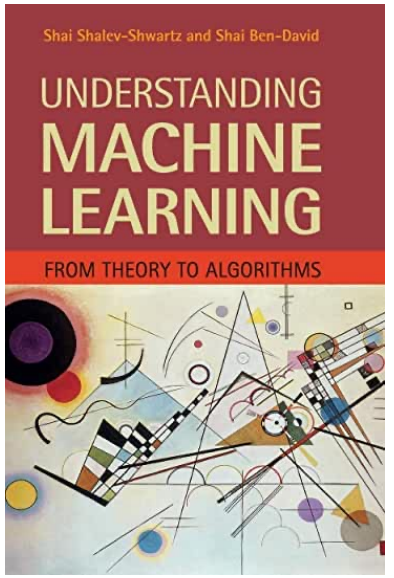
Leslie Valiant
(莱斯利•维利昂特)
(1949-)
2010年图灵奖

机器学习具有坚实的理论基础

- 复杂度理论
- 泛化性理论
- 收敛性理论
- …



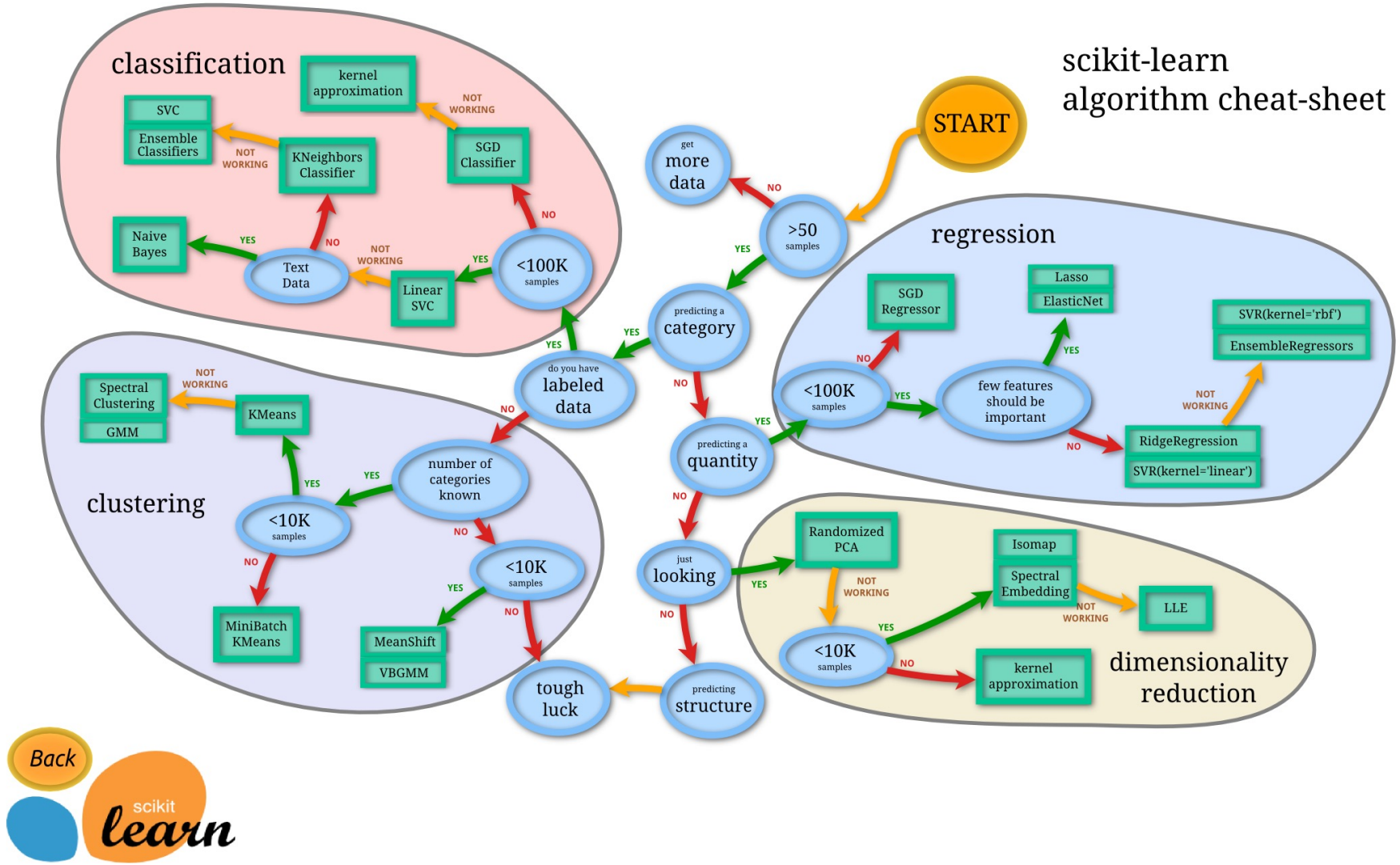




机器学习常用算法

- 线性回归算法 (Linear Regression)
- 逻辑回归算法 (Logistic Regression)
- 支持向量机算法 (Support Vector Machine, SVM)
- k-近邻算法 (K-Nearest Neighbors, KNN)
- k-Means算法
- 决策树算法 (Decision Tree)
- 随机森林算法 (Random Forest)
- 朴素贝叶斯算法 (Naive Bayes)
- 神经网络 (Neural Network)
- …

机器学习常用算法


scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
NO
YES
predicting a category
do you have labeled data
predicting a quantity
just looking
predicting structure
tough luck
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
NO
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
<10K samples
NO
MiniBatch KMeans
<10K samples
YES
MeanShift
VBGMM
NO
regression
SGD Regressor
NO
<100K samples
YES
few features should be important
Lasso
ElasticNet
YES
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
dimensionality reduction
Randomized PCA
YES
NOT WORKING
<10K samples
YES
NO
Isomap
Spectral Embedding
NOT WORKING
LLE
kernel approximation
Back
scikit learn

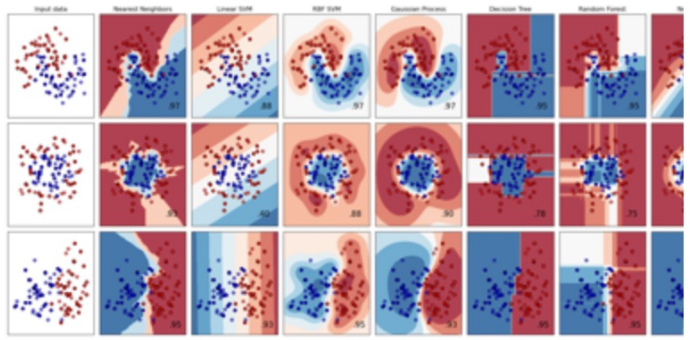
Sklearn

https://scikit-learn.org/

Classification

Identifying which category an object belongs to.

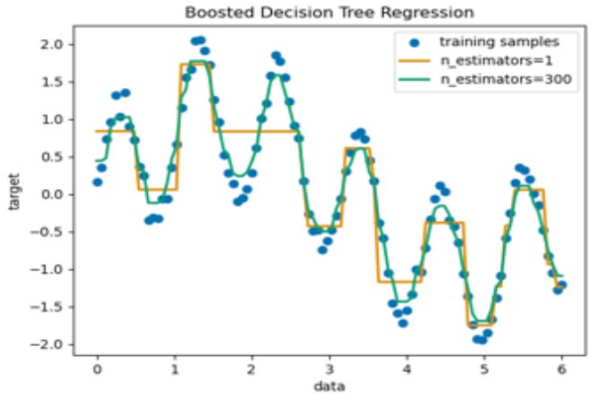
Applications: Spam detection, image recognition.
Algorithms: Gradient boosting, nearest neighbors, random forest, logistic regression, and more...



Regression

Predicting a continuous-valued attribute associated with an object.

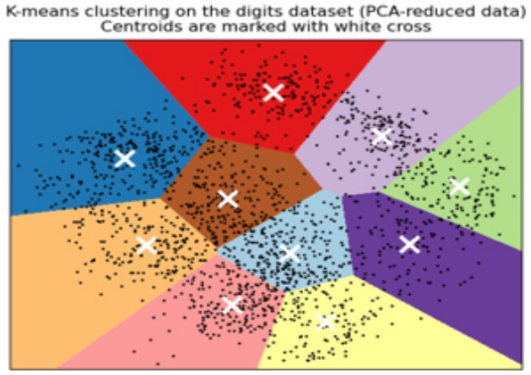
Applications: Drug response, Stock prices.
Algorithms: Gradient boosting, nearest neighbors, random forest, ridge, and more...



Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, HDBSCAN, hierarchical clustering, and more...



示例

训练阶段：

```
>>> import numpy as np
>>> fromsklearn.linear_model import LinearRegression
>>> X = np.array([100], [110], [180]])
>>> y = np.array([300], [330], [540]])
>>> reg = LinearRegression().fit(X, y)
```

测试阶段：

```
>>> reg.predict(np.array([[140]]))
```

示例

```
In [12]: import numpy as np

In [13]: from sklearn.linear_model import LinearRegression

In [14]: X = np.array([[100], [110], [180]])

In [15]: y = np.array([[300], [330], [540]])

In [16]: reg = LinearRegression().fit(X,y)

In [17]: reg.predict(np.array([[140]]))
Out[17]: array([[420.]])

In [18]:
```