R 机器学习高级班（2023 寒假）

第 01 讲　机器学习概论

张敬信

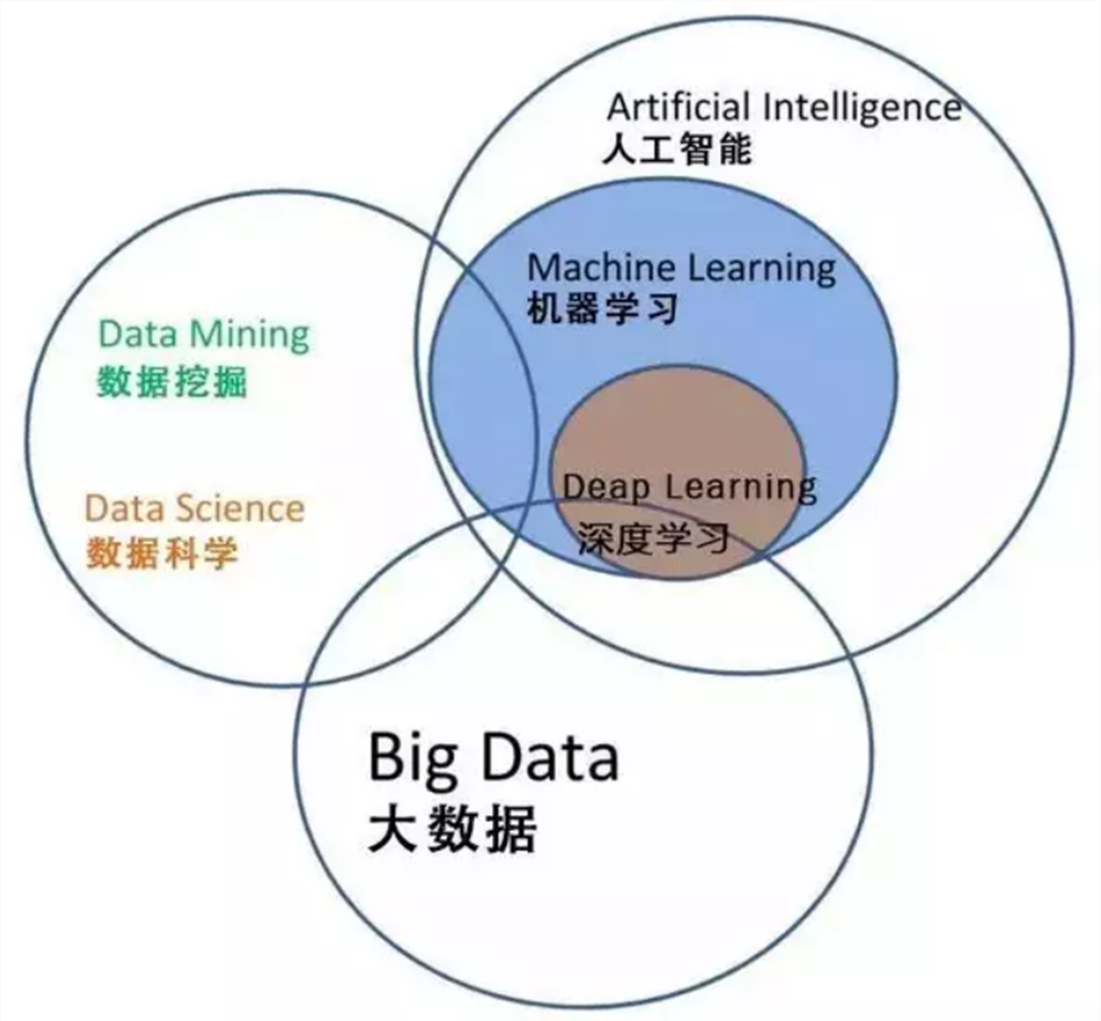
2023 年 1 月 8 日

哈尔滨商业大学

一. 什么是机器学习?

机器学习正在改变我们的世界 (Bernd Bischl, 2022):

- 搜索引擎学习你想要的东西
- 推荐系统学习你对书籍、音乐、电影的品味
- 算法做自动股票交易
- 谷歌翻译学习如何翻译文本
- Siri 学习理解语音
- DeepMind 在围棋方面击败了人类
- 汽车会自动驾驶
- 智能手表监测你的健康
- 选举活动使用算法的目标
- 广告来影响选民
- 数据驱动的发现在物理学、生物学、遗传学、天文学、化学、神经学...


Artificial Intelligence
人工智能
Machine Learning
机器学习
Deap Learning
深度学习
Data Mining
数据挖掘
Data Science
数据科学
Big Data
大数据

人工智能（AI），目前来说只是笼统和”炒作”的术语，泛指当机器被训练用来完成在此之前只能由人类解决的任务或难以解决需要“智能”的任务。

人工智能包括机器学习、自然语言处理、计算机视觉、机器人技术、计划、搜索、游戏、智能代理等。

如今的人工智能，更多的是停留在机器学习、数据挖掘层面。

机器学习（ML），在数学上有明确定义，解决了相当狭窄的任务；机器学习算法通常是从数据中构建预测/决策模型。

> 机器学习（数据挖掘）是一种数据科学技术，可以帮助计算机从现有数据中学习，以便预测未来的行为，结果和趋势。
>
> ——- 微软

深度学习（DL)，是机器学习的专门研究神经网络的一个子领域。

人工神经网络（ANNS）的诞生受到了人脑的启发，ANNs 已经被研究了几十年。DL 使用更多的层，特定的神经元被发明用于图像和张量，许多计算上的改进允许在大数据上进行训练。

DL 可以用于表格数据，但典型的应用是图像、文本或信号。在过去的 10-15 年里，产生了令人瞩目的成果和对人类能力的限制，其中的结果看起来很智能。

机器学习 vs 统计学

- 历史上，ML 和统计学是在不同的领域发展起来的，但许多方法，特别是数学基础是等同的。
- 传统上，ML 的模型更注重精确的预测，而统计学的模型则更注重解释产生数据的模式的能力和进行合理推理的能力。
- 现在，统计学中的 ML 和预测模型基本上用同样的工具解决同样的问题。
- 不幸的是，这些社区仍然是分裂的，没有像他们应该的那样互相交流，而且由于对相同的概念有不同的术语，每个人都感到困惑。
- ML 的大多数部分，也可以称之为：非参数统计 + 高效数值优化。

预测 vs 解释：

- **预测**：我们可能并不关心模型结构是什么样的，或者是否能够理解它。例如：预测股票价格将如何发展。只要能够在新数据上使用预测器，对我们就有直接的好处。
- **解释**：模型只是一种手段，让我们更好地理解数据中的内在关系。例如：了解哪些风险因素会影响患某种疾病的概率。我们可能不会在新的观测中使用学到的模型，而是在科学或社会背景下讨论其影响。

传统上，ML 对前者更感兴趣，而古典统计学则处理后者。在如今的许多任务中，两者都是相关的–在不同程度上。

机器学习一般流程（Liang，梁劲）

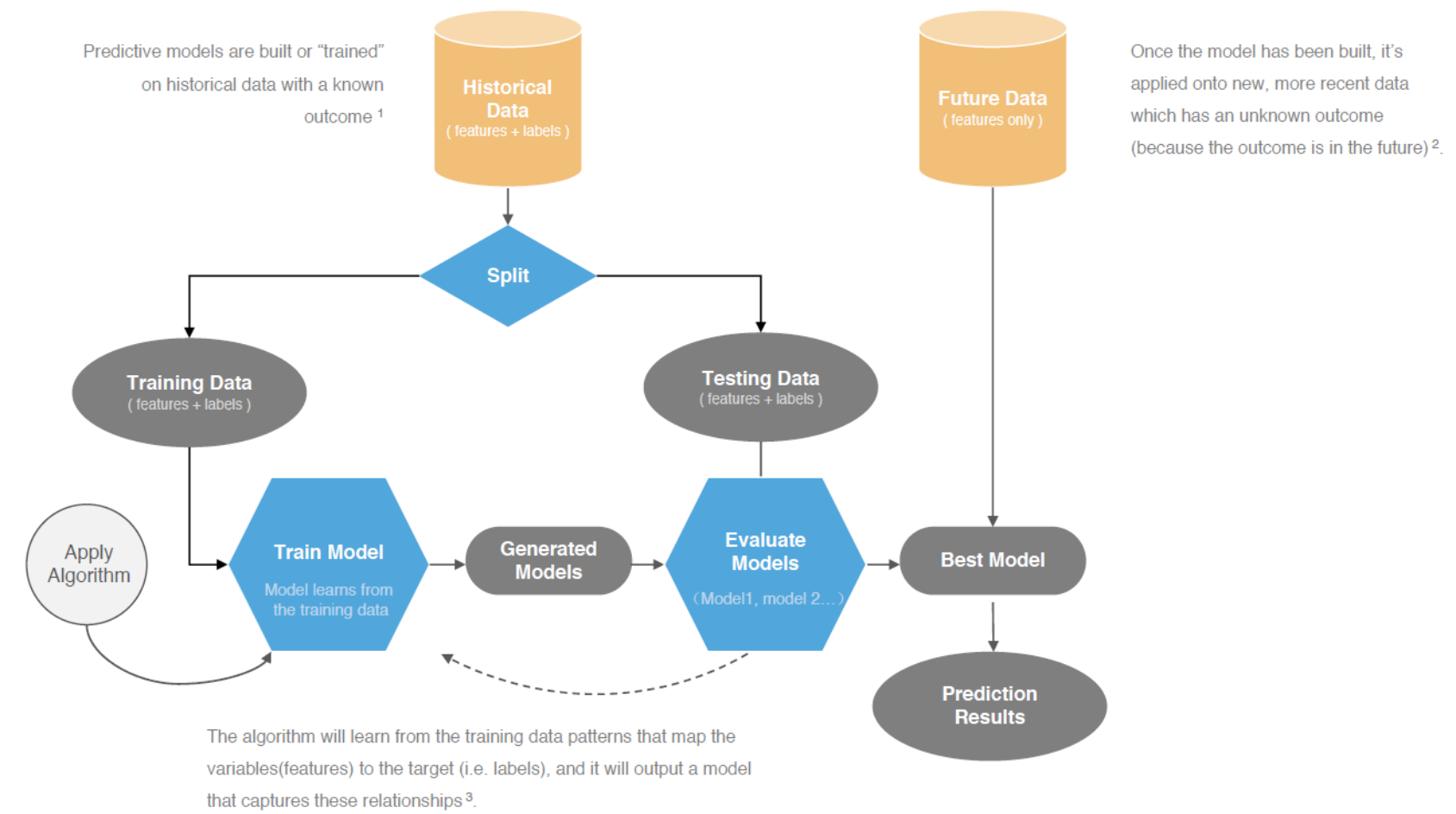


图 1: 机器学习一般流程 [@Liang19]

二. 机器学习类别

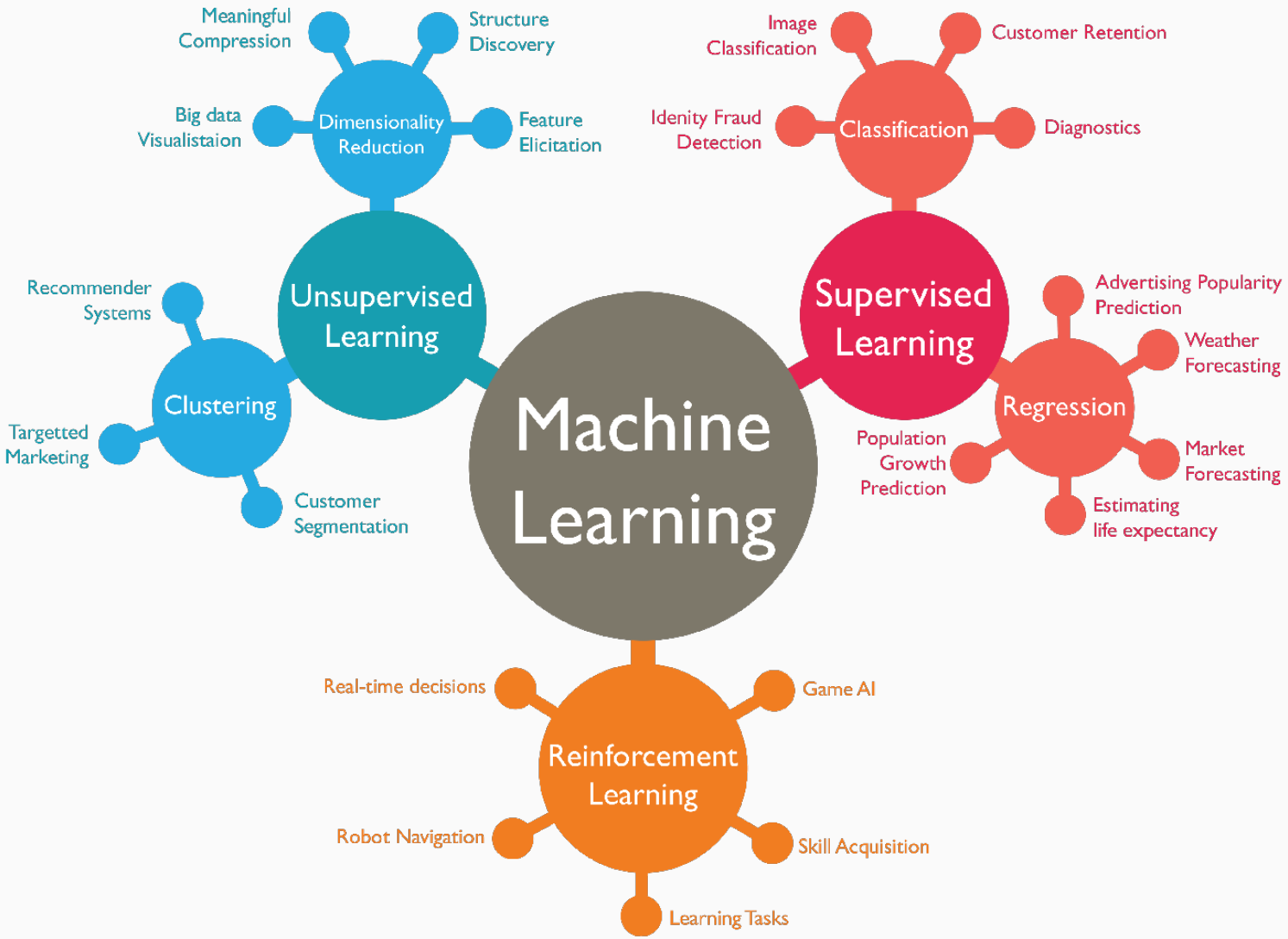


图 2: 机器学习类别

机器学习算法类别（Liang，梁劲）

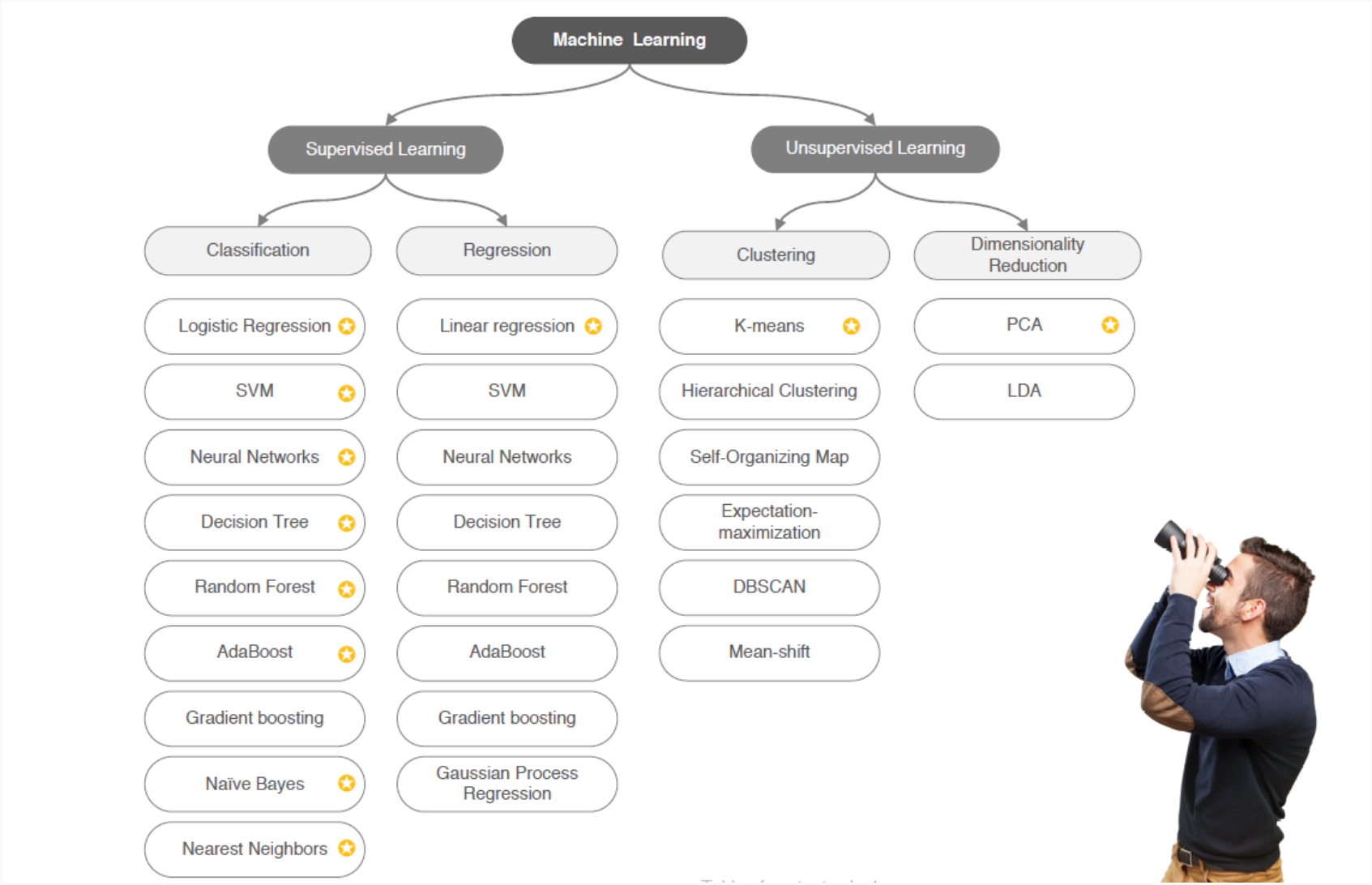
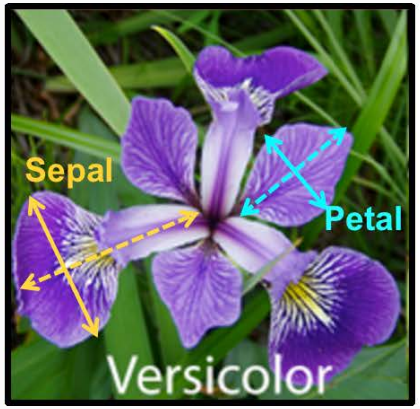


图 3: 机器学习算法类别

1. 有监督学习

数据（Data）：表格数据，通常由若干对象的不同方面的观察值构成。



Features x				Target y
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
4.3	3.0	1.1	0.1	setosa
5.0	3.3	1.4	0.2	setosa
7.7	3.8	6.7	2.2	virginica
5.5	2.5	4.0	1.3	versicolor

- **目标**：因变量/结果变量/预测目标
- **特征**：提供对象的简明描述的可测量属性

特征与目标变量都可以是不同的数据类型：

- 数值型：取值来自 $\mathbb{R}$
- 整数型：取值来自 $\mathbb{Z}$
- 类别型：取值来自有限的类别：$\{C_1, \cdots, C_g\}$
- 二值型：取值来自 $\{0, 1\}$.

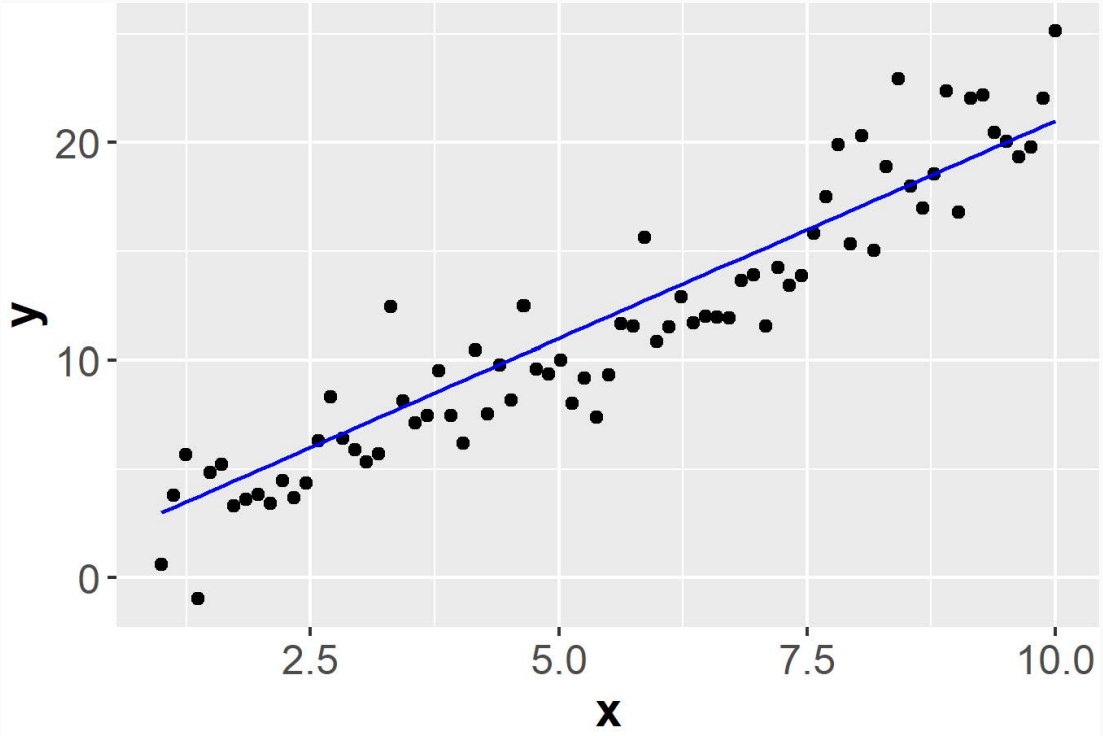
目标变量的不同类型：数值 OR 类别，决定了有监督学习的不同任务：回归 OR 分类。

大多数 ML 算法只能处理数值特征，但也有一些例外（比如，决策树）。对于其他类型的特征，通常要选择或创建合适的编码以把它们转化成数值，比如独热编码、虚拟编码。

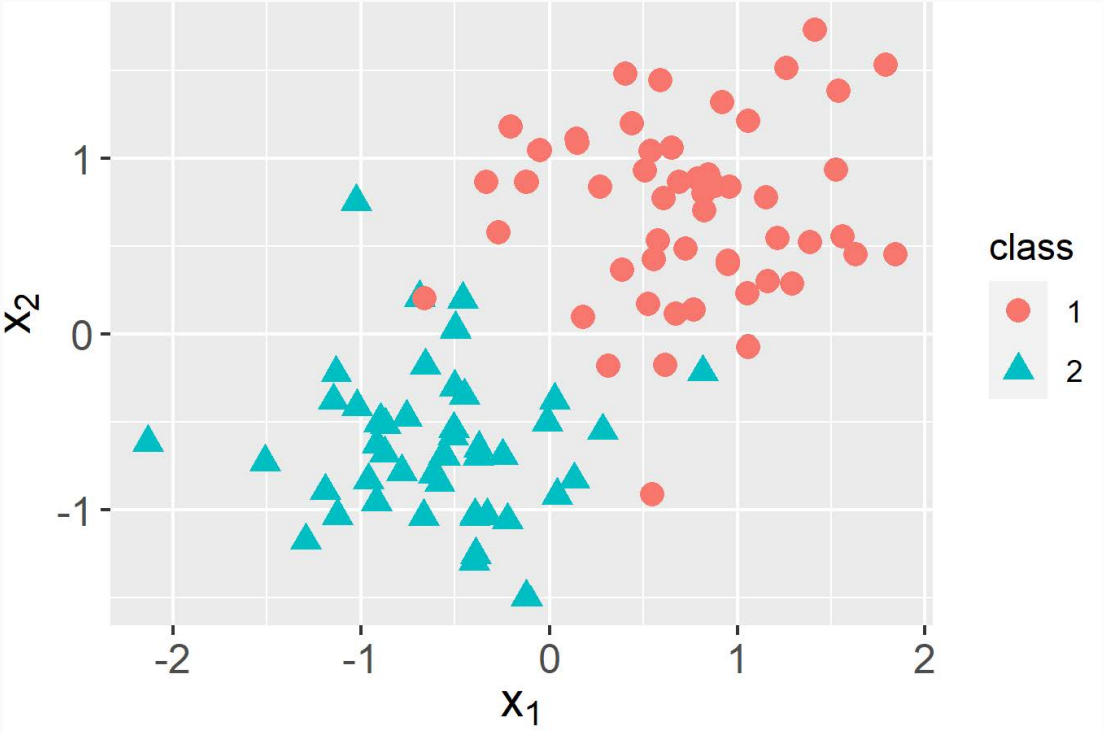
对于类别中具有自然顺序的特征，需要采用反映这种顺序性的编码，例如，整数值的序列。

有监督学习，就是学习输入（特征）和输出（目标）之间函数关系。

回归：因变量是连续型



分类：因变量是类别型



2. 无监督学习

针对没有因变量（结果标签）的数据，用来找出输入数据的模式，主要包括：

- 降维（PCA，自动编码器...），压缩 X 中的信息
- 聚类：将类似的观测聚为一组
- 异常值检测，异常检测
- 关联规则

3. 强化学习（RL，Reinforcement Learning)

强化学习（Reinforcement Learning)，是一个通用的人工智能框架，主要研究作为主体的智能体与作为客体的环境交互的序列决策过程，以及“主体”在环境中逐渐学习到能产生最大的利益的习惯性行为的过程。

目标：选择行动以使未来的回报最大化。

在强化学习中，数据 $=$ (特征, 评价)。以股票交易为例，股票市场是环境，智能体是交易系统，股票价格是状态，而买卖一定数量的股票是行为。交易系统会从一个很简单的交易开始，起初大概率是亏钱的（回报为负)，但是在学习过程中，交易系统与市场不断交互并得到反馈，从而会不断调整策略，越来越强大 (王圣元, 2020)。

4. 迁移学习（Transfer Learning）

迁移学习，就是把已训练好的模型（预训练模型）参数迁移到新的模型来帮助新模型训练。考虑到大部分数据或任务都是存在相关性的，所以通过迁移学习可以将已经学到的模型参数通过某种方式来分享给新模型，从而不用从零学习，以实现加快并优化模型的学习效率。

迁移学习是利用现成的深度学习模型的一种手段：

- **迁移学习**：冻结预训练模型的全部卷积层，只训练自己定制的全连接层。
- **提取特征向量**：先计算出预训练模型的卷积层对所有训练和测试数据的特征向量，然后抛开预训练模型，只训练自己定制的简配版全连接网络。
- **精细调参**：冻结预训练模型的部分卷积层（通常是靠近输入的多数卷积层，因为这些层保留了大量底层信息）甚至不冻结任何网络层，训练剩下的卷积层（通常是靠近输出的部分卷积层）和全连接层。

三. 如何使用数据集?

数据可分为结构化数据和非结构化数据。

结构化数据是可以用二维表结构表示的数据；非结构化数据，是不能用二维表表示的数据，比如图片、文本、语音等。

机器学习通常处理的是结构化数据，而非结构化数据，通常需要深度学习才能处理。

1. 数据集划分

数据集通常先要划分为两部分：**非测试集、测试集。**

- **非测试集**是训练模型用的，用的时候需要再划分为：
  - **训练集：**用来训练模型参数的数据集，模型直接根据训练集来调整自身获得更好的预测效果。
  - **验证集：**用于在训练过程中检验模型的性能状态、收敛情况：
    - 常用于超参数调参，根据几组模型验证集上的表现决定哪组超参数拥有最好的性能
    - 还可用来监控训练过程中模型是否发生过拟合以判断何时停止训练，一般来说验证集表现稳定后，若继续训练，训练集表现还会继续上升，但是验证集会出现不升反降的情况，这样一般就发生了过拟合。
- **测试集：**是测试模型用的，用来评价模型的泛化能力，即之前模型使用验证集确定了超参数，使用训练集调整了模型参数，最后使用一个从没有见过的数据集来判断该模型真正的性能如何。

2. 重抽样

重抽样，就是对数据集重复抽样，得到数据集的若干副本。

机器学习传统的数据划分：训练集 + 测试集，就是对数据的一种重抽样：**留出法**（“holdout”）。

留出法最简单，只得到了数据集的一个副本，所以只能做一次“拟合模型 + 模型预测 + 评估性能”。

一次考试就决定最终成绩，存在偶然性显然是不够科学的，因此有必要从数据集抽样出多个副本，以做多次“拟合模型 + 模型预测 + 评估性能”，取平均性能作为最终成绩。**k 折交叉验证**（“cv”），就是这种重抽样的代表：

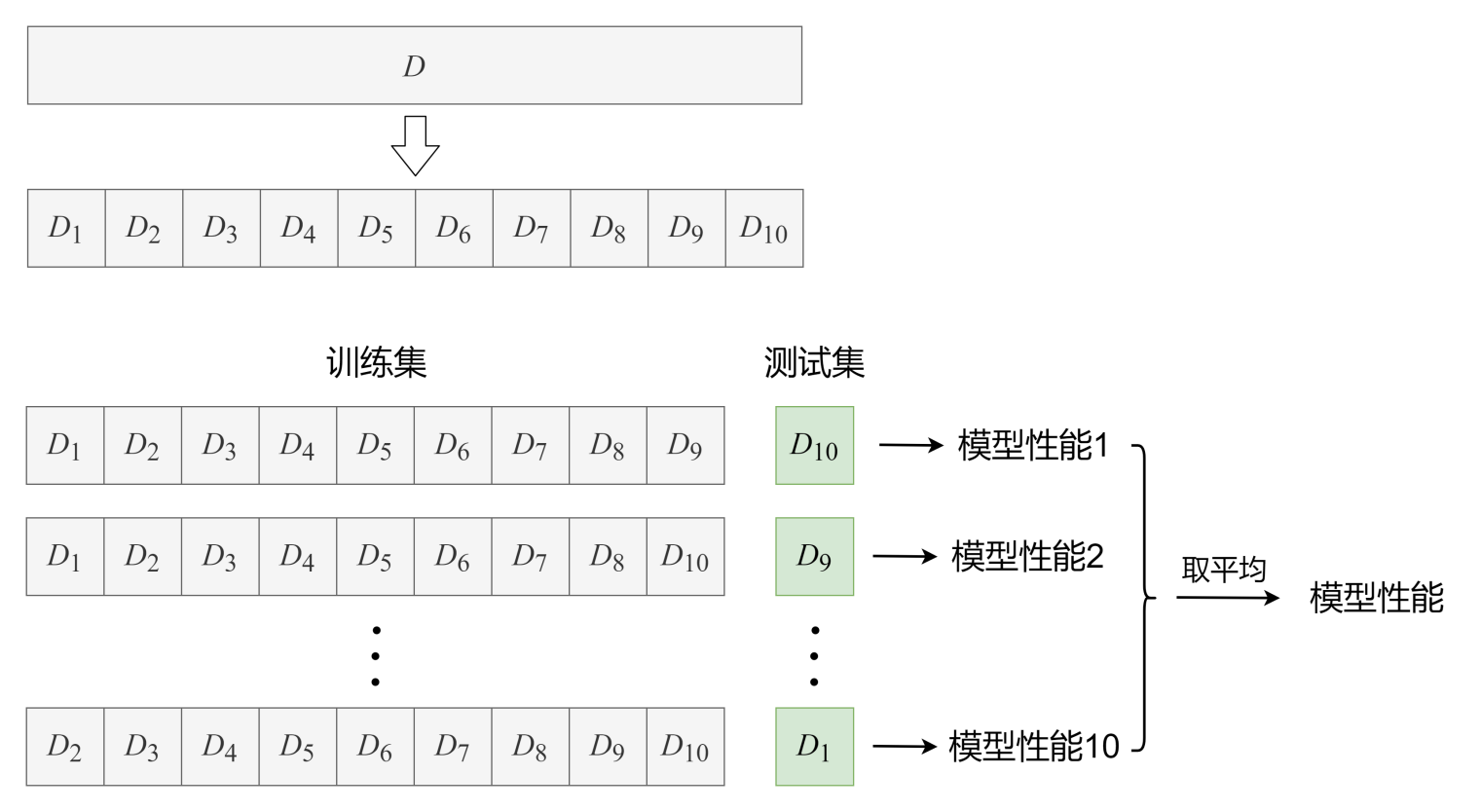


图 4: 10 折交叉验证示意图

k 折交叉验证，还可以重复做 m 次（ × 个副本），叫做**重复 k 折交叉验证**（“repeated_cv”）。

Bootstrap（自助）重抽样（多用于统计学习中计算统计量），是从包含 个观测的原数据集随机可重复抽样 个观测，得到分析数据是有重复的，未被抽取到的观测作为评估数据，也称为“袋外”（out-of-bag）样本。

另外，还有留一交叉验证（“loo”）、子抽样（“subsampling”）、样本内抽样（“insample”）、自定义抽样（“custom”）、自定义交叉验证抽样（“custom_cv”）。

3. 嵌套重抽样 (Marc Becker, 2022)

构建模型，是如何从一组潜在的候选模型（如不同的算法，不同的超参数，不同的特征子集）中选择最佳模型。在构建模型过程中所使用的重抽样划分，不应该原样用来评估最终选择模型的性能。

通过在相同的测试集或相同的 CV 划分上反复评估学习器，测试集的信息会“泄露”到评估中，导致最终的性能估计偏于乐观。

模型构建的所有部分（包括模型选择、预处理）都应该纳入到训练数据的模型寻找过程中。测试集应该只使用一次，测试集只有在模型完全训练好之后才能被使用，例如已确定好了超参数。这样从测试集获得的性能才是真实性能的无偏估计。

对于本身需要重抽样的步骤（如超参数调参），这需要两个嵌套的重抽样循环，即内层调参和外层评估都需要重抽样策略。这就需要**嵌套重抽样**。

嵌套重抽样，即两层重抽样，相当于是两层 `for` 循环：

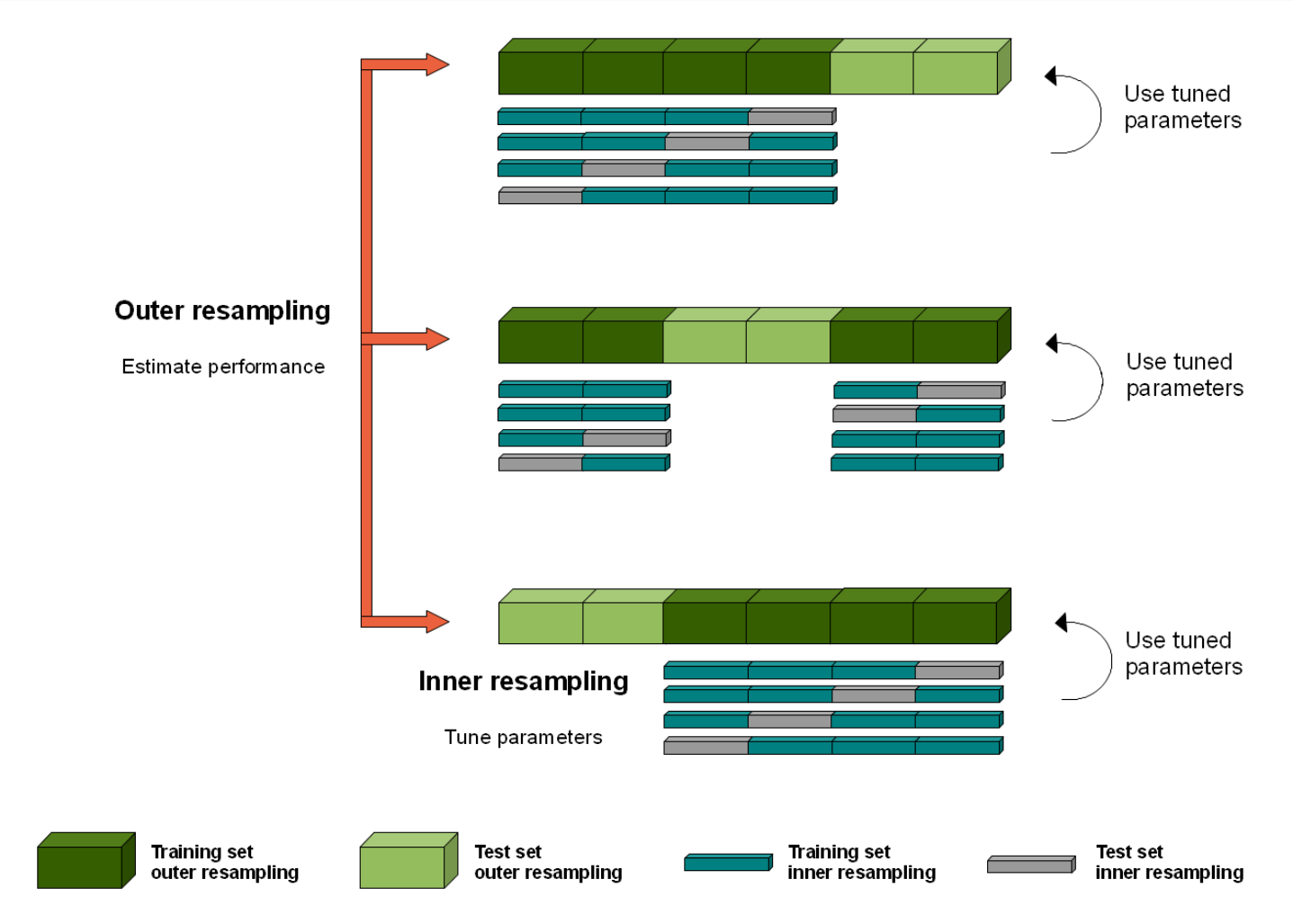


图 5: 嵌套重抽样示意图

外层是对整个数据集重抽样，生成整个数据集的若干副本，每个副本都划分为两部分：**非测试集**和**测试集**，于是就得到若干组非测试集和测试集划分，用于整体上进行外循环的多次迭代：“在非测试集上做特征选择/超参数调参 + 拟合最优特征子集/超参数模型”（也即一轮内循环所做的事情）和“在测试集上评估最优超参数模型性能”，取平均性能作为整个模型的最终性能；

内层是对每一次外循环的非测试集重抽样，生成非测试集的若干副本，每个副本都划分为两部分：训练集和验证集，于是就得到若干组训练集（拟合模型）和验证集（评估模型性能）划分，通常是用于做特征选择/超参数调参的内循环多次迭代，以选出最优的特征子集/超参数，确定该次外循环迭代的最优超参数模型；另外，内循环也可用于监视训练过程是否过拟合。

注 1：外层每次迭代，都是使用内层重抽样选出最优超参数或特征子集，在整个非测试集上重新训练模型，再在测试集上评估模型性能。

注 2：留出（“holdout”）重抽样，只生成数据的 1 个副本，无论用于外层或内层，都相当于只循环迭代 1 次。

只有严格地按照上述嵌套重抽样的方法使用数据，才能避免数据泄露，即

- 测试集上要评估性能的模型，不能提前接触到该测试集；
- 验证集上要评估性能的模型，不能提前接触到该验证集；

这又要在原数据的多个副本上轮转来做，从而才能无偏地估计模型性能。所以，只能这样嵌套重抽样。

四. 模型与训练模型

模型，相当于是一个将特征向量映射到目标值的函数。

函数 f 可以是很简单的（如线性、树结构），也可以非常复杂（如深度神经网络），而且可以有无穷多的选择来构建这样的函数。

如果没有对函数的限制，在所有可用的模型中找到一个”好”的模型的任务就不可能解决。这意味着：必须先验地确定我们模型的类别，从而大大缩小选择范围，称为**结构性先验**。

选择一种 ML 算法 $+$ 固定一组超参数 $\lambda \in \Lambda$，就相当于假定了一种共同的函数结构（模型空间）：

$$\mathcal{H} = \{f_\theta : \ f_\theta \text{ 属于某个确定的以 } \theta \text{ 为参数的函数族}\}$$

这些函数是用来自参数空间 Θ 的参数向量 $\theta = (\theta_1, \theta_2, \cdots, \theta_d)$ 来区分的，一旦确定了最好的一组参数[1]，我们的模型就完全确定了。

这意味着：训练模型时，在模型空间寻找最优模型 $f \in \mathcal{H}$，就等同于在参数空间寻找最优参数值集合 $\theta \in \Theta$。

[1]这里的参数是指模型参数，而不是超参数；训练模型参数，不是超参数调参.

补充：关于超参数调参

机器学习的模型参数是模型的一阶（直接）参数，是训练模型时用梯度下降法寻优的参数，比如正则化回归模型的回归系数；而超参数是模型的二阶参数，需要事先设定为某值，才能开始训练一阶模型参数，比如正则化回归模型的惩罚参数、KNN 的邻居数等。

超参数会对所训练模型的性能产生重大影响，所以不能是随便或凭经验随便指定，而是需要设定很多种备选配置，从中选出让模型性能最优的超参数配置，这就是**超参数调参**。

超参数调参是一项多方联动的系统工作，需要设定：搜索空间、学习器、任务、重抽样策略、模型性能度量指标、终止条件。

有监督学习由如下 3 部分构成：

$$\text{学习} = \text{假设空间} + \text{损失} \ (+ \ \text{正则项}) + \text{优化}$$

- **假设空间**：定义（和限制！）从数据中可以学到什么样的模型；
- **损失（+ 正则项）**：量化一个特定模型在一个给定数据集上的表现。这使我们能够对候选模型进行排序，以选择最佳模型；
- **优化**：定义了如何在假设空间中寻找最佳模型，即损失最小的模型。

训练过程的数学表示：

$$\mathcal{I}：\mathcal{D}\times\Lambda\to\mathcal{H},\qquad(\mathcal{D},\lambda)\to\hat{f}_{\mathcal{D},\lambda}$$

有监督学习就是从训练数据和某种超参数设定下，迭代训练直到找到最优模型的过程：

(1) 选定一种模型算法

(2) 提供训练数据，设定一组超参数

(3) 寻找最优的模型参数，让模型尽可能地匹配数据

- 先随机生成一组模型参数，这样模型就能确定
- 将训练数据的输入带入模型，就能得到预测值
- 选择合适的损失函数，度量预测值与真实值相差多少
- 迭代训练：用梯度下降法或其变种，改进模型参数，逐步减小损失
- 在终止条件下结束训练

(4) 保存此时的（最优）模型或模型参数供后续使用

2. 经验风险函数（内部损失）

经验风险函数，用来评价模型预测值与真实值的不一致程度，通常用 $L(\theta)$ 来表示。该函数是一个非负实值函数，值越小表示针对训练数据的模型性能越好。

另外，为了避免让模型过于复杂，经常给损失函数再加上一个正则项，对模型复杂化进行惩罚。

有监督学习过程可归结为如下的损失函数 $+$ 正则项的优化问题：

$$\mathcal{L}(\theta) = \min_{\theta \in \Theta} \sum_{i=1}^{n} L(y^{(i)}, f(\mathbf{x}^{(i)}|\theta) + \lambda J(f)$$

注：内部损失，往往要求函数是处处可导的。

2. 梯度下降法（GD）

- θ 为模型参数，机器学习通常都是采用梯度下降法让损失函数 $L(\theta)$ 达到最小，从而找到最优的模型参数 θ.

梯度下降法是广泛用于机器学习，其核心思想是迭代地调整参数，使得损失函数达到最小值。

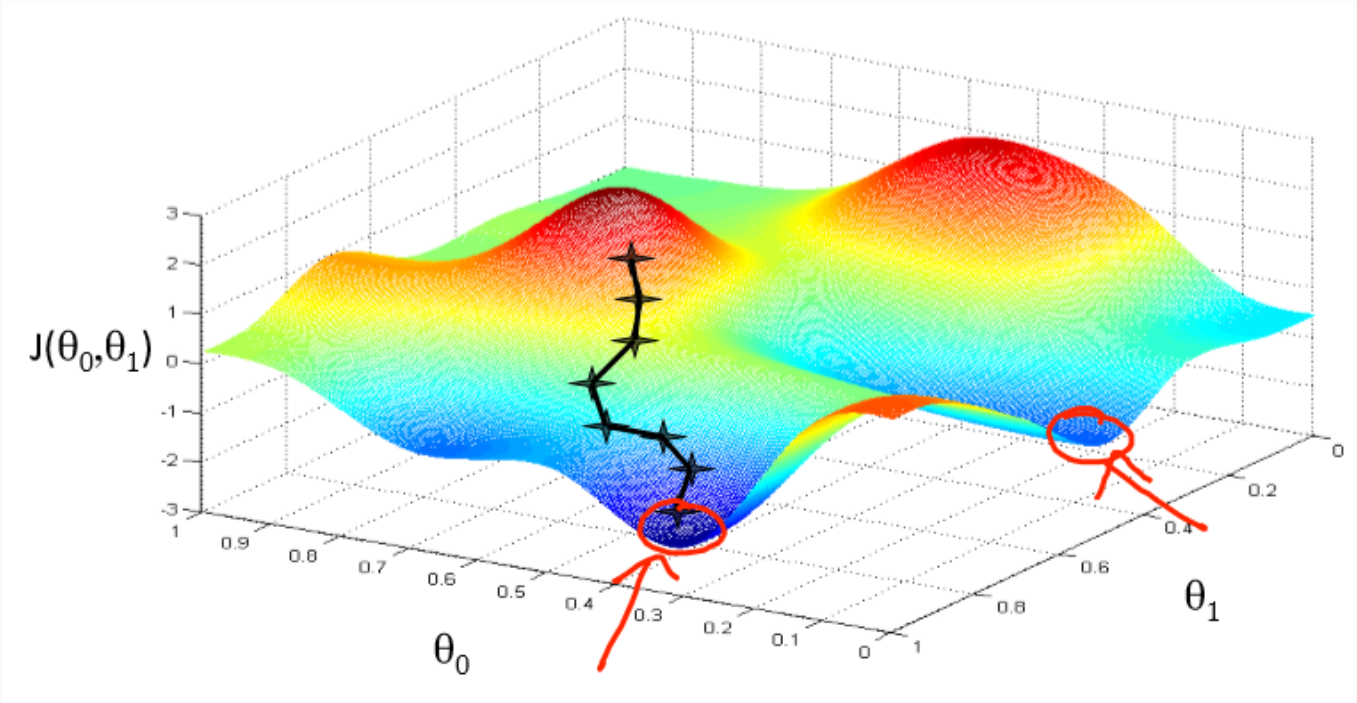
- 比如线性回归的经验风险函数（损失函数）:

$$\mathcal{L}(\theta) = \|\mathbf{X}\theta - \mathbf{y}\|_2^2 = \sum_{i=1}^{n} \left(\theta^T \mathbf{x}^{(i)} - y^{(i)}\right)^2$$

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\, \mathcal{L}(\theta)$$

梯度下降法，就好比在浓雾笼罩的山上下山，每次只能看到前方一步远，那么就 360° 每个方向迈一步的话下降的最多，那就往哪个方向迈一步，重复该过程，逐步到达较低点（不一定是最低点）。

根据数学知识，一步下降最快的方向就是梯度方向！



具体来说，GD 法是从当前的候选参数 $\theta^{[t]}$，沿负梯度方向（即最陡峭的下降方向），以学习率 α 迭代到下一个 $\theta^{[t+1]}$：

$$\theta^{[t+1]} = \theta^{[t]} - \alpha \frac{\partial}{\partial \theta} \mathcal{L}(\theta^{[t]})$$

- GD 是一阶方法（只用到一阶梯度）。二阶方法牛顿法使用 `Hessian` 矩阵来细化搜索方向，以加快收敛速度。
- GD 有许多改进版本，比如灵活控制学习率、摆脱鞍点，模仿二阶行为而不计算 `Hessian` 矩阵（拟牛顿法）。
- 梯度下降法每一步梯度向量的计算，都是基于整个训练集，若改为每次随机选择的子集，即为随机梯度下降（SGD）。对于大数据问题计算效率更高。

四. 评估模型性能（外部损失）

训练集上训练好的模型的未来性能如何，需要这样来检验：

- 将模型用到测试集（不能与训练集、验证集有交集，否则会有数据泄露）得到预测值；
- 选择一种合适的性能度量指标，来度量预测值与真实值平均相差多少。

$$\widehat{GE}(\hat{f}, L) = \frac{1}{m} \sum_{(\mathbf{x}, y) \in \mathcal{D}_{\text{test}}} \left[L(y, \hat{f}(\mathbf{x})) \right]$$

1. 回归度量

- 均方误差（MSE, L2 损失)

$$\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}-\hat{y}^{(i)}\right)^2$$

- 均方根误差（RMSE)

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)}-\hat{y}^{(i)}\right)^2}$$

- 平均绝对误差（MAE）

$$\frac{1}{m}\sum_{i=1}^{m}\left|y^{(i)}-\hat{y}^{(i)}\right|$$

- 平均相对误差（MAPE）

$$\frac{1}{m}\sum_{i=1}^{m}\left|\frac{y^{(i)}-\hat{y}^{(i)}}{y^{(i)}}\right|$$

- 可决系数 R^2：反映了模型所能解释的方差占总方差的比重

$$\mathrm{SST}=\sum_{i=1}^{n}(y_i-\bar{y})^2,\quad \mathrm{SSR}=\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2,\quad \mathrm{SSE}=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

$$R^2=\frac{\mathrm{SSR}}{\mathrm{SST}}=1-\frac{\mathrm{SSE}}{\mathrm{SST}}$$

注意： R^2 越大，不代表模型拟合的越好（**模拟：** $y=1.1x+\varepsilon$, 其中 $\varepsilon\in N(0,0.15)$, 比较用一半数据和用全部数据）。

经典统计度量，如 R^2、似然、AIC、BIC、偏差都是基于训练误差，对于性能有限、数据充足、不违反分布假设的模型，它们可能会起作用。对高维、复杂的机器学习数据不一定有效。

- ML 用的一般化 R^2:

$$1 - \frac{Loss_{\text{复杂模型}}}{Loss_{\text{简单模型}}}$$

- **错误率 (misclassification error rate, MCE):**

$$\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} \neq \hat{y}^{(i)}]$$

- **准确率 (Accuracy, ACC):**

$$\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} = \hat{y}^{(i)}]$$

对数损失

- 二分类（交叉熵损失）：

$$\frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\ln\hat{\pi}^{(i)}-(1-y^{(i)})\ln\left(1-\hat{\pi}^{(i)}\right)\right]$$

其中，$y^{(i)}$ 为样本真实类别，$\hat{\pi}^{(i)}=\Pr\{y=1|\mathbf{x}^{(i)}\}$ 为第 i 个数据点预测为正例的概率。

- 多分类（对数似然损失）：

$$-\frac{1}{m}\sum_{i=1}^{m}o_k^{(i)}\ln\hat{\pi}_k^{(i)}$$

其中，$o_k^{(i)}$ 为第 i 个样本虚拟变量表示的真实类别，$\hat{\pi}_k^{(i)}$ 为第 i 个样本预测属于每个类别的概率向量。

Brier 评分

相当于回归度量 MSE 推广到概率情形。

- 二分类：

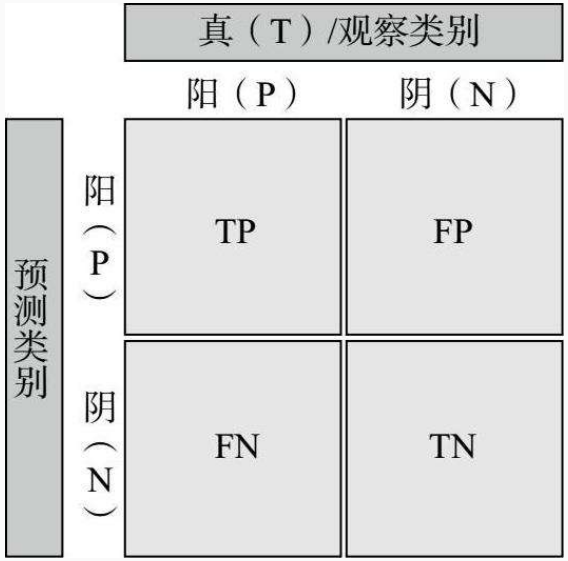
$$\frac{1}{m}\sum_{i=1}^{m}\left(\hat{\pi}^{(i)}-y^{(i)}\right)^2$$

- 多分类：

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{g}\left(\hat{\pi}_k^{(i)}-o_k^{(i)}\right)^2$$

常用二分类度量

- **混淆矩阵（Confusion Matrix）：**



- TP（True Positive，真正）：将正类预测为正类数
- TN（True Negative，真负）：将负类预测为负类数
- FP（False Positive，假正）：将负类预测为正类数误报（Type I error）
- FN（False Negative，假负）：将正类预测为负类数–> 漏报（Type II error）

- **查准率（Precision）**：表示被分为正例的示例中实际为正例的比例

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

- **召回率（Recall）**：是覆盖面的度量，度量有多少个正例被正确地分为正例

$$Recall = \frac{TP}{TP + FN}$$

- **F1 得分（F-Score）**：查准率和召回率的加权组合，均衡查准率和召回率的结果

$$F1 = 2 \cdot \frac{Precision * Recall}{Precision + Recall}$$

举例说明 (知乎 NaNNN, 2022)，想评估某医院新开发的癌症 AI 诊断系统，病人得了癌症定义为 Positive，没得癌症定义为 Negative。

- **Accuracy** 回答的是：**在一堆癌症病人和正常人中，有多少人被系统给出了正确诊断结果（患癌或没患癌)?**
- **Precision** 回答的是：**在诊断为癌症的一堆人中，到底有多少人真得了癌症?**
- **Recall** 回答的是：**在一堆得了癌症的病人中，到底有多少人能被成功检测出癌症?**

那么，什么时候应该更注重 Recall 而不是 Precision 呢?（**癌症诊断系统**）

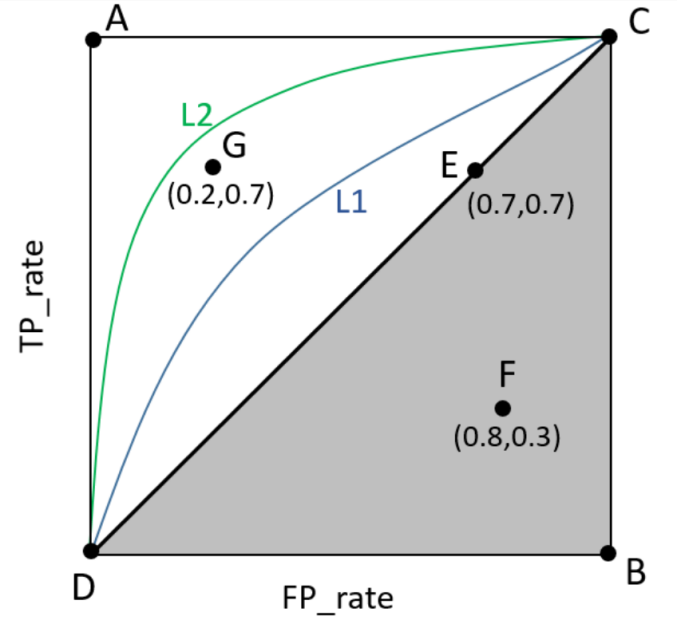
- 当 False Negative (FN) 的成本代价很高（后果很严重)，希望尽量避免产生 FN 时，应该着重考虑提高 Recall 指标。上例中 FN 是得了癌症的病人没有被诊断出癌症，这种情况是最应该避免的。宁可把健康人误诊为癌症 (FP)，也不能让真正患病的人检测不出癌症 (FN) 而耽误治疗离世。

什么时候应该更注重 Precision 而不是 Recall 呢？（**垃圾邮件屏蔽系统**）

- 当 False Positive (FP) 的成本代价很高（后果很严重）时，即期望尽量避免产生 FP 时，应该着重考虑提高 Precision 指标。比如，垃圾邮件屏蔽系统，垃圾邮件为 Positive，正常邮件为 Negative，FP 是把正常邮件识别为垃圾邮件，这种情况是最应该避免的。宁可把垃圾邮件标记为正常邮件 (FN)，也不能让正常邮件直接进垃圾箱 (FP)。

而 F1 得分是 Precision 和 Recall 两者的综合。以给犯人定罪为例，有罪 (Positive)，无罪（Negative)。Recall 即「天网恢恢，疏而不漏，任何罪犯都插翅难飞」；Precision 即「绝不冤枉一个好人，但是难免有罪犯成为漏网之鱼，逍遥法外」。如果想二者折中为好，那就 F1 得分。

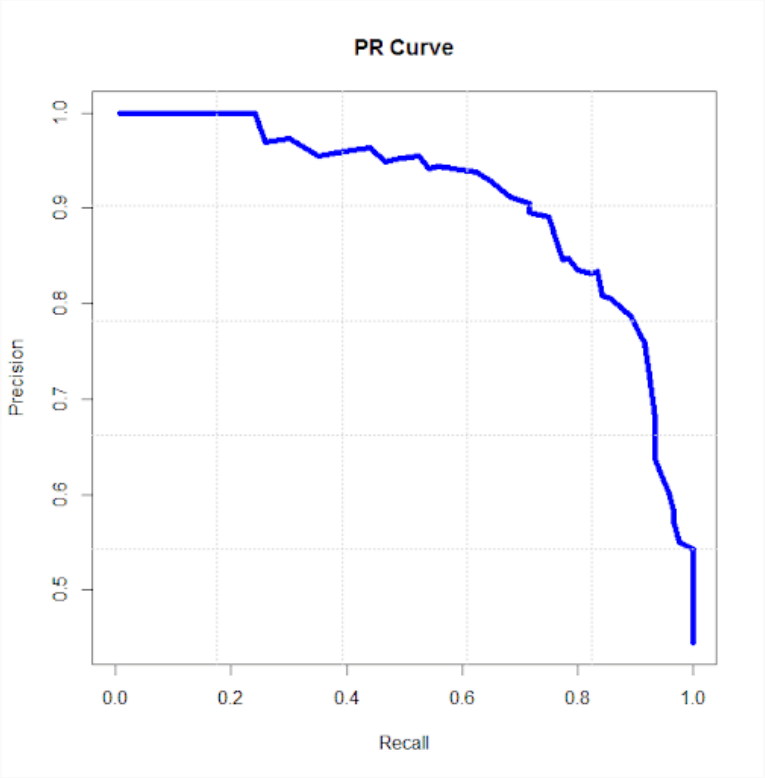
- **ROC 曲线**，是在不同分类阈值上对比真正率（TP_rate）与假正率（FP_rate）的曲线，ROC 曲线下面的面积叫做 **AUC**：



$$TP_{rate} = \frac{TP}{TP + FN}, \qquad FP_{rate} = \frac{FP}{FP + TN}$$

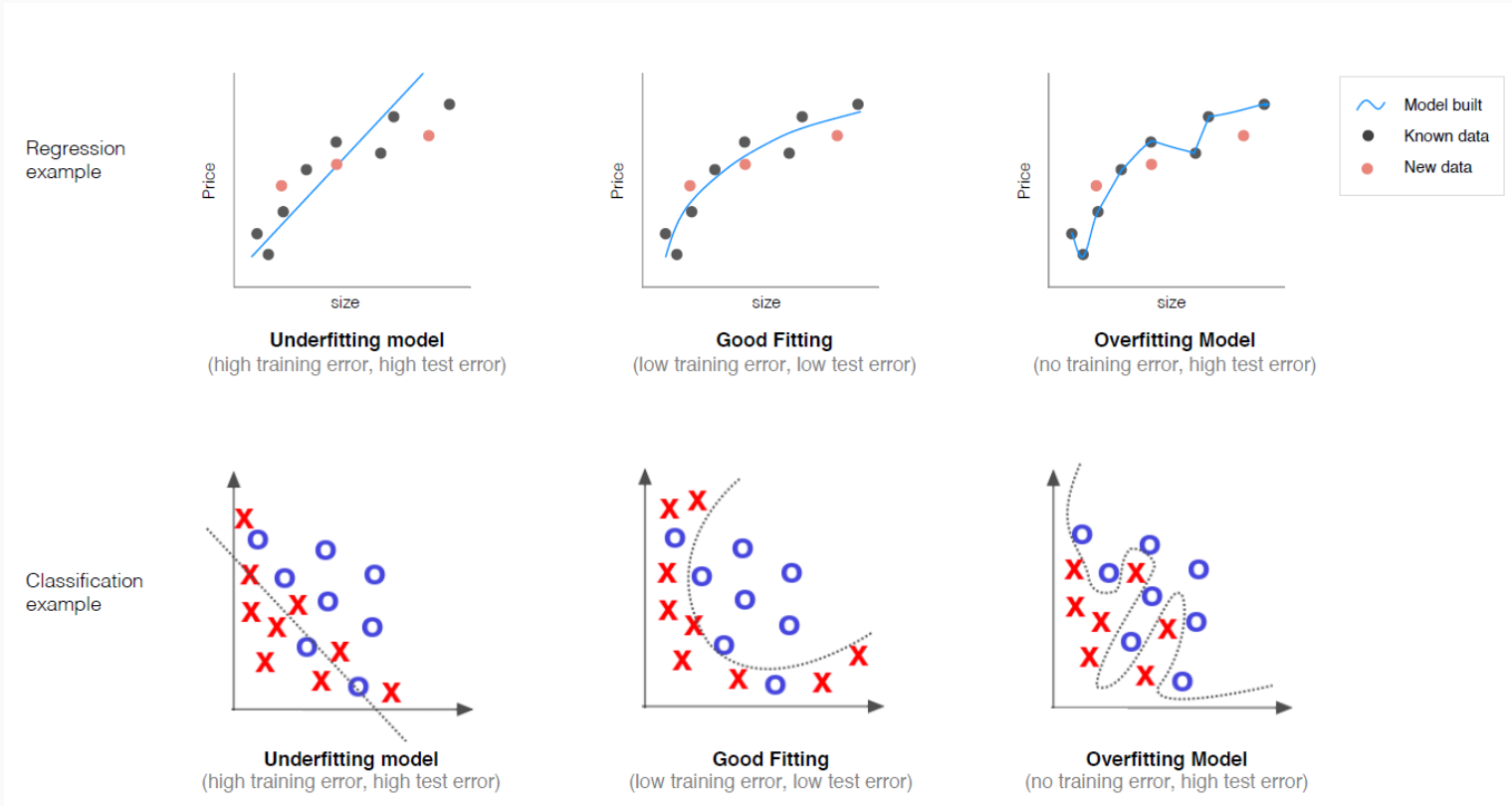
- ROC 曲线越接近左上角（AUC 面积越大），表示分类性能越好；ROC 曲线的优点是，对类分布/不平衡数据不敏感；正例负例的占比变化，ROC 曲线不变。
- 若负例数远大于正例数，若 FP 很大，即有很多负例被预测为正例，用 ROC 曲线则会判断其性能很好，但是实际上其性能并不好。此时应改用 PR 曲线，因为 Precision 综合考虑了 TP 和 FP 的值，因此在极度不平衡的数据下（正例的样本较少），PR 曲线可能比 ROC 曲线更实用。

- **PR 曲线（Precision-Recall）**，即在不同分类阈值上，对比查准率（Precision）与召回率（Recall）的曲线，越靠近右上角越好：



注：这些经典的二分类度量，大部分也可以推广到多分类度量（放到后文）。

五. 欠拟合与过拟合、偏差与方差



为什么会出现过拟合，如何避免：

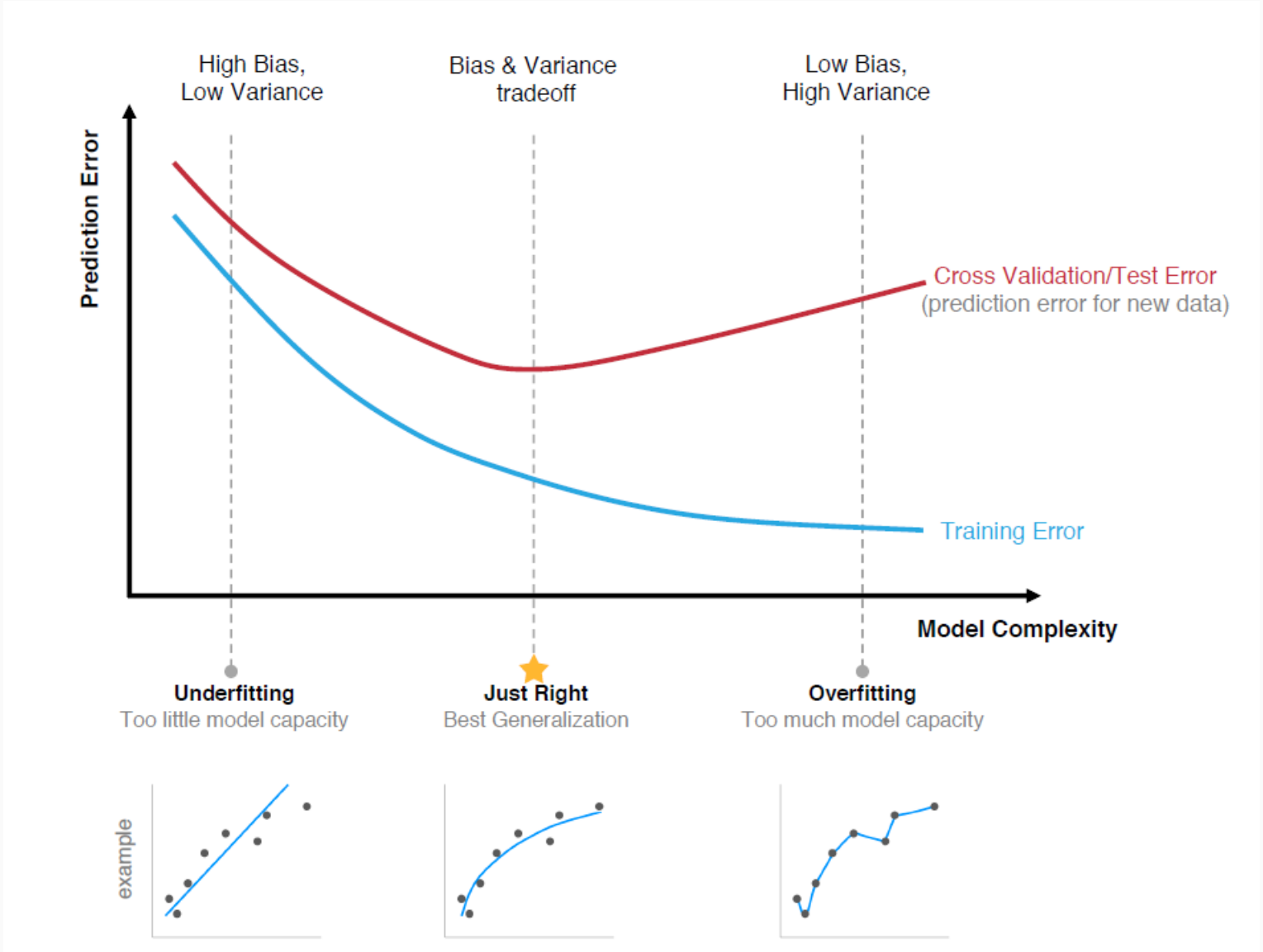
- **没有足够多的数据**：收集更多数据
- **数据噪声影响**：收集更好的数据（减少噪声）
- **模型过于复杂**：使用不那么复杂的模型
- **过度的损失优化**：减少优化，及早停止

在测试集上评估模型的性能表现（泛化能力）时，训练集的损失与一般化的测试集的损失之间的差异就，称为**泛化误差**，它有如下分解：

$$\underbrace{\sigma^2}_{\text{Variance of the data}} + \mathbb{E}_{xy}\underbrace{\left[\text{Var}_{\mathcal{D}_n}\left(\hat{f}_{\mathcal{D}_n}(\mathbf{x}) \mid \mathbf{x}, y\right)\right]}_{\text{Variance of inducer at } (\mathbf{x},y)} + \mathbb{E}_{xy}\underbrace{\left[\left(\left(f_{\text{true}}(\mathbf{x}) - \mathbb{E}_{\mathcal{D}_n}\left(\hat{f}_{\mathcal{D}_n}(\mathbf{x})\right)\right)^2 \mid \mathbf{x}, y\right)\right]}_{\text{Squared bias of inducer at } (\mathbf{x},y)}$$

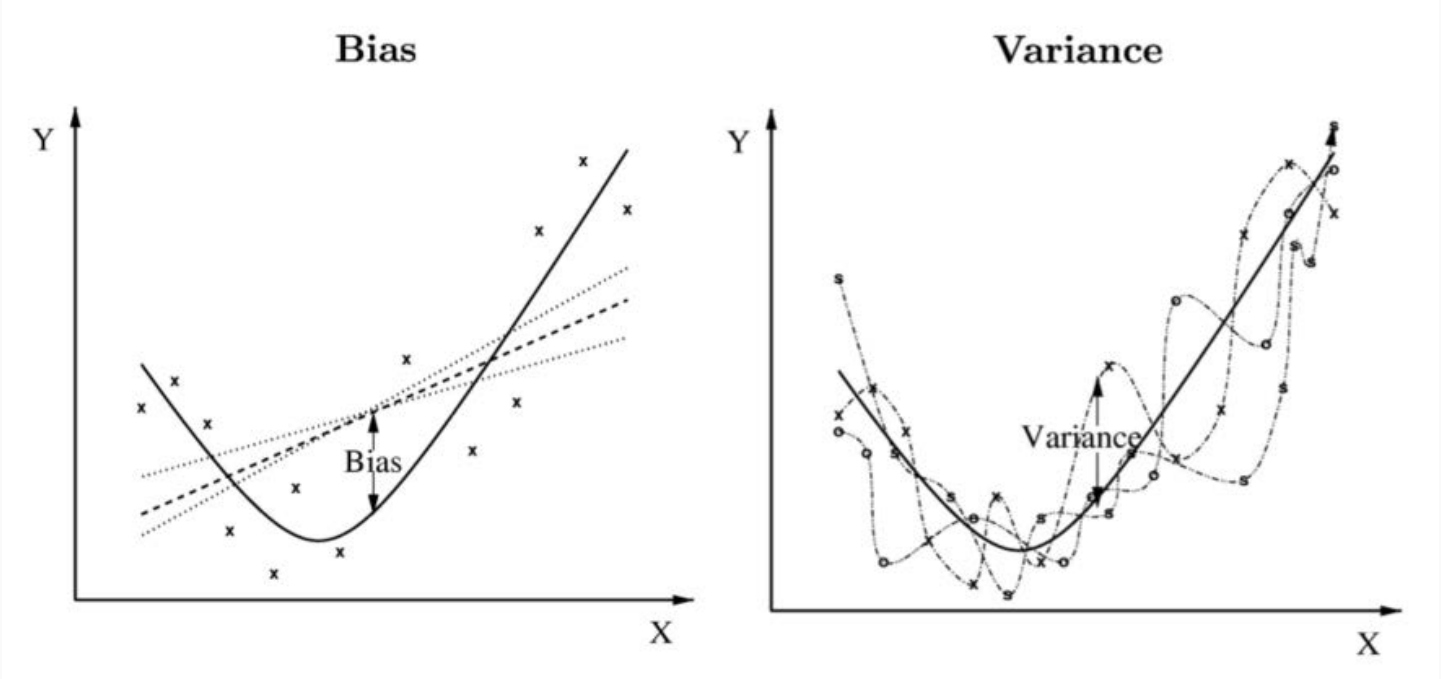
- **噪声**：来自数据的噪声，无论做什么都不会低于该误差；
- **方差**：如果改变训练数据，预测值在测试点上的平均波动情况，也表达了学习器不顾实际信号而学习随机事物的倾向（过拟合）；
- **偏差**：测试点的平均”偏离”程度（欠拟合）。

简单模型（欠拟合）往往高偏差、低方差；复杂模型（过拟合）低偏差、高方差。


High Bias,
Low Variance
Bias & Variance
tradeoff
Low Bias,
High Variance
Prediction Error
Cross Validation/Test Error
(prediction error for new data)
Training Error
Model Complexity
Underfitting
Too little model capacity
Just Right
Best Generalization
Overfitting
Too much model capacity
example

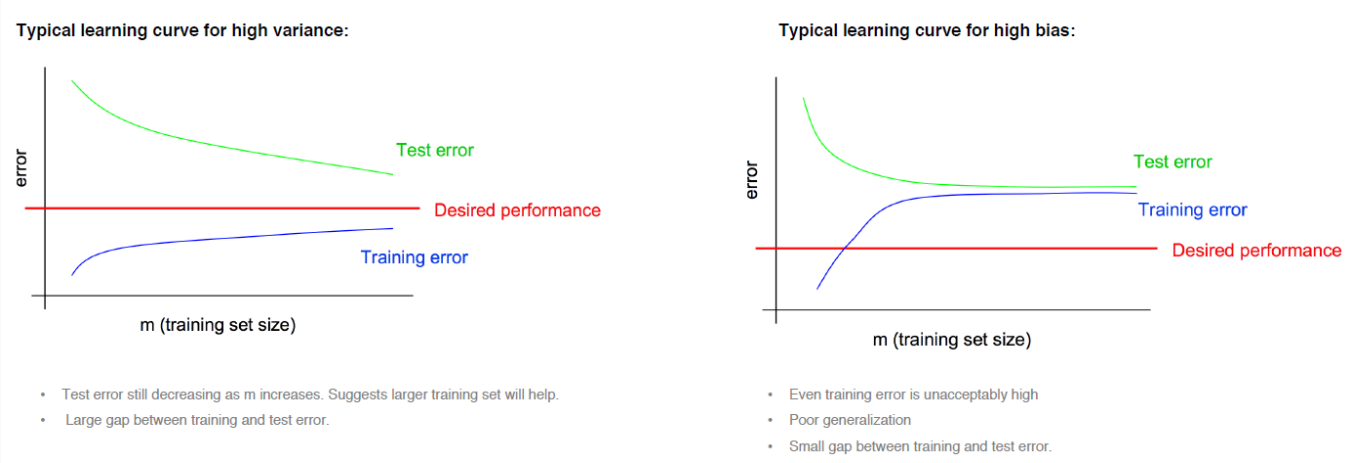
偏差与方差是相互冲突的，经常需要在二者之间找到一个平衡，称为**偏差-方差权衡**。

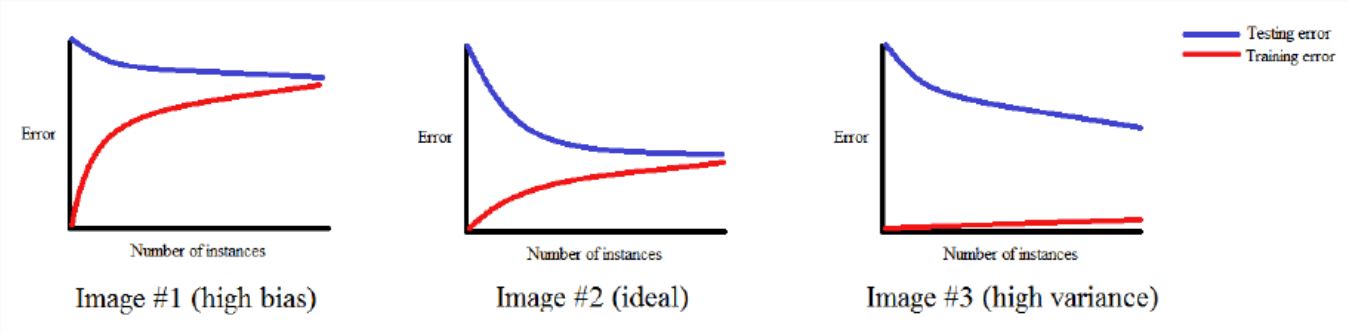
- 当模型过于简单时，模型拟合能力弱，数据的扰动不足以使模型产生显著变化，此时偏差是泛化误差的主要来源；
- 当模型过于复杂时，模型拟合能力强，数据的轻微扰动导致模型发生显著变化，此时方差是泛化误差的主要来源。



- **左图**：一个具有高偏差的模型无法拟合数据中存在的曲线关系。
- **右图**：理论上，一个没有偏差和高方差的模型可以学习数据中的真实模式，但实际上，不同的训练集会产生截然不同的结果。

随着某种让模型从简单到复杂的迭代过程的进行，**学习曲线**可以监视训练集误差和测试集误差的变化，以用来诊断欠拟合（偏差）和过拟合（方差）：





参考文献

Bernd Bischl, Ludwig Bothmann, e. a. (2022). *Introduction to Machine Learning (I2ML)*.

Liang(梁劲), J. (2019). *Getting Started with Machine Learning*.

Marc Becker, Przemysław Biecek, M. B. e. a. (2022). *Flexible and Robust Machine Learning Using mlr3 in R*. mlr-org.

王圣元 (2020). 快乐机器学习. 电子工业出版社.

知乎 NaNNN (2022). 多分类模型 *Accuracy, Precision, Recall* 和 *F1-score* 的超级无敌深入探讨.