经典模型

第一章 课程简介与回顾

1. 课程简介
2. 概念回顾
3. 模型回顾
4. 分类评价指标回顾
5. 小结

1. 课程简介

简介

- 《经典模型》是一门以机器学习中常用的、经典的模型为主要内容的课程。
- 目的：熟练掌握机器学习基本概念；熟练掌握多种常见机器学习算法模型原理；熟练应用常见机器学习模型。
- 主要算法：线性回归、逻辑回归、人工神经网络、支持向量机、奇异值分解、线性判别分析。

课程定位

- 承上启下：以方向基础课程为基础，继续学习机器学习领域的重要算法；
- 强化基础：加强对机器学习基本概念的理解与应用；
- 提升视野：掌握更多经典机器学习模型，具备解决一定实际问题的能力。

多种多样的机器学习模型

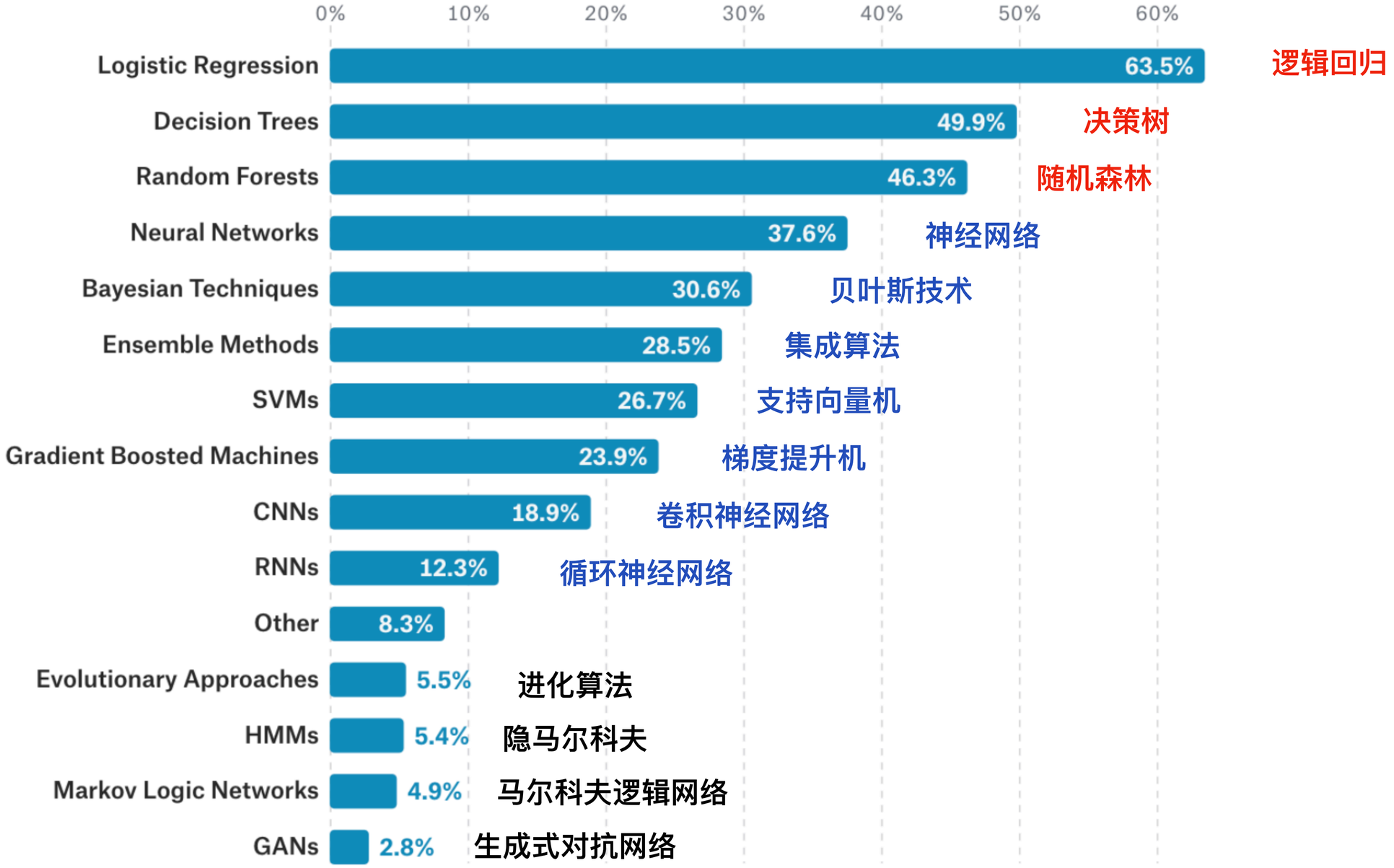
- K近邻
- 朴素贝叶斯
- 决策树
- 集成学习
- 聚类
- 主成分分析
- 线性回归
- 逻辑回归
- 最大熵模型
- 多项式回归
- 人工神经网络
- 深度学习
- 强化学习
- 降维与可视化分析
- 支持向量机
- EM算法
- 独立成分分析
- 度量学习
- 线性判别分析
- 隐马尔科夫模型
- 概率图模型
- 关联分析
- 规则学习
- ……

面对多种多样的机器学习模型，我们应该从何学起?

多种多样的机器学习模型

- 构成机器学习领域基石的模型被认为是“好模型”；
- 具备良好理论基础和性质的模型被认为是“好模型”；
- 工程实践中使用较多且效果较好的模型被认为是“好模型”。

kaggle: 2017机器学习及数据科学调查



- 《机器学习》（周志华著）中有：线性模型、决策树、神经网络、支持向量机、朴素贝叶斯、集成学习、聚类、降维与度量学习、计算理论学习、概率图模型、规则学习、强化学习等。

- 《统计学习方法》（李航著）中有：感知机、K近邻法、朴素贝叶斯、决策树、逻辑斯蒂回归与最大熵模型、支持向量机、提升方法、EM算法及其推广、隐马尔科夫模型、条件随机场等。

- 《机器学习实战》（Peter Harrington著）中有：K近邻算法、决策树、朴素贝叶斯、逻辑斯蒂回归、支持向量机、AdaBoost、树回归、K均值聚类、Apriori、FP-growth、PCA、SVD等。

经典模型

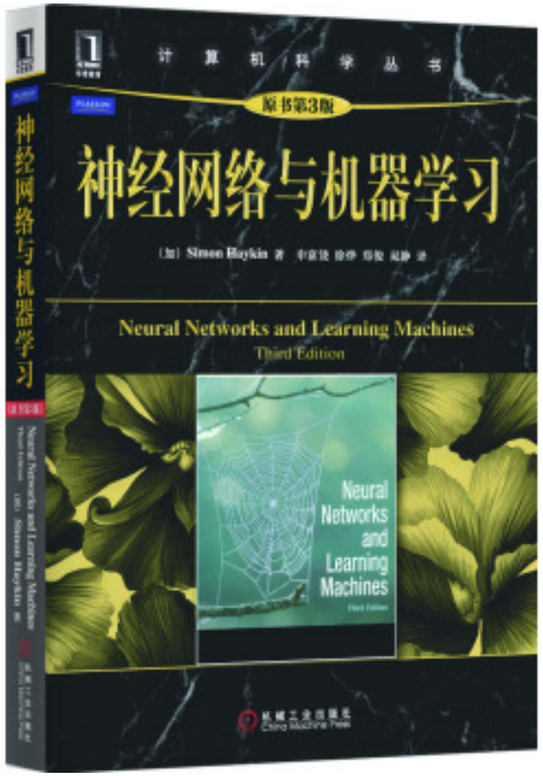
根据所学基础算法以及重要程度，《经典模型》课程主要讲解：

- 线性回归、逻辑回归、
- 人工神经网络、
- 支持向量机、
- 奇异值分解、
- 线性判别分析等。

主要参考书


机器学习
周志华 著
MACHINE
LEARNING
机器学习
清华大学出版社


统计学习方法
（第2版）
李航 著
清华大学出版社

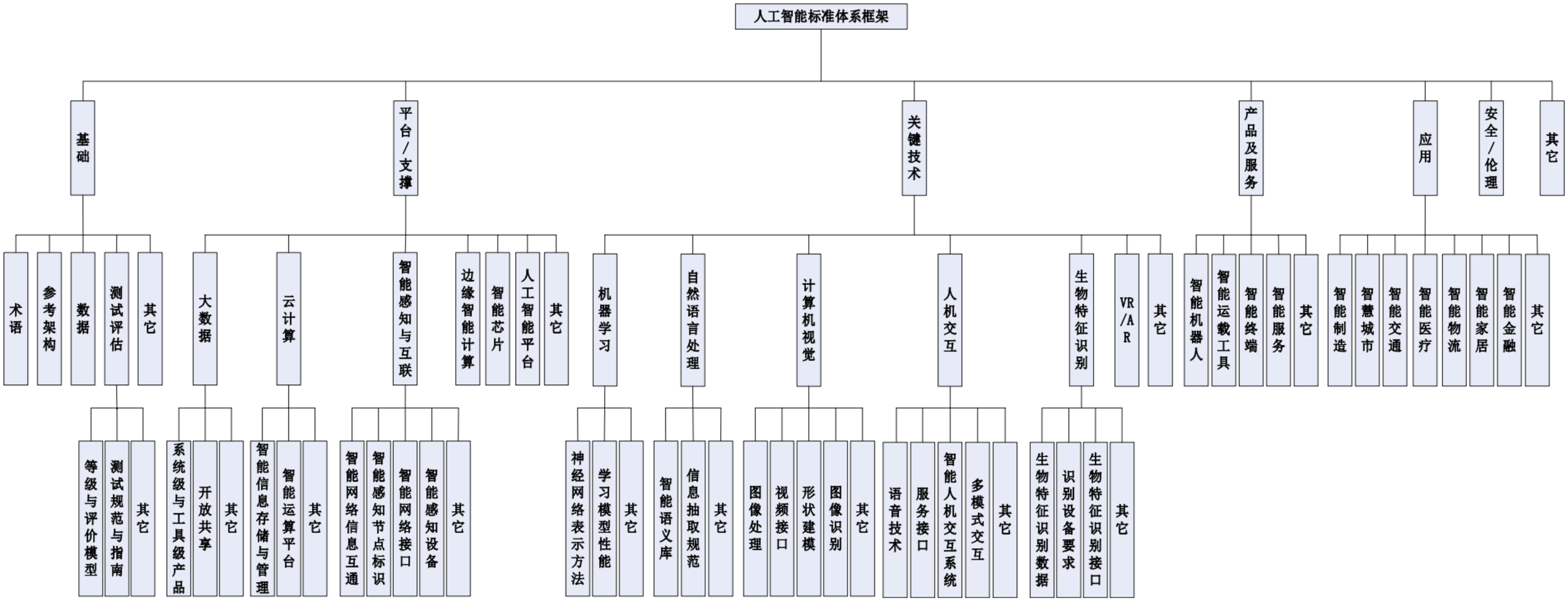

计算机科学丛书
原书第3版
神经网络与机器学习
Neural Networks and Learning Machines
Third Edition
Neural
Networks
and
Learning
Machines

2. 概念回顾

2.1 人工智能与机器学习

人工智能（Artificial Intelligence，AI），也称为机器智能、计算机智能，是指由人工制造出来的系统所表现出来的智能，或者用人工方法在机器上实现的智能行为，包括：感知、推理、学习、通信以及复杂环境下的动作行为。

人工智能标准体系框架


人工智能标准体系框架
基础
术语
参考架构
数据
测试评估
等级与评价模型
测试规范与指南
其它
其它
平台／支撑
大数据
系统级与工具级产品
开放共享
其它
云计算
智能信息存储与管理
智能运算平台
其它
智能感知与互联
智能网络信息互通
智能感知节点标识
智能网络接口
智能感知设备
其它
边缘智能计算
智能芯片
人工智能平台
其它
关键技术
机器学习
神经网络表示方法
学习模型性能
其它
自然语言处理
智能语义库
信息抽取规范
其它
计算机视觉
图像处理
视频接口
形状建模
图像识别
其它
人机交互
语音技术
服务接口
智能人机交互系统
多模式交互
其它
生物特征识别
生物特征识别数据
识别设备要求
生物特征识别接口
其它
VR/AR
其它
产品及服务
智能机器人
智能运载工具
智能终端
智能服务
其它
应用
智能制造
智慧城市
智能交通
智能医疗
智能物流
智能家居
智能金融
其它
安全／伦理
其它

机器学习（machine learning）算法是一类从数据中**自动分析获得规律**，并**利用规律对未知数据进行预测**的算法。因为学习算法中涉及了大量的统计学理论，机器学习与推断统计学联系尤为密切，也被称为**统计学习理论**。

1952年，阿瑟·萨缪尔（Arthur Samuel）在 IBM 公司研制了一个具有自学习能力的西洋跳棋程序，可通过大量棋局分析逐渐辨识出当前局面下的“好棋”和“坏棋”，从而不断提高弈棋水平。

1956年，阿瑟·萨缪尔应约翰·麦卡锡（人工智能之父）之邀，在标注着人工智能学科诞生的达特茅斯会议上介绍了这项工作，并发明了“机器学习”这个词，将其定义为“**不显式编程地赋予计算机能力的研究领域**”。

2.2 学习算法

机器学习形式化定义：给定一个**任务T**和**性能度量P**，一个计算机程序被认为可以从**经验E**中学习是指，**通过经验E改进后，它在任务T上由性能度量P衡量的性能有所提升**。

- **样本**：所研究对象的一个个体。相当于统计学中的实例。
- **特征**（属性）：用于表征样本的观测信息。
- **特征空间**：分别以每个特征作为一个坐标轴，所有特征所在坐标轴张成的一个用于描述不同样本的空间。
- **标记**（标签）：用于表征样本类别或感兴趣信息的观测信息。
- **数据集**（样本集）：由若干样本构成的集合。

任务T

- **分类**：基于已知类别标签样本构成的训练集，学习预测模型。预测结果为事先指定的两个或多个类别中 的某一个，或预测结果来自数目有限的离散值之一。
- **回归**：基于已知答案的样本构成的训练集，估计自变量与因变量之间关系的统计过程，进而基于该关系对新的观测产生的 输出进行预测，预测输出为连续的实数值 。
- **聚类**：对给定的数据集进行划分，得到若干“簇”，使得“簇内”样本之间较“簇间”样本之间更为相似。通过聚类得到的可能各簇对应一些潜在的概念结构，聚类是自动为给定的样本赋予标记的过程。
- **特征降维**：将初始的数据高维表示转化为关于样本的低维表示，借助由高维输入空间向低维空间的映射，来简化输入。

- **机器翻译**：输入是一种语言的符号序列，计算机程序必须将其转化成另一种语言的符号序列。
- **转录**：对输入一些相对非结构化表示的数据，转录为离散的文本形式。如OCR、语言识别。
- **异常检测**：在一组事件或对象筛选，并标记不正常或非典型的个体。如信用卡欺诈检测。
- **合成和采样**：生成与训练数据相似的样本。
- **去噪**：输入是干净样本经过未知损坏过程后得到的损坏样本，算法根据损坏样本预测干净样本。
- **密度估计**：此问题中，机器学习函数可以解释成样本采样空间的概率密度函数。
- ……

性能度量P

- 通常使用“**测试集**”数据来评估系统性能。
- 对于不同任务，性能度量方式往往不同。如对于分类、转录任务，可以使用错误率；对回归任务则可以使用均方误差衡量。
- 同类型任务，不同场景和应用下，性能度量也可以不相同。
- 在有些任务中，知道应该度量哪些数值，但是度量它是不现实的，只能设计对象的替代标准，近似度量。这在密度估计中较为常见。

经验E

- 根据学习过程不同，机器学习算法可以分为**监督学习算法**、**无监督学习算法**和**强化学习算法**。
- **无监督学习**算法使用含有很多特征的数据集训练，然后**学习出在这个数据集上有用的结构性质**。
- **监督学习**算法使用含有很多特征和标签的数据集训练，然后学习出**能够预测标签值**的模型。
- 学习算法从数据集中获取经验。

小练习

请指出以下场景的T、P、E

- 鸢尾花分类
- 手写数字识别
- 波士顿房价预测

2.3 假设、假设空间、三要素

- **假设**：每一个具体的模型就是一个假设。
- **假设空间**：所有假设构成的空间。
- **版本空间**：基于有限规模训练集进行假设的匹配搜索，会存在多个假设与训练集一致的情况，称这些假设组成的集合为版本空间。
- **机器学习三要素**：**模型**、**策略**（选取模型的准则）、**算法**（优化算法）

2.4 风险和经验风险

- **期望风险**（风险）：模型在真实数据分布上的代价。
- **经验风险**：模型在给定的训练集上的代价。
- **经验风险最小化**：最小化经验风险达到模型学习的目的，**经验风险最小化容易产生过拟合**。
- **结构风险最小化**：假设空间中，使结构风险最小话。

2.5 容量、过拟合和欠拟合

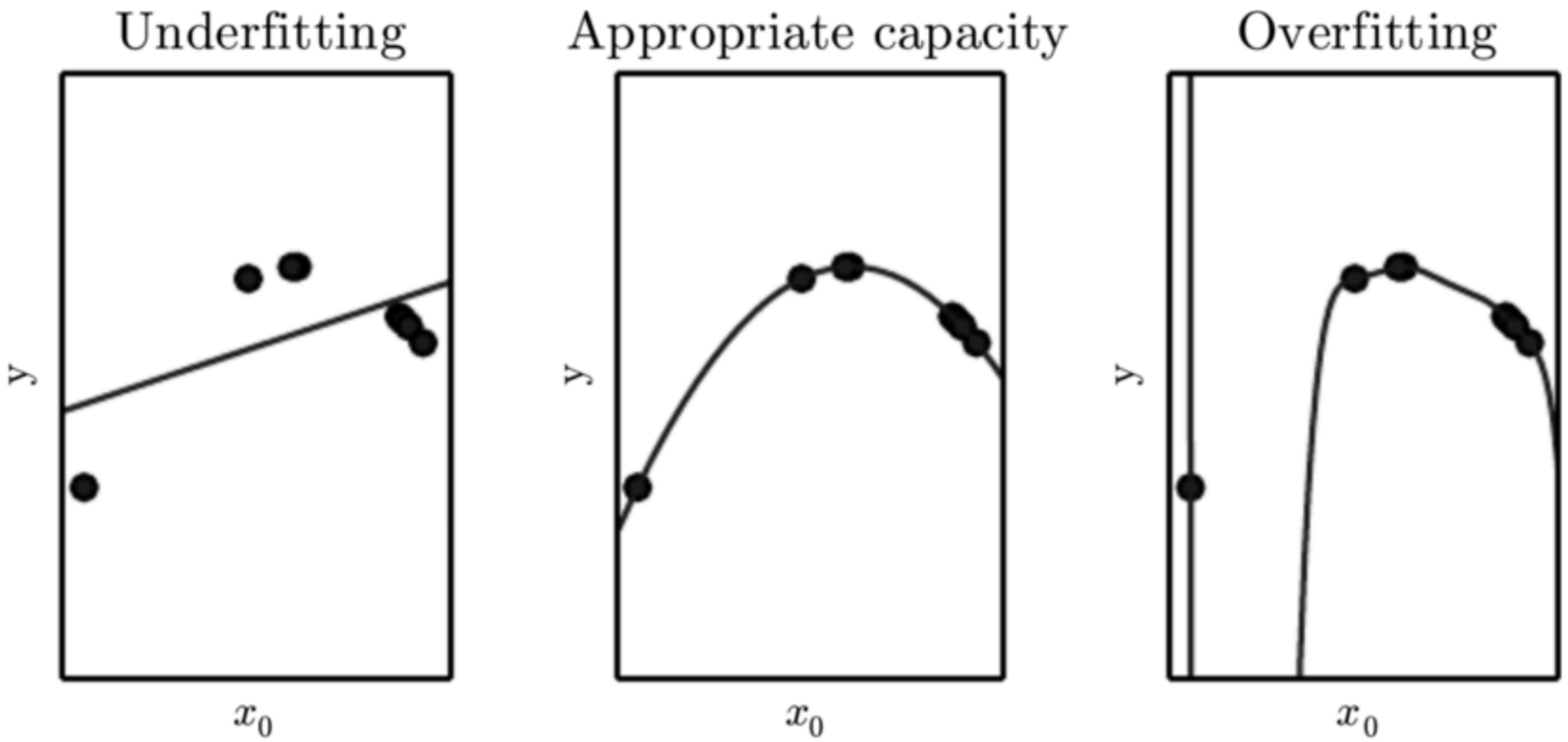
- 机器学习的**主要挑战是我们的算法必须能够在未观测的新输入上表现良好，而不只是在训练集表现良好**。
- **泛化**：在先前未观测到的输入上表现良好的能力。
- **训练误差**：在训练集上计算的误差。
- **泛化误差（测试误差）**：在测试集上计算的误差。
- 机器学习和优化不同的地方在于，我们**希望降低训练误差的同时也降低泛化误差**。

- **欠拟合**：模型训练误差足够高。
- **过拟合**：模型训练误差足够低，而泛化误差高。
- **容量**：模型拟合各种函数的能力。
- 统计学习理论提供了量化模型容量的多种方法，最著名的是**VC维度（Vapnik-Chervonenkis dimension, VC）**。
- **模型容量高时，模型容易出现过拟合，容量低时容易出现欠拟合。**

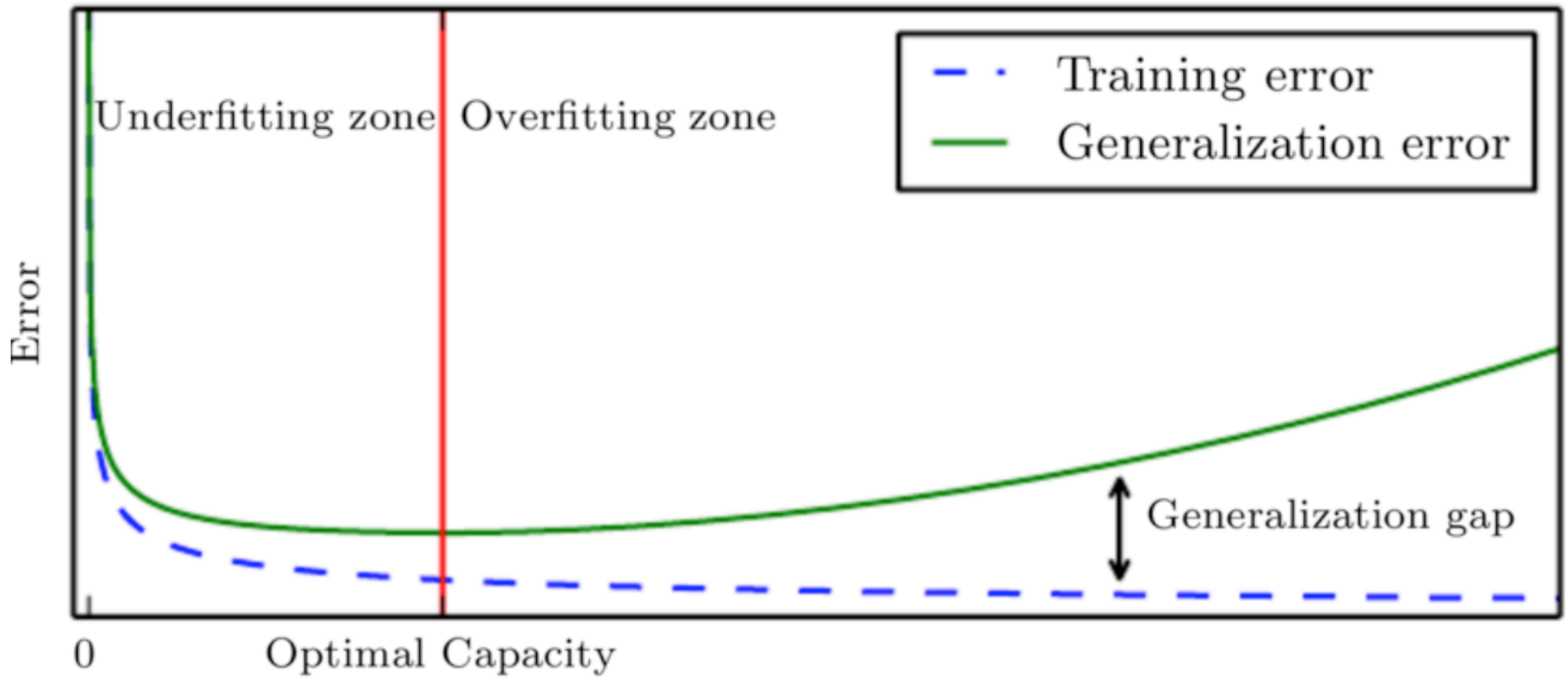
模型选择原则

- **奥卡姆剃刀原则**：若多个假设与经验观测一致，则选择最简单的那个（即模型容量低的那一个）。

- **多释原则**：保留与经验观察一致的所有假设，集成学习与此思想一致。



上图中，使用线性函数拟合数据导致欠拟合；使用二次函数拟合泛化得很好；使用多项式函数拟合数据导致了过拟合。



随着模型容量的增加，模型的训练误差和测试误差情况

没有免费午餐定理

- **没有免费午餐定理**：在所有可能的数据生成分布上平均之后，每一个分类算法在未事先观测的点上都有相同的错误率。即在某种意义上，没有一个机器学习算法总是比其它的要好。我们能够设想的最先进的算法和简单地将所有点归为同一类的简单算法有着相同的评价性能（在所有可能的任务上）

- 没有免费午餐定理意味着机器学习研究的目标不是找一个通用学习算法或是绝对最好的学习算法，而是找在我们关注的数据生成分布上效果最好的算法。

正则化

- **正则化**：通过修改学习算法等方法，**降低泛化误差的方法**。
- 没有免费午餐定理清楚地阐述了没有最优的学习算法，特别地，**没有最优的正则化形式**。

2.6 超参数和验证集

- 大多数机器学习算法都有**超参数**，可以设置来控制算法行为。
- 超参数的值通常不是通过学习算法本身学习出来的（尽管可以设置嵌套的学习过程，来学的超参数）。
- 有时一个选项被设为不用学习的超参数，是因为它太难优化了，但更多的情况是，该选项必须是超参数，例如控制模型容量的超参数。
- **验证集**：**用于挑选超参数的数据子集**。

K折交叉验证

- 当数据集划分成训练集与测试集时，若测试集规模太小，会使得测试误差估计的统计不确定性增大，使得很难判断不同算法性能好坏。替代方法是使用所有样本估计平均测试误差，代价是增加了计算量。

- **K折交叉验证**：将数据分成K个不重合的子集，每次选出K-1份数据用于训练，剩余1份用于测试（每次剩余的1份均不相同），测试误差可以估计为K次计算后的平均测试误差。

其它数据集划分方法

- **留出法**：将数据集划分为互斥的训练集和测试集两部分。
- **留一法**（交叉验证法）：当K折交叉验证中K等于样本数量时，为留一法交叉验证。
- **自助法**：给定包含m个样本的数据集，从中有放回的采样m个样本，组成新数据集作为训练集，未被采样到的样本作为测试集。

3. 模型回顾

3.1 K近邻

K 近邻算法要点

- **模型思想**：近朱者赤近墨者黑。
- 懒惰学习（无显示训练过程），非参数模型。
- 数据集、距离度量方式、近邻数、决策规则（多数表决、加权投票）等决定了模型能力。
- 距离度量：L_p距离、绝对值距离、欧氏距离、切氏距离。
- 使用交叉验证或验证集验证对 K 值进行选择
- 使用KD树提升搜索速度。

K近邻分类算法流程

① 训练集 $D = \{(\boldsymbol{x}_i, y_i), i = 1,...,N\} \subset R^d \times Y$

② 对训练集 D 属性进行预处理，记录预处理参数，制定距离度量，选择 K 值

③ 训练集 D 内找到预处理的样本 $\boldsymbol{x}$ 的前 K 个近邻，记为 $N_k(\boldsymbol{x})$

$N_k(\boldsymbol{x}) = N_{k,1}(\boldsymbol{x}) \cup ... \cup N_{k,C}(\boldsymbol{x})$ 其中 C 表示类别数量

④ 结合指定的分类规则，对 $\boldsymbol{x}$ 的类别 y 进行预测

$$\hat{y} = \underset{i \in Y=1,...,C}{\operatorname{argmax}} \sum_{\boldsymbol{x}_j \in N_k(\boldsymbol{x})} I(y_j = i) \quad \text{其中 } I(y_j = i) = \begin{cases} 1, & if \quad y_j = i \\ 0, & if \quad y_j \neq i \end{cases}$$

K近邻分类算法流程

① 训练集 $D = \{(\boldsymbol{x}_i, y_i), i = 1,...,N\} \subset R^d \times Y$

② 对训练集 D 属性进行预处理，记录预处理参数，制定距离度量，选择 K 值

③ 训练集 D 内找到预处理的样本 $\boldsymbol{x}$ 的前 K 个近邻，记为 $N_k(\boldsymbol{x})$

$N_k(\boldsymbol{x}) = N_{k,1}(\boldsymbol{x}) \cup ... \cup N_{k,C}(\boldsymbol{x})$ 其中 C 表示类别数量

④ 结合指定的分类规则，对 $\boldsymbol{x}$ 的类别 y 进行预测

$$\hat{y} = \underset{i \in Y=1,...,C}{\operatorname{argmax}} \sum_{\boldsymbol{x}_j \in N_k(\boldsymbol{x})} I(y_j = i). \text{ 其中 } I(y_j = i) = \begin{cases} 1, & if \quad y_j = i \\ 0, & if \quad y_j \neq i \end{cases}.$$

K近邻回归算法流程

① 训练集 $D = \{(\boldsymbol{x}_i, y_i), i = 1,...,N\} \subset R^d \times R$

② 对训练集 D 属性进行预处理，记录预处理参数，制定距离度量，选择 K 值

③ 训练集 D 内找到预处理的样本 $\boldsymbol{x}$ 的前 k 个近邻，记为 $N_k(\boldsymbol{x})$

④ 结合指定的分类规则，对 $\boldsymbol{x}$ 的类别 y 进行预测

$$y = \frac{1}{k} \sum_{\boldsymbol{x}_i \in N_k(\boldsymbol{x})} y_i \,.$$

3.2 决策树

决策树算法要点

- **模型思想：分而治之。**
- 包括ID3、C4.5、CART（CART分类树、CART回归树）等
- 模型既可处理度量型特征，又可处理非度量型特征（ID3不能处理度量特征）。
- 分类树根据训练集中每个特征对样本划分的带来的纯度增幅从大到小依次构建。
- CART回归树
- 不纯度度量方式：熵不纯度、基尼不纯度、误差不纯度
- 模型决策速度快、语义可表示、可嵌入专家的先验知识。

熵不纯度

$$I_{Entropy}(D) = -\sum_{i=1}^{K} P_i \log_2 P_i$$

约定 $0 \log 0 = 0$

各类别等概率出现： $I_{Entropy}(D) = -\sum_{i=1}^{K} \frac{1}{K} \log_2 \frac{1}{K} = \log_2 K$ 类越多，不纯度越高

有多个类别且各类别等概率出现时，不纯度最高，只出现一个类别时，不纯度最低。

只出现一个类别： $I_{Entropy}(D) = 0$

Gini不纯度

$$I_{Gini}(D) = \sum_{j=1}^{k} \sum_{i=1, i \neq j}^{k} P_i P_j = 1 - \sum_{j=1}^{k} P_j^2$$

各类别等概率出现： $I_{Gini}(D) = 1 - \sum_{i=1}^{K} \frac{1}{K^2} = \frac{K-1}{K}$ **类越多，不纯度越高**

只出现一个类别： $I_{Gini}(D) = 0$

误差不纯度

$$I_{Error}(D) = 1 - \max_{j \in \{1,\ldots,K\}} P_j$$

各类别等概率出现： $I_{Error}(D) = 1 - \frac{1}{K} = \frac{K-1}{k}$ **类越多，不纯度越高**

只出现一个类别： $I_{Error}(D) = 0$

利用不纯度进行特征选择

- 决策树利用每一个属性的不同取值对数据集进行划分，但期望优先使用划分后使得纯度最大的方法；
- 通过将样本以不同属性的不同取值进行划分之后，利用不纯度计算公式可得最优划分属性；
- 使用最优划分属性，根据属性值不同取值将数据集划分成几份。
- 通常的使用基于熵不纯度或基尼不纯度的方法衡量划分带来的不纯度减少情况（即广义的信息增益）。

绝对信息增益

$$Gain(D,a) = I_{Entropy}(D) - \sum_{i=1}^{m} \frac{|D^{(i)}|}{D} I_{Entropy}(D^{(i)})$$

其中 a 表示某个特征，m 表示特征可能取值数量，k 表示类别数量

$$I_{Entropy}(D) = -\sum_{j=1}^{k} P_j \log_2 P_j = -\sum_{j=1}^{k} \frac{|D_j|}{|D|} \log_2 \frac{|D_j|}{|D|}$$

通过对每个特征的划分结果求信息增益，得到最大信息增益的特征：

$$a^* = \underset{a \in A}{\operatorname{argmax}}\, Gain(D,a)$$

相对信息增益（信息增益率）

$$Gain\_ration(D,a)=\frac{Gain(D,a)}{IV(a)}$$

其中 $IV(a)=-\sum_{i=1}^{m}\frac{|D^{(i)}|}{D}\log_2\frac{|D^{(i)}|}{|D|}$

$IV(a)$ 反映了特征的不纯度，除以 $IV(a)$ 会减少对数目较多属性的偏好。

通过对每个特征的划分结果求信息增益率，得到最大信息增益率的特征：

$$a^*=\underset{a\in A}{\operatorname{argmax}}\,Gain\_ration(D,a)$$

基于“基尼指数”的信息增益

$$Gain\_index(D,a)=\sum_{i=1}^{m}\frac{|D^{(i)}|}{|D|}I_{Gini}(D^{(i)})$$

I_{Gini} 表示从数据集中任取两个样本不一致的概率，值越小，表示数据集纯度越高

得到划分纯度最高的的特征：

$$a^{*}=\underset{a\in A}{\operatorname{argmin}}\,Gain\_index(D,a)$$

决策树基本算法

输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
属性集 $A = \{a_1, a_2, \ldots, a_d\}$.

过程: 函数 TreeGenerate(D, A)

1: 生成结点 node;
2: **if** D 中样本全属于同一类别 C **then**
3: 将 node 标记为 C 类叶结点; **return**
4: **end if**
5: **if** $A = \varnothing$ **OR** D 中样本在 A 上取值相同 **then**
6: 将 node 标记为叶结点, 其类别标记为 D 中样本数最多的类; **return**
7: **end if**

8: 从 A 中选择最优划分属性 a_*;
9: **for** a_* 的每一个值 a_*^v **do**
10: 为 node 生成一个分支; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
11: **if** D_v 为空 **then**
12: 将分支结点标记为叶结点, 其类别标记为 D 中样本最多的类; **return**
13: **else**
14: 以 TreeGenerate(D_v, $A \setminus \{a_*\}$)为分支结点
15: **end if**
16: **end for**

输出: 以 node 为根结点的一棵决策树

决策树剪枝

- 决策树规模很小时，模型可能表现出欠拟合；
- 决策树规模很大时，模型可能表现出过拟合；
- 预剪枝：当前节点划分后不能导致模型泛化能力提升，则剪枝。可能会导致欠拟合。
- 后剪枝：决策树生长完成后，合并部分叶子节点，如合并后模型泛化能力不降低，则剪枝。需要更多时间开销。

ID3、C4.5、CART

- ID3使用绝对信息增益选择划分属性；仅适用于离散、或者非数值型特征描述的样本集；不处理缺失信息、不涉及剪枝。
- C4.5使用相对信息增益选择划分属性；可处理连续数值特征的处理（连续值特征可重复使用）；可进行缺失值的处理；通过剪枝降低过拟合。
- CART采用二元划分方式，既可用于分类，又可用于回归；可处理连续数值特征的处理；面向分类采用基尼指数划分，面向回归使用最小平方残差等方式划分；通过剪枝降低过拟合。

3.3 朴素贝叶斯

算法要点

- 基于贝叶斯定理和特征条件独立假设（朴素的含义）的分类方法。
- 类先验概率、类条件概率是待估计的量。类条件概率有指数级的参数，其估计实际是不可行的，所以做出了条件独立性假设简化模型。
- 类先验概率、类条件概率可使用极大似然估计得到。
- 朴素贝叶斯法学习到生成数据机制，属于生成模型。

条件独立假设下的贝叶斯法

朴素贝叶斯对条件概率分布作了条件独立性假设：

$$P(X=x\,|\,Y=c_k)=P(X^{(1)}=x^{(1)},\ldots,X^{(n)}=x^{(n)}\,|\,Y=c_k)=\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}\,|\,Y=c_k)$$

贝叶斯定理：

$$P(Y=c_k\,|\,X=x)=\frac{P(Y=c_k)P(X=x\,|\,Y=c_k)}{\sum_j P(Y=c_j)P(X=x\,|\,Y=c_j)}$$

带入根据条件独立假设的结论可得：

$$P(Y=c_k\,|\,X=x)=\frac{P(Y=c_k)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}\,|\,Y=c_k)}{\sum_k P(Y=c_k)P(X=x\,|\,Y=c_k)}$$

对于每个 c_k，式子的分母都相同，所以在比较时，可以省略分母的计算

算法流程

输入：训练数据，$D=\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$ 其中 $x_i=(x_i^{(1)},\ldots,x_i^{(n)})^{\mathrm{T}}$ ， $x_i^{(j)}$ 是第 i 个样本的第 j 个特征。$x_i^{(j)}\in a_{j1},a_{j2},\ldots,a_{js_j}$ ， a_{jl} 是第 j 个特征可能取的第 l 个值，$j=1,2,...,n$ ， $l=1,2,...,s_j$ ， $y_i\in\{c_1,c_2,\ldots,c_k\}$ ， 实例 x ；

输出：实例 x 的分类。

① 计算类先验概率与类条件概率

$$P(Y=c_k)=\frac{\sum_{i=1}^{N} I(y_i=c_k)}{N},\quad k=1,2,...,K$$

$$P(X^{(j)}=a_{jl}\,|\,Y=c_k)=\frac{\sum_{i=1}^{N} I(y_i=c_k, x_i^{(j)}=a_{jl})}{\sum_{i=1}^{N} I(y_i=c_k)},\quad j=1,2,...,n;\quad l=1,2,...,s_j;\quad k=1,2,...,K$$

② 对给定实例，$x=(x^{(1)},x^{(2)},\ldots,x^{(n)})^{\mathrm{T}}$ 计算 $P(Y=c_k)\prod_{j=1}^{n} P(X^{(j)}=x^{(j)}\,|\,y=c_k)$

③ 确定实例 x 的类 $y=\underset{c_k}{\operatorname{argmax}} P(Y=c_k)\prod_{j=1}^{n} P(X^{(j)}=x^{(j)}\,|\,y=c_k)$

4. 分类评价指标回顾

4.1 两类别分类

两类别分类

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)

正确率： $Accuracy = \dfrac{TP + TN}{TP + TN + FN + FP}$

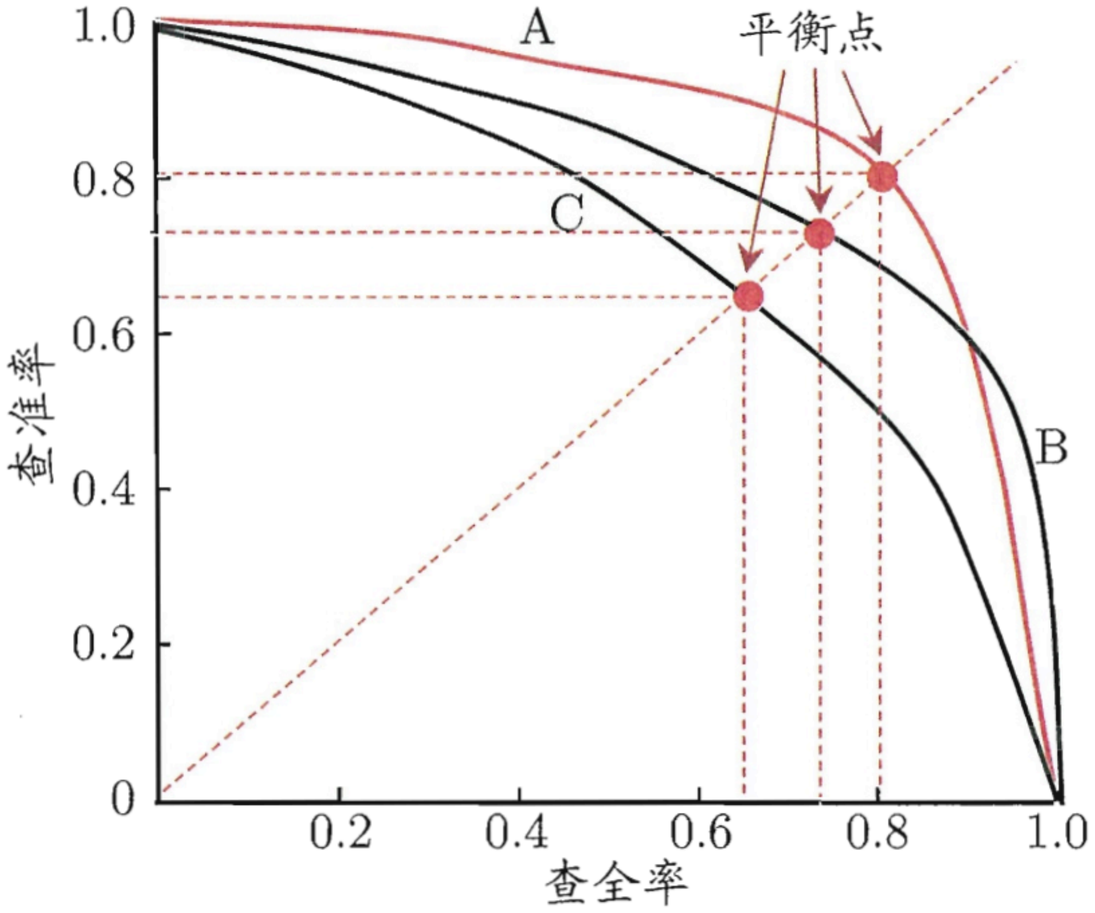
查准率： $P = \dfrac{TP}{TP + FP}$

查全率： $R = \dfrac{TP}{TP + FN}$

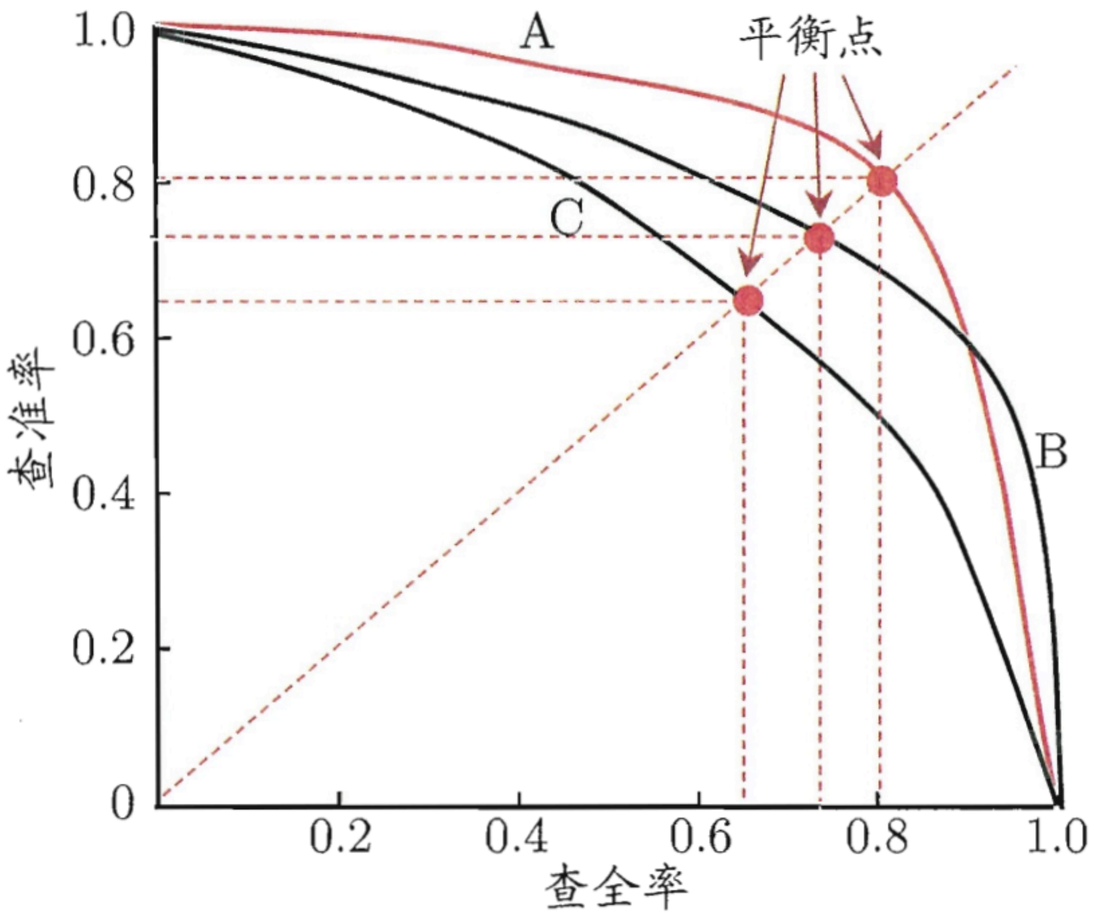
查准率与查全率是一对矛盾的量。一般的，查准率高，则查全率低，反之，查准率低，则查全率高。简单的任务中，可能使得查准率和查全率都高。

平衡查准率和查全率

1. 将模型预测结果按置信度从大到小排序
2. 逐个把样本（包括前面的）作为正类预测
3. 记录每次划分下的查准率与查全率
4. 以查全率为横轴，查准率为纵轴，根据上述记录查准率与查全率画出P-R曲线图（实际为折线图）。
5. 曲线的下面积叫做AP值，AP越大，则说明模型表现越好。



平衡查准率和查全率



1. AP值的计算往往不太容易更多的时候只关注曲线中的“平衡点”，也就是P=R的时候，此时查准率与查全率都比较高。
2. 为了综合衡量P=R的这个点的性能，我们使用F1分数描述，即P=R这一点的调和平均值：

$$F1 = \frac{2 \times P \times R}{P + R} = \frac{2 \times TP}{|D| + TP - TN}$$

3. 在实际应用中，对查准率和查全率的重视，不同，我们可以引入：

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

ROC和AUC

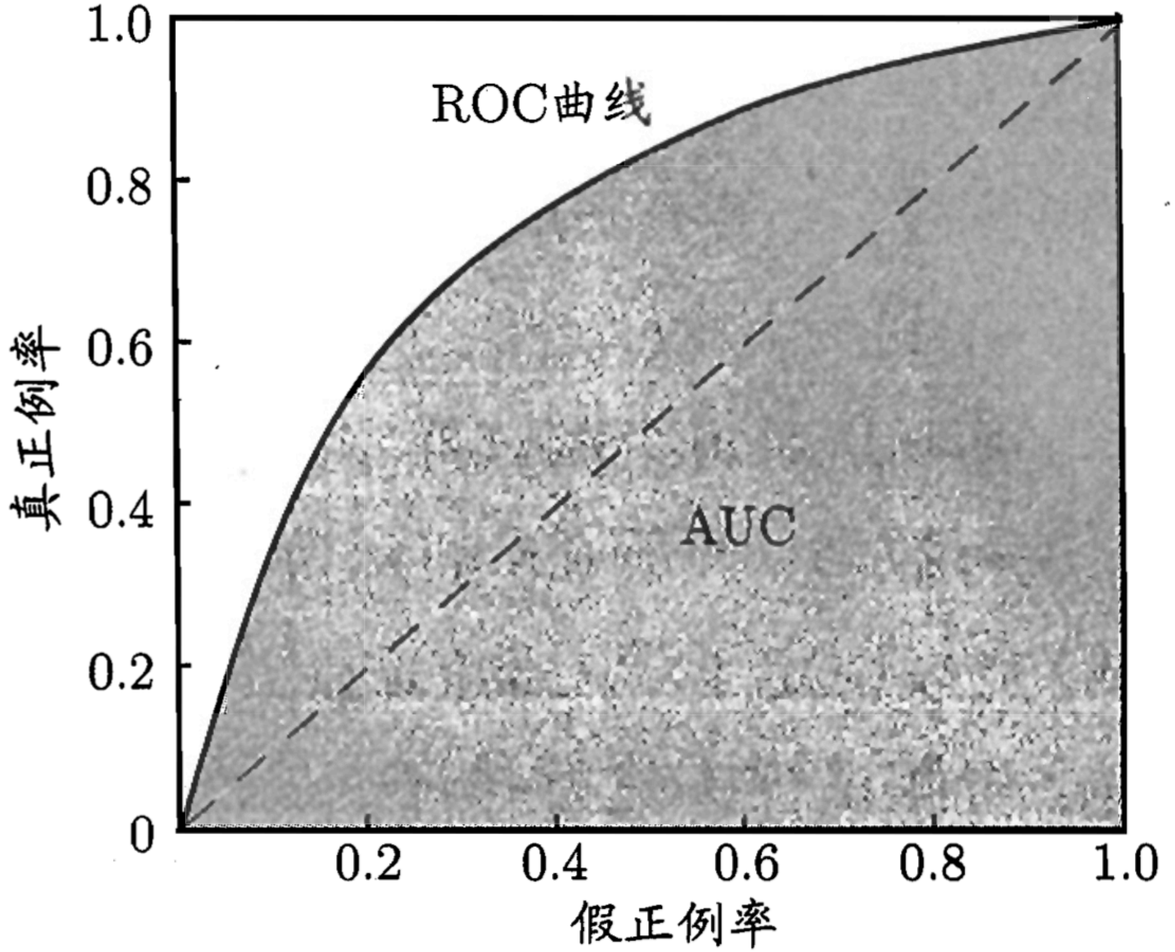
真正例率： $TPR = \frac{TP}{TP + FN}$ 与查全率一致

所有正例当中预测为正例的概率

假正例率： $FPR = \frac{FP}{TN + FP}$

所有反例当中预测为正例的概率

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)



4.2 多类别分类

宏平均

- 写出C 阶混淆矩阵，其元素 a_{ij} 中，i 表示真实类别第 i 类，预测结果为 j 第的样本数。
- 第 i 类查准率 $P_i = \frac{a_{ii}}{\sum_{j=1}^{C} a_{ji}}$　　宏查准率 $Macro\_P = \frac{1}{C}\sum_{i=1}^{C} P_i$
- 第 i 类查全率 $R_i = \frac{a_{ii}}{\sum_{j=1}^{C} a_{ij}} = Acc$　宏查全率 $Macro\_R = \frac{1}{C}\sum_{i=1}^{C} R_i$

- 宏 F_1分数 $Macro\_F_1 = \frac{2 \times Macro\_P \times Macro\_R}{Macro\_P + Macro\_R}$

微平均

- 由 C 阶混淆矩阵得到 C 个“ i 类对非 i 类”混淆矩阵。分别对 C 阶混淆矩阵计算 TP, FP, FN， 求得平均值 $\overline{TP}, \overline{FP}, \overline{FN}$

- 微查准率 $Micro\_P = \dfrac{\overline{TP}}{\overline{TP} + \overline{FP}}$

- 微查全率 $Micro\_R = \dfrac{\overline{TP}}{\overline{TP} + \overline{FN}}$

- 微 F_1 分数 $Micro\_F_1 = \dfrac{2 \times Micro\_P \times Micro\_R}{Micro\_P + Micro\_R}$

5. 小结

- 人工智能基本概念与机器学习形式化定义
- 模型容量与过拟合与欠拟合的关系
- 超参数的作用、确定方式；数据集的划分方法
- K近邻分类算法流程；决策树中信息增益的计算方式与基本算法流程；朴素贝叶斯的算法基本流程
- 混淆矩阵；二分类评价指标；多分类评价指标

THANKS