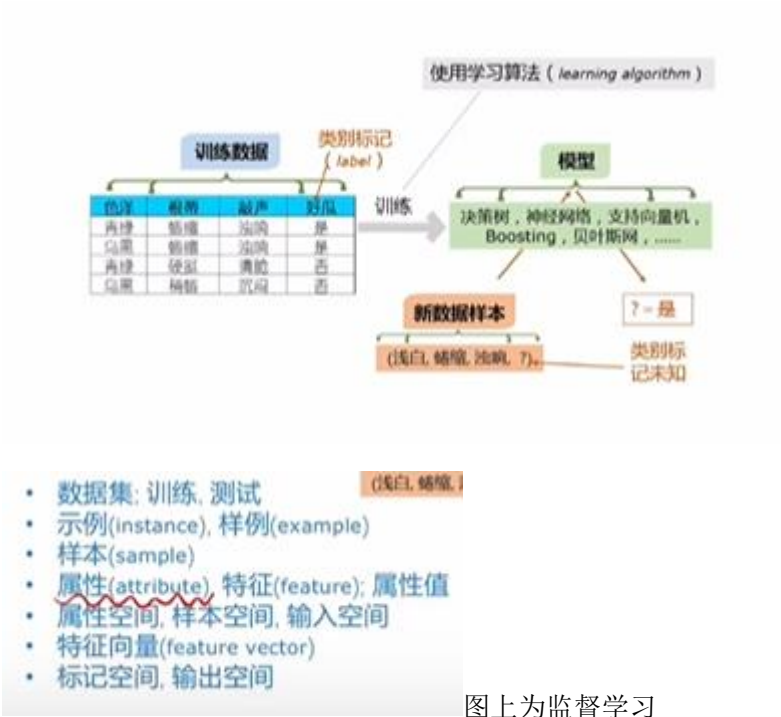


基础课程 1：机器学习简介

机器学习：利用经验改善系统自身的性能，智能数据分析

基本术语: one-hot 编码 【0,1】【1,0】



假设 (hypothesis)

真相 (ground-truth)

学习器 (learner)

分类 (数据离散), 回归 (数据连续)

二分类 (正类, 反类; 正和反是相对的概念), 多分类

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

独立: 数据之间没有相关性

典型的机器学习过程: 如果泛化能力强就是一个好模型

计算学习理论

Computational learning theory

最重要的理论模型:

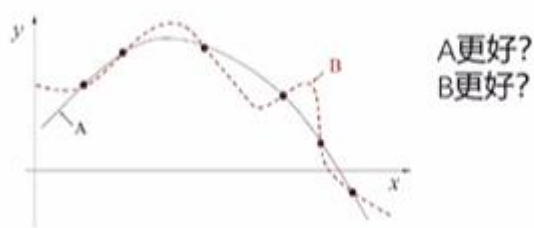
PAC (Probably Approximately Correct, 概率近似正确) learning model [Valiant, 1984]

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

机器学习的理论基础: 计算学习理论: PAC

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



任何一个有效的机器学习算法必有其偏好

一般原则: 奥卡姆剃刀 (Occam's Razor)

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!

没有免费的午餐!

NFL定理: 一个算法 \mathcal{A}_a 若在某些问题上比另一个算法 \mathcal{A}_b 好, 必存在另一些问题 \mathcal{A}_b 比 \mathcal{A}_a 好

基于具体的任务

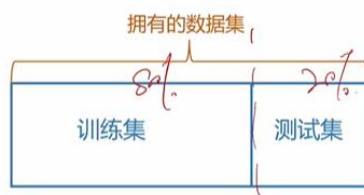
评估方法:

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:

1. 留出法

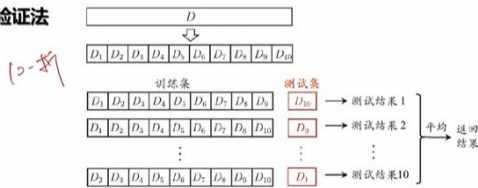


注意:

- 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)

2. K-折交叉验证法

k-折交叉验证法



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

□ 回归(regression) 任务常用均方误差:

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

MSE (mean square error)

错误率
vs.
精度

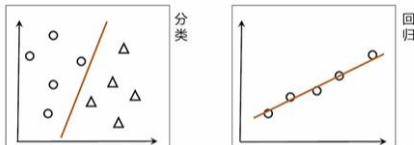
□ 错误率:

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

□ 精度:

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

线性模型 (linear model):



线性模型(linear model)试图学得一个通过属性的线性组合来进行预测的函数

$$f(x) = w_1 x_1 + w_2 x_2 + \dots + w_d x_d + b$$

向量形式: $f(x) = \mathbf{w}^T \mathbf{x} + b$

简单、基本、可理解性好

线性回归:

$$f(x_i) = wx_i + b \text{ 使得 } f(x_i) \simeq y_i$$

离散属性的处理: 若有 "序" (order), 则连续化; 否则, 转化为 k 维向量

令均方误差最小化, 有

$$\begin{aligned} (w^*, b^*) &= \arg \min_{(w, b)} \sum_{i=1}^m (f(x_i) - y_i)^2 \\ &= \arg \min_{(w, b)} \sum_{i=1}^m (y_i - wx_i - b)^2 \end{aligned}$$

对 $E_{(w, b)} = \sum_{i=1}^m (y_i - wx_i - b)^2$ 进行最小二乘参数估计

least square

优化目标:

WGZ^d 1000, -5000, 2.5, -0.2

基本思路: 优化模型的经验误差, 同时控制模型复杂度

1000.

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \ell_{0/1}(y_i (w^T x_i + b) - 1)$$

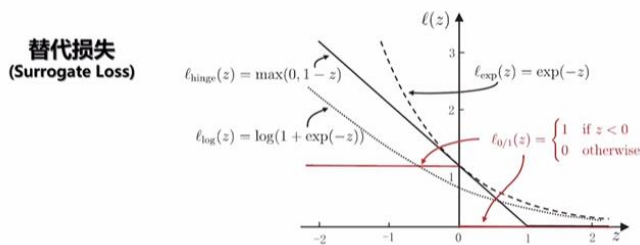
1000.

其中 $\ell_{0/1}$ 是 0/1 损失函数 (0/1 loss function):

$$\ell_{0/1}(z) = \begin{cases} 1, & z < 0; \\ 0, & \text{otherwise.} \end{cases}$$

障碍: 0/1 损失函数非凸, 非连续, 不易优化

终极目标



- 采用替代损失函数, 是在解决困难问题时的常见技巧
- 求解替代函数得到的解是否仍是原问题的解? 理论上称为替代损失的“一致性” (Consistency) 问题

软间隔支持向量机

原始问题

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \max(0, 1 - y_i (w^T x_i + b))$$

Support Vector Machine (SVM)

引入“松弛量” (Slack Variables)

$$\min_{w,b,\xi_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i$$

s.t. $y_i (w^T x_i + b) \geq 1 - \xi_i \quad \xi_i \geq 0, i = 1, 2, \dots, m.$

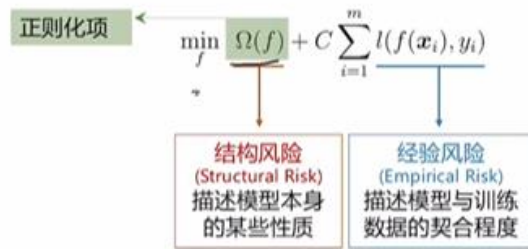
对偶问题

$$\max_{\alpha} \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i^T x_j$$

s.t. $\sum_{i=1}^m \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, \dots, m.$

正则化:

统计学习模型 (例如 SVM) 的更一般形式



□ 正则化可理解为“罚函数法”

通过对不希望的结果施以惩罚, 使得优化过程趋向于希望目标

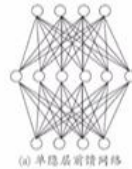
□ 从贝叶斯估计的角度, 则可认为是提供了模型的先验概率

多层前馈神经网络结构

多层网络: 包含隐层的网络

前馈网络: 神经元之间不存在同层连接也不存在跨层连接

隐层和输出层神经元亦称“功能单元” (Functional Unit)

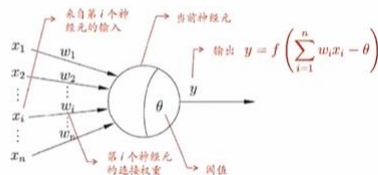


仅需一个包含足够多神经元的隐层, 多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数 [Hornik et al., 1989]

但是, 如何设置隐层神经元数是未决问题 (Open Problem). 实际常用“试错法”

“简单单元”: 神经元模型

M-P 神经元模型 [McCulloch and Pitts, 1943]



神经网络学得的知识蕴含在连接权与阈值中

决策树模型

Decision Tree

《Madam Long》:

决策树基于“树”结构进行决策

- 每个“内部结点”对应于某个属性上的“测试” (test)
- 每个分支对应于该测试的一种可能结果 (即该属性的某个取值)
- 每个“叶结点”对应于一个“预测结果”

学习过程: 通过对训练样本的分析来确定“划分属性” (即内部结点所对应的属性)

预测过程: 将测试示例从根结点开始, 沿着划分属性所构成的“判定测试序列”下行, 直到叶结点



图 4.1 西瓜问题的一棵决策树

策略: 分而治之

- 优化方法**
- 零阶优化方法
 - 一阶优化方法
 - 高阶优化方法
 - 随机优化方法

1. 无约束优化
2. x 在一个特点的域里 (例子: SVM)

类别不平衡 (class-imbalance)

不同类别的样本比例相差很大; “小类” 往往更重要

大类小类分别算:

Handwritten formula: $(1/n) \cdot (1/m) + (1/n) \cdot (1/m)$

基本思路: *加权平均*

若 $\frac{y}{1-y} > 1$ 则 预测为正例. \rightarrow 若 $\frac{y}{1-y} > \frac{m^+}{m^-}$ 则 预测为正例.

基本策略
—— “再缩放” (rescaling):

$$\frac{y'}{1-y'} = \frac{y}{1-y} \times \frac{m^-}{m^+}$$

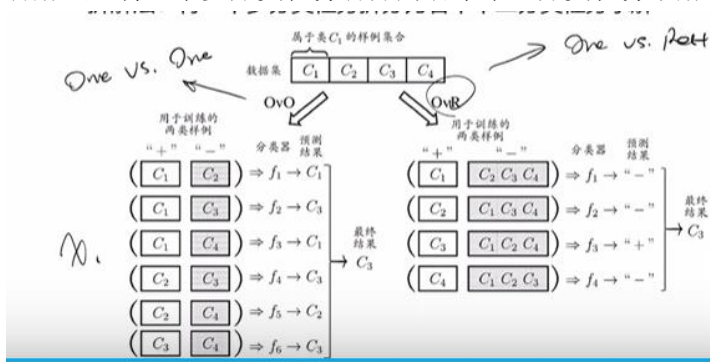
然而, 精确估计 m^-/m^+ 通常很困难!

常见类别不平衡学习方法:

- 过采样 (oversampling)
例如: SMOTE
- 欠采样 (undersampling)
例如: EasyEnsemble
- 阈值移动 (threshold-moving)

多分类学习

拆解法: 将一个多分类任务拆分为若干个二分类任务求解



多数情况下效果都差不多

讨论：

传统机器学习方法 vs. 当代大模型方法

1. 对于图像和文本数据，大模型处理很好，但是对于表格，时间序列处理不好（或者数据资源很少时）
2. 传统机器学习的核心思想会一直被使用，思路互通