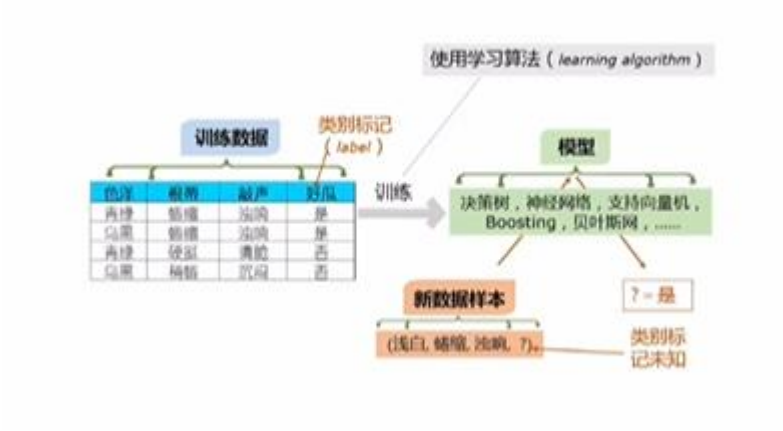
2024NJUNLP 夏令营-南京大学自然语言处理组

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】



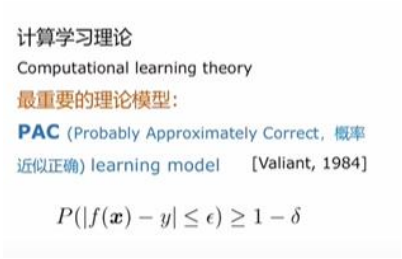
图上为监督学习

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

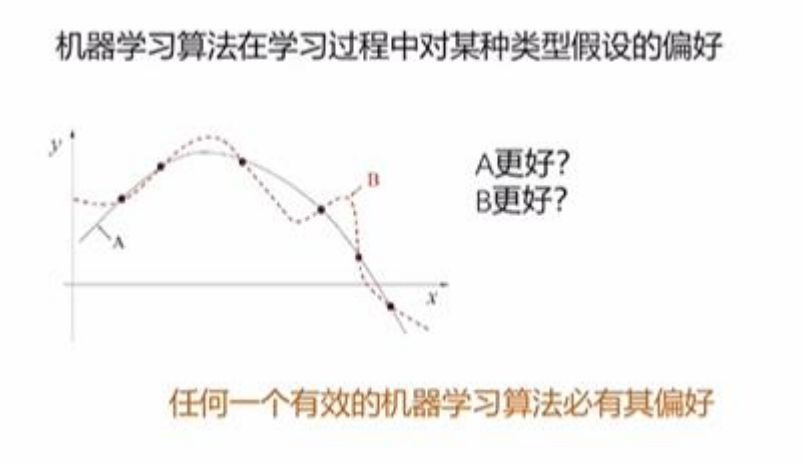
独立：数据之间没有相关性

典型的机器学习过程：如果泛化能力强就是一个好模型

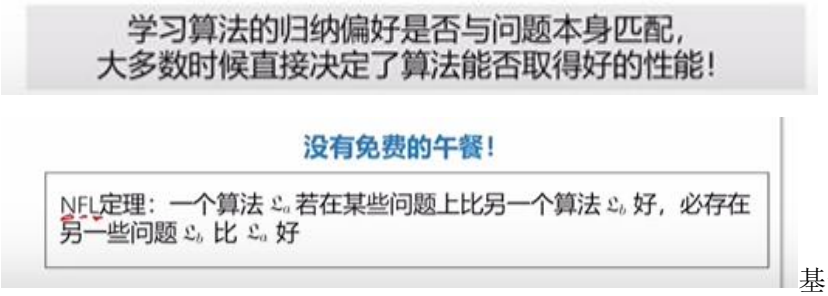


机器学习的理论基础：计算学习理论：PAC

归纳偏好（Inductive Bias）



一般原则：奥卡姆剃刀（Ocam’s Razor）



基于具体的任务

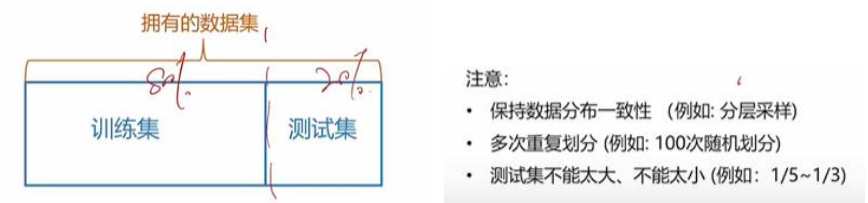
评估方法：

关键：怎么获得测试集（test set）

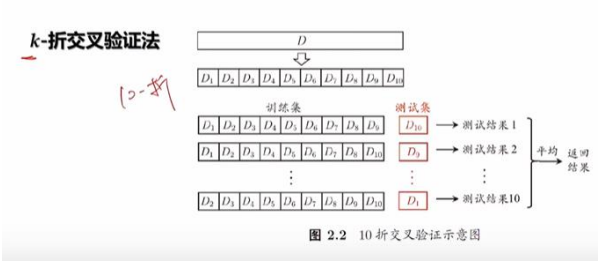
测试集应该与训练集互斥

常见方法：

1. 留出法



2. 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：

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

MSE（mean square error）

错误率 vs. 精度

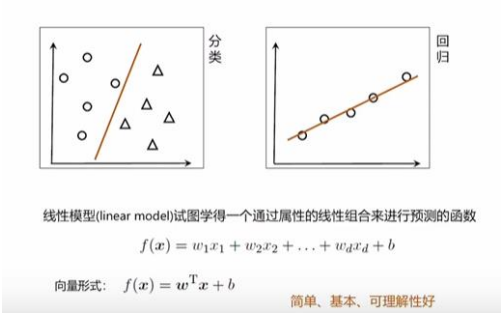
□ 错误率：

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

□ 精度：

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

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

$$f(\boldsymbol{x})=w_1x_1+w_2x_2+\ldots+w_dx_d+b$$

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

简单、基本、可理解性好

线性回归：

$f(x_i)=wx_i+b$ 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：

基本思路：优化模型的经验误差，同时控制模型复杂度

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \ell_{0/1}\left(y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right) - 1\right)$$

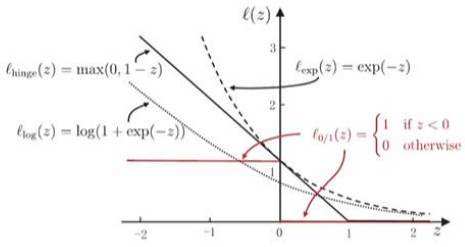
其中 $\ell_{0/1}$ 是 0/1损失函数 (0/1 loss function):

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

障碍：0/1损失函数非凸、非连续，不易优化

终极目标

替代损失
(Surrogate Loss)



- 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

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \max\left(0, 1 - y_i\left(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b\right)\right)$$

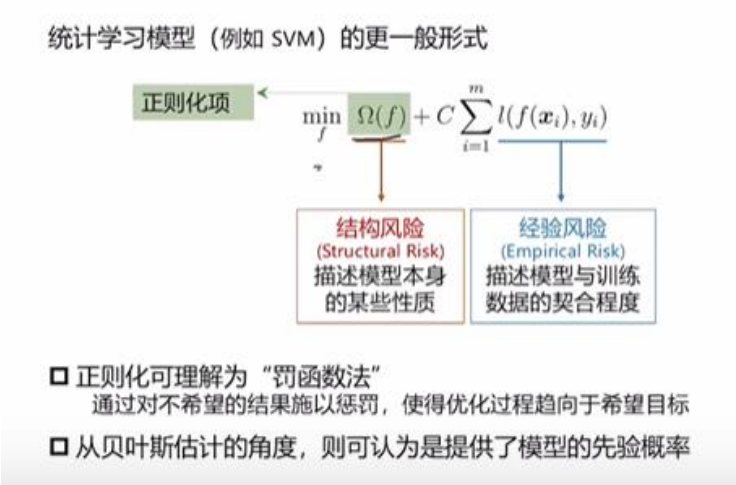
引入“松弛量”
(Slack Variables)

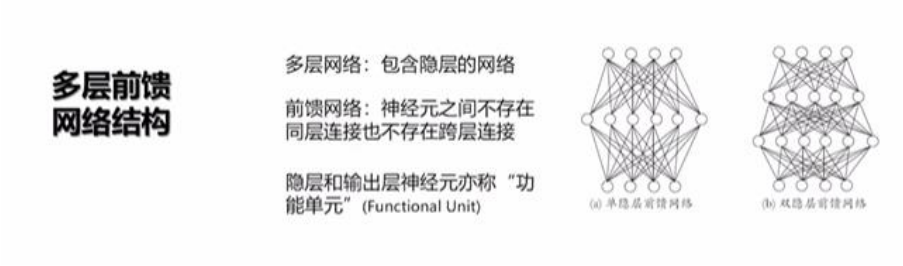
$$\min_{\boldsymbol{w},b,\xi_i} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i$$
$$\text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geqslant 1 - \xi_i \quad \xi_i \geqslant 0\ ,\ i = 1, 2, \ldots, m.$$

对偶问题

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{x}_j$$
$$\text{s.t. } \sum_{i=1}^{m} \alpha_i y_i = 0\ ,\ 0 \leqslant \alpha_i \leqslant C\ ,\quad i = 1, 2, \ldots, m\ .$$

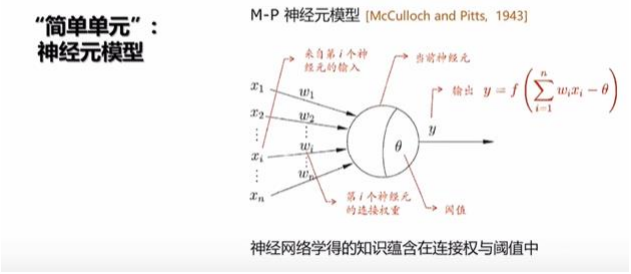
正则化：

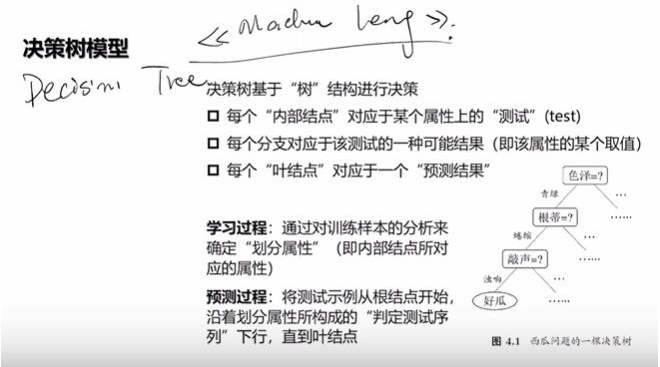




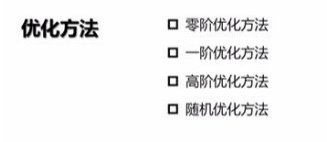
仅需一个包含足够多神经元的隐层，多层前馈神经网络就能以任意精度逼近任意复杂度的连续函数 [Hornik et al., 1989]

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”



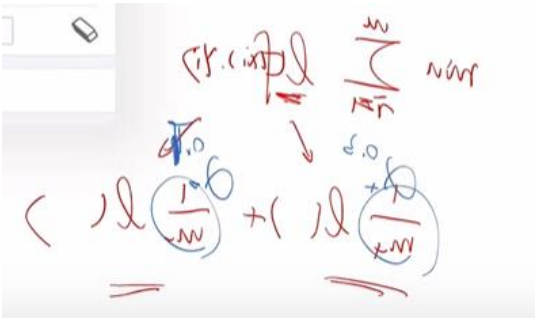


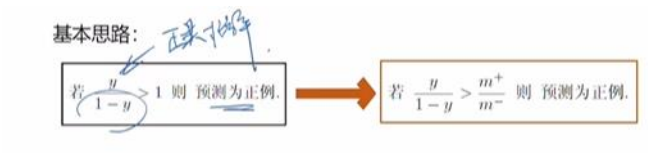
策略：分而治之

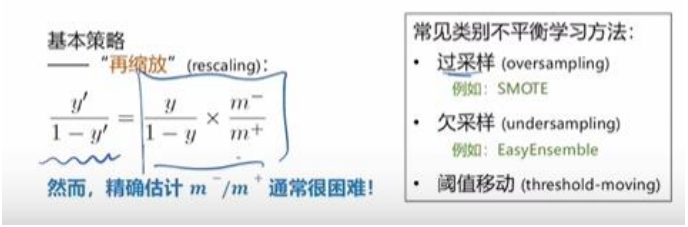


1. 无约束优化
2. X 在一个特点的域里（例子：SVM）

大类小类分别算：

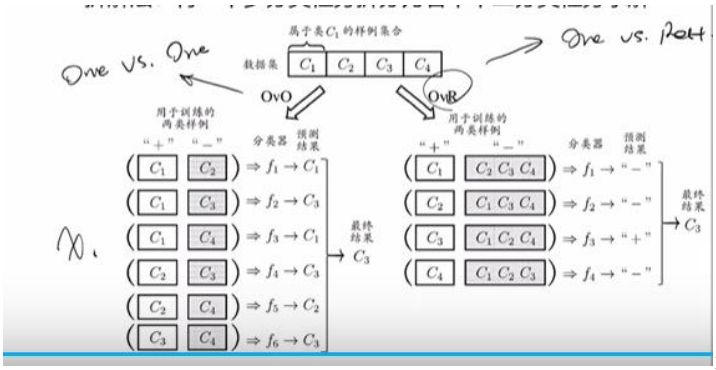






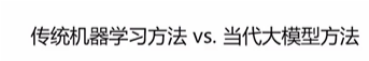
多分类学习

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：



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