

Lecture 2: 模型评估和选择

泛化能力

概念：什么是好的模型

- 引入：
 - 什么是好？
 - 搞清楚你要的到底是什么，然后针对需求、定义出好和坏；
 - 想搞清楚到底要什么，然后搞清楚我给你的到底是不是你要的；
- 概念：
 - 核心的就是：模型的**泛化能力是否强**；

泛化误差

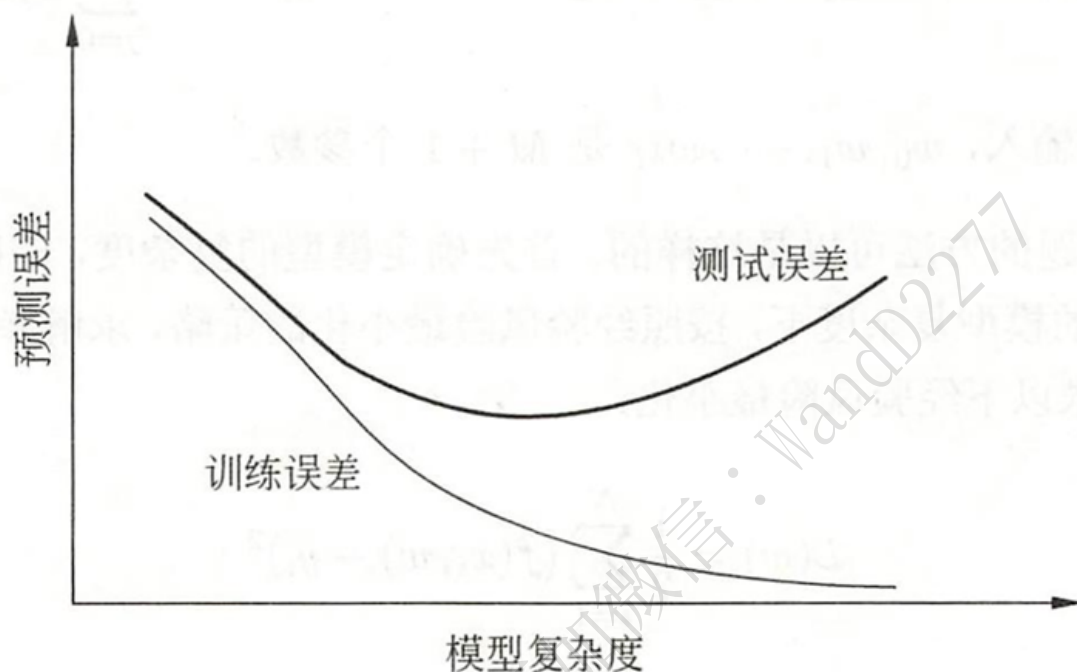
概念：泛化误差和经验误差

- 概念：泛化误差：
 - 描述：
 - 在“未来”样本上的误差；
 - 定义：
$$R_{\text{exp}}(\hat{f}) = E_P[L(Y, \hat{f}(X))] = \int_{\mathcal{X} \times \mathcal{Y}} L(y, \hat{f}(x)) P(x, y) dx dy$$
- 经验误差：
 - 已经给的数据（训练集）可以称之为经验；
 - 在训练集上的误差，亦称“训练误差”；
 - 我们总是在训练集上进行测试，然后得到某种结果，但是训练出的模型总是要用在未来、未见过的、非训练集上；
- 要求：
 - 未来的数据和现在的数据这两者之间有一些联系，未来的数据和现在的数据符合同一个规律；

概念：过拟合

- 引入：
 - 经验误差是否越小越好？
 - NO
 - 因为会导致过拟合；
- 概念：
 - 当型的复杂度增大时，训练误差会逐渐减小并趋向于 0。而测试误差会先减小，达到最小值后又增大；

- 当选择的模型复杂度过大时，过拟合现象就会发生；
- 为什么过拟合不好：因为会把训练样本的部分属性当成一般规律，应用在对未来的数据上；
- 所以训练误差并不是越小越好；
- 英文：
 - overfit
- 图示：



- 启发：
 - 过拟合是机器学习里面的核心内容，可以认为所有的机器学习算法都在某个环节存在过拟合问题；
 - 因此当我们在学习一个算法时，都可以问自己一个问题：这个算法是靠什么来缓解过拟合？缓解过拟合的策略在什么时候会失效？把这两个问题想清楚了、就可以知道这个算法该在什么时候用；

模型选择

概念：三个问题

- 如何获得测试结果？
 - 评估方法
- 如何评估性能优劣？
 - 性能度量 (到底要的是什么)
- 如何判断实质差别？
 - 比较检验

正则化

概念：正则化

- 概念：
 - 模型选择的典型方法是正则化 (regularization)；
 - 正则化是结构风险最小化策略的实现，是在经验风险上加一个正则化项 (regularizer) 或罚项 (penalty term)；
 - 正则化项一般是**模型复杂度的单调递增函数**，模型越复杂，正则化值就越大；
 - 比如，正则化项可以是模型参数向量的范数；

- 定义：

- 正则化一般具有如下形式：

- $$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)) + \lambda J(f)$$

- 第 1 项是经验风险，第 2 项是正则化项，拥有调整两者之间关系的系数；

- 描述：

- 第 1 项的经验风险较小的模型可能较复杂 (有多个非零参数)，这时第 2 项的模型复杂度会较大；
 - 正则化的作用是**选择经验风险与模型复杂度同时较小的模型**；
 - 正则化符合奥卡姆剃刀 (Occam's razor) 原理。奥卡姆剃刀原理应用于模型选择时变为以下想法：
 - 在所有可能选择的模型中，能够很好地解释已知数据、并且十分简单才是最好的模型，也就是应该选择的模型；
 - 从贝叶斯估计的角度来看，正则化项对应于**模型的先验概率**；
 - 可以假设**复杂的模型有较小的先验概率，简单的模型有较大的先验概率**；

其他评估方法

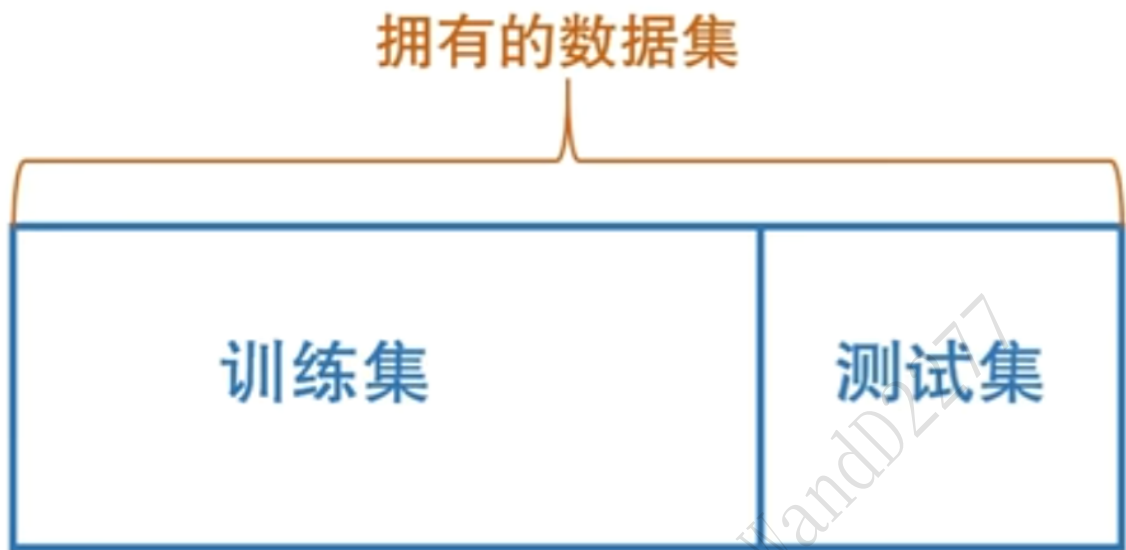
概念：怎么获得测试集

- 原则：
 - 测试集应该与训练集“互斥”；
- 方法：
 - 留出法 (hold-out)

- 交叉验证法 (cross validation)
- 自助法 (bootstrap)

概念：留出法

- 图示：



- 注意：

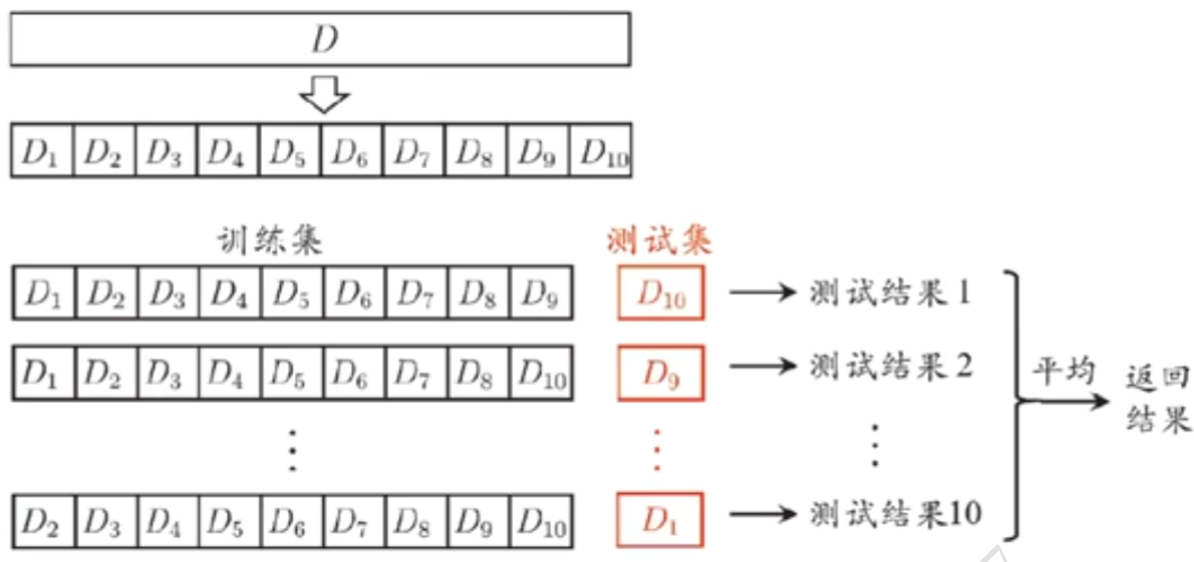
- (1) 保持数据分布一致性 (例如：分层采样)；
- (2) 多次重复划分 (例如：100 次随机划分)
- (3) 测试集不能太大、不能太小 (例如：1/5~1/3)
 - 假设训练出来的模型是 M_{100} ，假设其性能为 err_{80}
 - 假设数据集为 100，切分为 80 和 20，如果此时在 80 的数据上训练，得到 M_{80} ；
 - 此时这两者 M_{100} 和 M_{80} 已经不一样了；
 - 如果用 M_{80} 来逼近 err_{80} ，此时是有问题的；
 - 因此 M_X 中 X 不能太小，但 X 太小又可能导致 err_{80} 有问题；

- 问题：

- 训练集中可能有数据、拥有都没选上，而未来处理的问题、可能就是那一个漏检查的那个；

概念：k-折交叉验证法

- 图示：10 折交叉验证示意图



概念:

- 假设有数据 D ，分成 10 个完全一样大的子集合；
- 做十次测试，每一次测试从其中选择一个作为测试集，其余九个作为训练集；
 - 即用 M_{90} 去逼近 M_{100}
- 这样轮转、一共做十次；
- 这样所有的数据都被测试过，**没有漏掉的数据**；

分析:

- 假设：如果每次是用 99 个数据来进行训练，只留一个进行测试，即用 M_{99} 去逼近 err_{99} ，这样得到的结果和结果高度一致；
- 但这样不一定更准确，虽然训练的和真正要得到的模型很近似，但是测试可能会有偏差；
- 比如：假设当前有 50 个男生、50 个女生，进行训练。如果使用 M_{99} 进行训练，假设测试集是一个男生，则此时有 49 个男生、50 个女生，训练出来的结果是女生。但如果此时进来的是男生，则结果就完全错误，精度为 0；

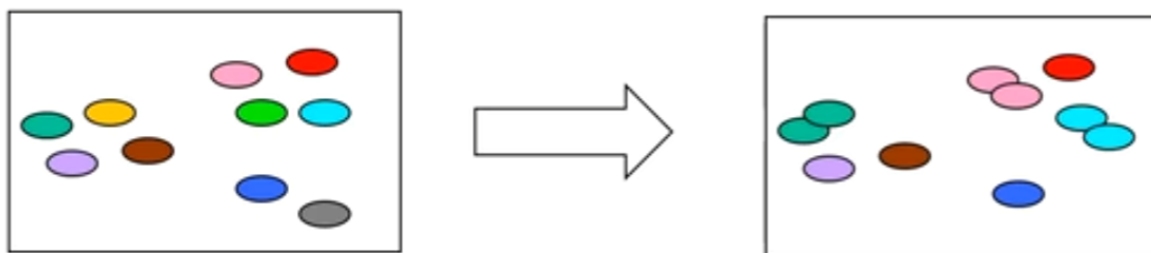
概念：自助法

引入:

- 假设一共只有 100 个数据，但希望训练的时候就得到 M_{100} ，但还要有剩下的数据进行测试，此时就可以使用自助法；

概念:

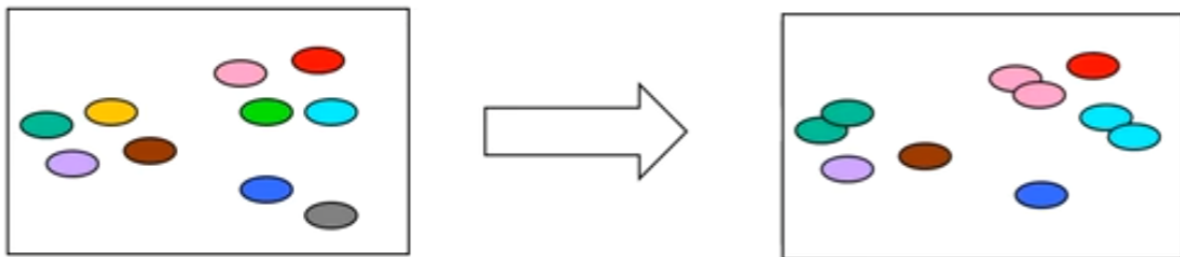
- 基于“自助采样”(bootstrap sampling)，亦称“有放回采样”、“可重复采样”；



- 在一个盒子里面、有 10 个数据，每次摸出一个球后，还会放回去；
- 有的样本可能会重复；
- 经过多次训练后、没使用过的数据，可以拿出来、用于测试；**

- 概念：包外估计

- m 个数据，进行 m 次估计，它不出现的可能性是 $1 - \frac{1}{m}$ ；
- 然后取了 m 次、因此要乘以 m 的次方；
- 当 m 非常大的时候，得到的结果就是 $\frac{1}{e}$ ，即约有 36.8% 的样本不出现；



约有 36.8% 的样本不出现

↓ $\lim_{m \rightarrow \infty} \left(1 - \frac{1}{m}\right)^m = \frac{1}{e} \approx 0.368$

“包外估计” (out-of-bag estimation)

- 此时就可以用这 36.8% 的样本作为测试集，即包外估计
- 缺点：
 - 它改变了训练数据的分布；

调参和验证集

概念：调参

- 算法的参数：
 - 一般由人工设定，亦称“超参数”
- 模型的参数：
 - 一般由学习确定
- 调参：
 - 调参过程相似：先产生若干模型，然后基于某种评估方法进行选择；
 - 比如 $y = ax^2 + bx + c$
 - 或者 $y = ax^3 + c$
 - 假设一个参数是当前多项式的次数，此时用户可以制定此参数、进而选择不同的多项式；
 - 调参数本质就是一种模型的选择，参数调得好不好对性能往往对最终性能有关键影响；
 - 每次调参数就是对模型进行选择，选择不同的算法也是进行模型选择；

概念：训练集 vs. 测试集 vs. 验证集 (validation set)

- 从数据集中切除一部分来进行测试，构成测试集。测试是为了看模型的性能；
- 在有了调参数的需求后，此时可以从训练集中切分出一部分来，专门用于调参数，其实也是一种测试集，只不过是用于验证参数是否是好的；
- 算法参数选定后，要用“训练集+验证集”重新训练最终模型；

- 即：
 - |60 训练集|20 验证集|20 测试集|

性能度量

概念： 训练集的平均损失与训练误差

- 概念： 训练集的平均损失
 - 假设学习到的模型是 $Y = \hat{f}(X)$, 训练误差是模型 $Y = \hat{f}(X)$ 关于训练数据集的平均损失：

$$R_{\text{emp}}(\hat{f}) = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i))$$

- 概念： 训练误差
 - 测试误差是模型 $Y = \hat{f}(X)$ 关于测试数据集的平均损失：

$$e_{\text{test}} = \frac{1}{N'} \sum_{i=1}^{N'} L(y_i, \hat{f}(x_i))$$

- 举例： 损失函数是 0-1 损失：
 - 误差率：

$$e_{\text{test}} = \frac{1}{N'} \sum_{i=1}^{N'} I(y_i \neq \hat{f}(x_i))$$

- 准确率：

$$r_{\text{test}} = \frac{1}{N'} \sum_{i=1}^{N'} I(y_i = \hat{f}(x_i))$$

概念： 性能度量

- 概念：
 - 怎么样的模型可能是好的；
 - 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；
 - 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；
- 分析： 对于回归任务
 - 回归 (regression) 任务常用均方误差；

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

概念： 错误率和精度

- 概念： 错误率
 - m 个样本，有多少次做的和目标不一样 (y)，即错误率；

-

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

- 概念：精度

-

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

比较检验

概念：比较检验

- 引入：
 - 在某种度量下取得评估结果后，是否可以直接比较以评判优劣？
 - 不能；
 - 因为测试性能不等于泛化性能，测试性能随着测试集的变化而变化，很多机器学习算法本身有一定的随机性；
 - 因此不能简单的通过一个值来进行判断；
 - 并且机器学习是“以概率近似正确”；
- 概念：
 - 统计假设检验 (hypothesistest) 为学习器性能比较提供了重要依据；
- 概念：两学习器比较
 - 交叉验证 t 检验 (基于成对 t 检验)；
 - k 折交叉验证； 5 ** 2 交叉验证；
 - McNemar 检验 (基于列联表，卡方检验)

分类问题

概念：分类问题的性能标准

- 概念：分类准确率
 - 评价分类器性能的指标一般是分类准确率 (accuracy)；
 - 对于给定的测试数据集，分类器正确分类的样本数与总样本数之比；
 - 也就是损失函数是 0-1 损失时测试数据集上的准确率；
 - 对于二类分类问题，常用的评价指标是精确率 (precision) 与召回率 (recall)；
- 概念：查准率和查全率
- 图示：
 - 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)

-
- FP : 模型说是正例, 实际的结果是反例;
- TN : 本来就是反例, 预测出来也是反例;
- 反对角线部分是错的;

- 概念: 查准率

- $$P = \frac{TP}{TP + FP}$$

- 查全率:

- $$R = \frac{TP}{TP + FN}$$

- 所有的正样本里面, 有多少被你找出来了;
- FN 是你预测为假的, 实际是真的;
- TP 就是预测为真的, 实际是真的;
- 因此 $TP + FN$ 就是全部的正确样本;

- 概念: $F1$ 指标

- $$F1 = \frac{2 \times P \times R}{P + R}$$

$$= \frac{2 \times TP}{\text{样例总数} + TP - TN}$$

- $$\frac{1}{F1} = \frac{1}{2} \cdot \left(\frac{1}{P} + \frac{1}{R} \right)$$

- 即: P 和 R 的调和平均;
- 调和平均可以保证: 较小的值也不会被忽视掉;

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

- $$\frac{1}{F_{\beta}} = \frac{1}{1 + \beta^2} \cdot \left(\frac{1}{P} + \frac{\beta^2}{R} \right)$$

- $\beta > 1$ 时查全率有更大影响;

- $\beta < 1$ 时查准率有更大影响

获取笔记持续更新可加微信: WandD2277