第十一章 集成学习
（Ensemble Learning）

卿来云

集成学习

- 我们已经开发了很多机器学习算法/代码。

- 1. 对给定的任务，如何评价模型的性能？如何得到最佳的模型？
  - 模型评价指标
  - 超参数调优

- 2. 单个模型的性能已经调到最优，很难再有改进。
  - 集成学习：用很少量的工作，组合多个基模型，使得系统性能提高

Outline

- 模型性能评价
  - No Free Lunch Theorems
  - Occam剃刀原理
  - 校验集/交叉验证
- Bagging
  - 随机森林
- Boosting
  - AdaBoost
  - Gradient Boosting Decision Tree (GBDT)
  - XGBoost
  - LightGBM
- Stacking和blending

Recall: 机器学习定义

- 机器学习：对于某类任务T和性能度量P，如果一个计算机程序在T上以P衡量的性能随着经验E而自我完善，那么我们称这个计算机程序在从经验E学习。(Tom M. Mitchell)

No Free Lunch（NFL）Theorem

- All algorithms are equivalent, on average, by any of the following measures of error: $\mathbb{E}(L|f,\mathcal{D})$, where
  - $\mathbb{E}$：期望
  - $\mathcal{D}$ = training set;
  - f = 'target' input-output relationships; and
  - L = off-training-set 'loss' associated with f

- 没有一个学习算法可以在任何领域总是产生最准确的学习器。

David H. Wolpert and William G. Macready, No Free Lunch Theorems for Optimization, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 1, NO. 1, APRIL 1997

No Free Lunch Theorem

- NFL定理的重要前提：所有问题出现的机会相同，或所有问题同等重要。所以脱离具体问题，空泛地谈论”什么学习算法更好” 毫无意义。
- 从模型的角度看，一个特定的模型必然会在解决某些问题时误差较小，而在解决另一些问题时误差较大；
- 从问题的角度看，在解决一个特定的问题时，必然有某些模型具有较高的精度，而另一些模型的精度就没那么理想了。
- NFL定理最重要的指导意义在于先验知识的使用，即具体问题具体分析。机器学习的目标不是放之四海而皆准的通用模型，而是关于特定问题有针对性的解决方案。因此在模型的学习过程中，一定要关注问题本身的特点，也就是关于问题的先验知识。只有当模型的特点和问题匹配时，模型才能发挥最大的作用。

No Free Lunch Theorem

- NFL定理可以进一步引出一个普适的守恒率 ： 对每个可行的学习算法来说，它的性能对所有可能的目标函数的求和结果为零。即我们要想在某些问题上得到正的性能的提高，必须在另一些问题上付出等量的负的性能的代价！比如时间复杂度和空间复杂度。

奥卡姆剃刀（Occam's Razor）原理

- “Entities” (or explanations) should not be multiplied beyond necessity. 如无必要，勿增实体
- Among competing hypotheses, the one with the fewest assumptions should be selected.
- For PR/ML, NOT use machines that are more complicated than necessary.
  - “necessary” can be determined by the quality of fit to the training data.

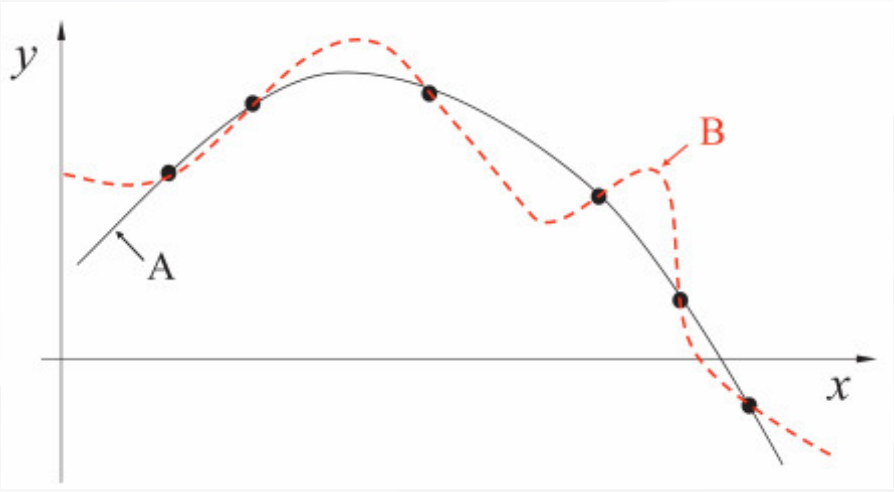
奥卡姆剃刀（Occam's Razor）原理

- 奥卡姆剃刀原理的关注点是模型复杂度。
- 机器学习模型应该能够识别出数据背后的模式。
  - 当模型本身过于复杂时，特征和类别之间的关系中所有的细枝末节都被捕捉，主要的趋势反而在乱花渐欲迷人眼中没有得到应有的重视，导致过拟合（overfitting）的发生。
  - 反之，如果模型过于简单，它不仅没有能力捕捉细微的相关性，甚至连主要趋势本身都没办法抓住，这样的现象就是欠拟合（underfitting）。

例：

■训练数据和模型A&B

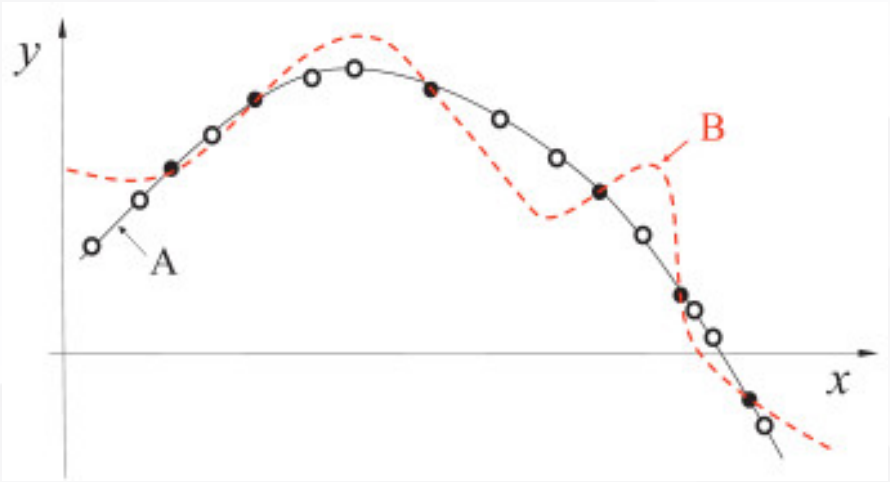
- A线和B线都能够很好的拟合这几个数据点。
- 哪条曲线更好？



仅仅从这几个数据点来看，我们无法判断哪个更好，或者说，A和B一样好。

例：

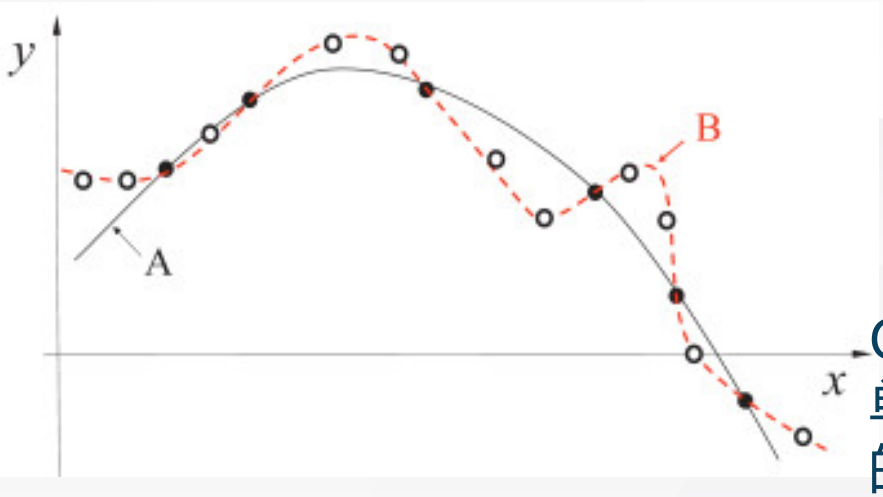
- 更多测试数据1（空心点）
  - A更好



例：

■更多测试数据2（空心点）

- B更好

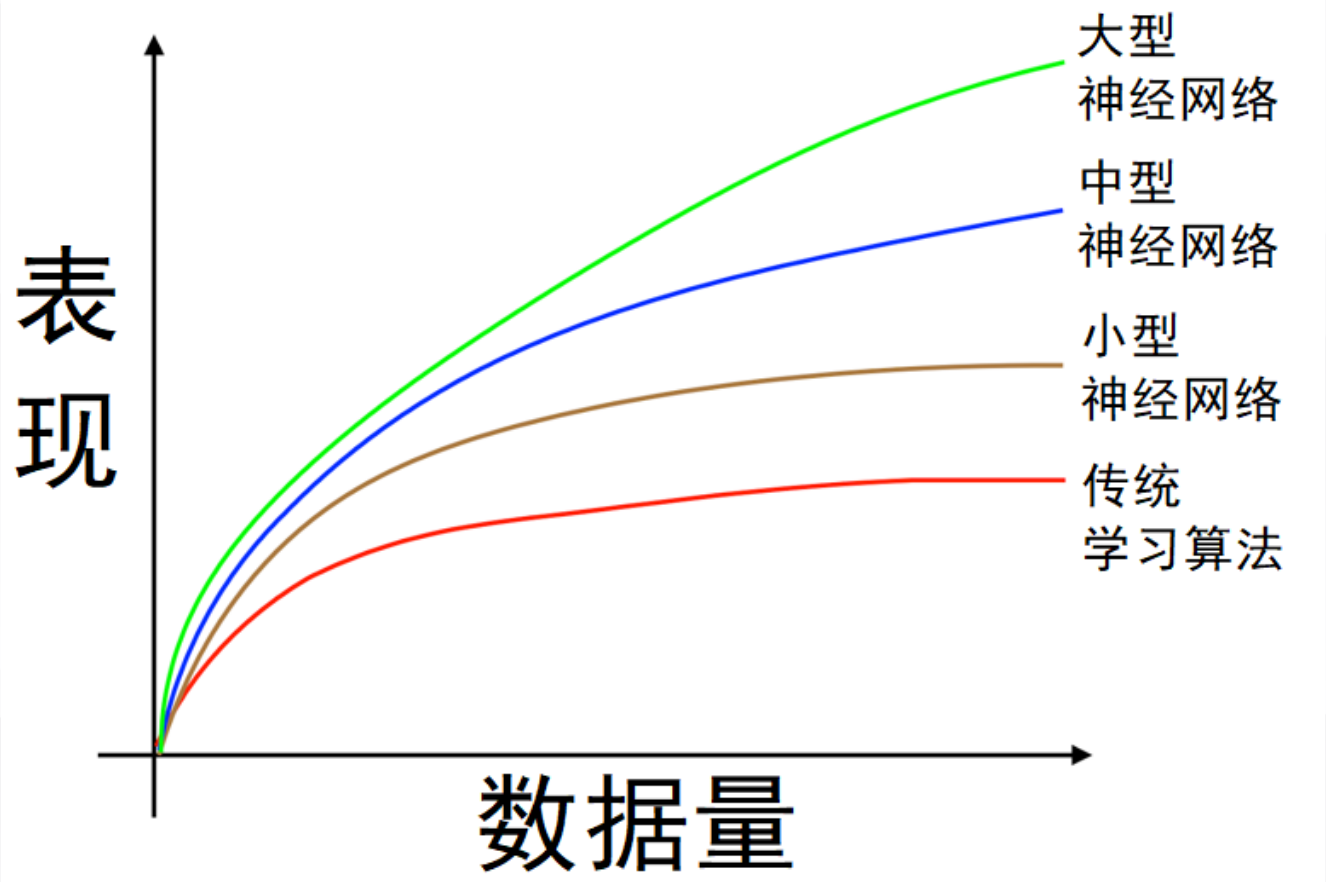


NFL：具体哪一个函数更好，取决于数据本身的规律。而这个规律，从有限的观测数据中，是不可能绝对准确把握的。

Occam ‘s Razor：A更好，因为它足够简单，且拟合得足够好。这是因为我们所面临的多数问题并不复杂，通常使用比较简单的方法就可以取得很好的效果。

实际应用技巧

1.训练集的大小



Andrew Ng：machine-learning-yearning （机器学习训练秘籍）
https://github.com/deeplearning-ai/machine-learning-yearning-cn

实际应用技巧

- 2. 数据划分
  - 训练集：训练模型参数
  - 验证集/开发集：用于参数调优、特征选择、及对学习算法作出其它决
    - 可能需要通过交叉验证的方式获取
  - 测试集：用于评估算法的性能，但不会据此改变学习算法或参数

开发集和测试集应该服从同一分布

选择合适的开发集和测试集，使之能够代表将来实际数据的情况
开发集和测试集应该服从相同的分布

实际应用技巧

- 3. 分析误差并迭代
  - 不要在一开始就试图设计和构建一个完美的系统
  - 相反，应尽可能快地构建和训练一个系统雏形
  - 使用误差分析法去识别出最有前景的方向，并据此不断迭代改进算法

实际应用技巧

- 4. 偏差和方差：误差的两大来源
- 学习曲线（Learning Curves）：不同训练集大小对应的训练集和验证集的性能
  - 用学习曲线观察机器学习算法是否为欠拟合或过拟合，
  - 亦可用于诊断偏差与方差。

学习曲线（Learning Curve）

- 学习曲线：通过画出不同训练集大小时训练集和验证集的性能
  - •横轴：训练样本的数量
  - •纵轴：模型性能
  - •train_sizes, train_scores, validation_scores = learning_curve (*estimator, X, y, *, groups=None, train_sizes=array([0.1, 0.33, 0.55, 0.78, 1.]), cv=None, scoring=None, exploit_incremental_learning=False, n_jobs=None, pre_dispatch='all', verbose=0, shuffle=False, random_state=None, error_score=nan, return_times=False*)

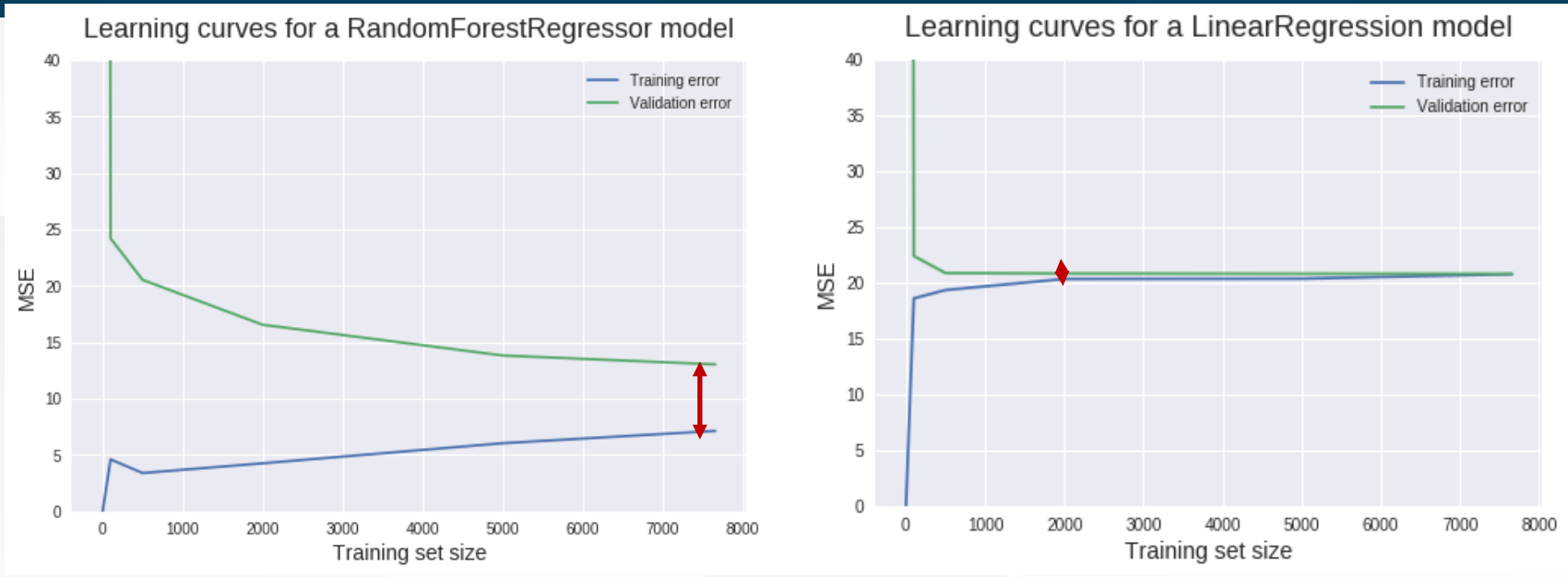
    返回值：训练集大小、训练集和验证集上的误差得分

    参数：学习器、数据、训练集大小、交叉验证参数、评价指标

训练误差随训练集增大而增大，然后趋于稳定
验证误差随训练集增大而减少，然后趋于稳定
二者之间的差异随随训练集增大而减少，然后趋于稳定

不同模型区域稳定的样本集合大小不同（简单模型需要更少的训练数据）

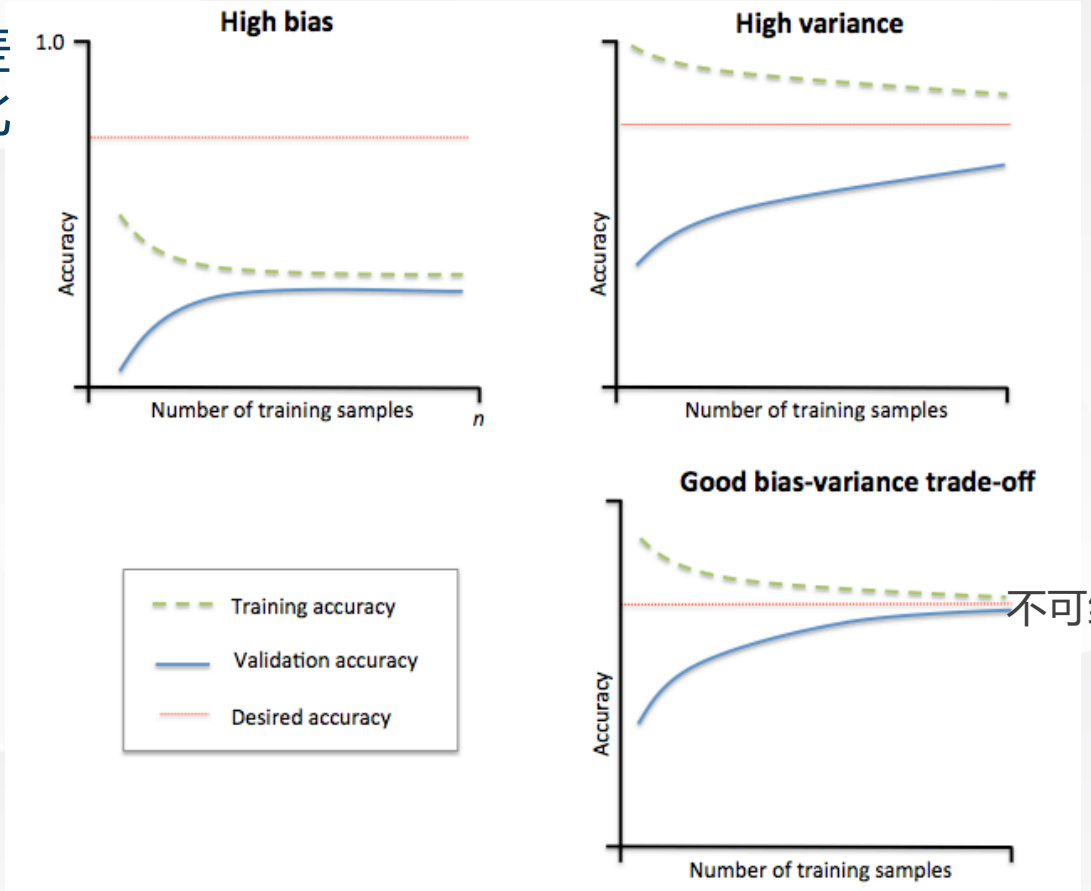
学习曲线



- 偏差：当训练误差稳定时，此时训练误差的大小可视为模型偏差（训练数据充分时，模型与训练数据的拟合程度）
  - 随机森林偏差小、线性模型偏差大
- 方差：当训练误差稳定时，训练误差与验证误差之间的差异可视为模型的方差（由于数据不同模型性能的差异）
  - 随机森林方差大、线性模型方差小

学习曲线（Learning Curve）

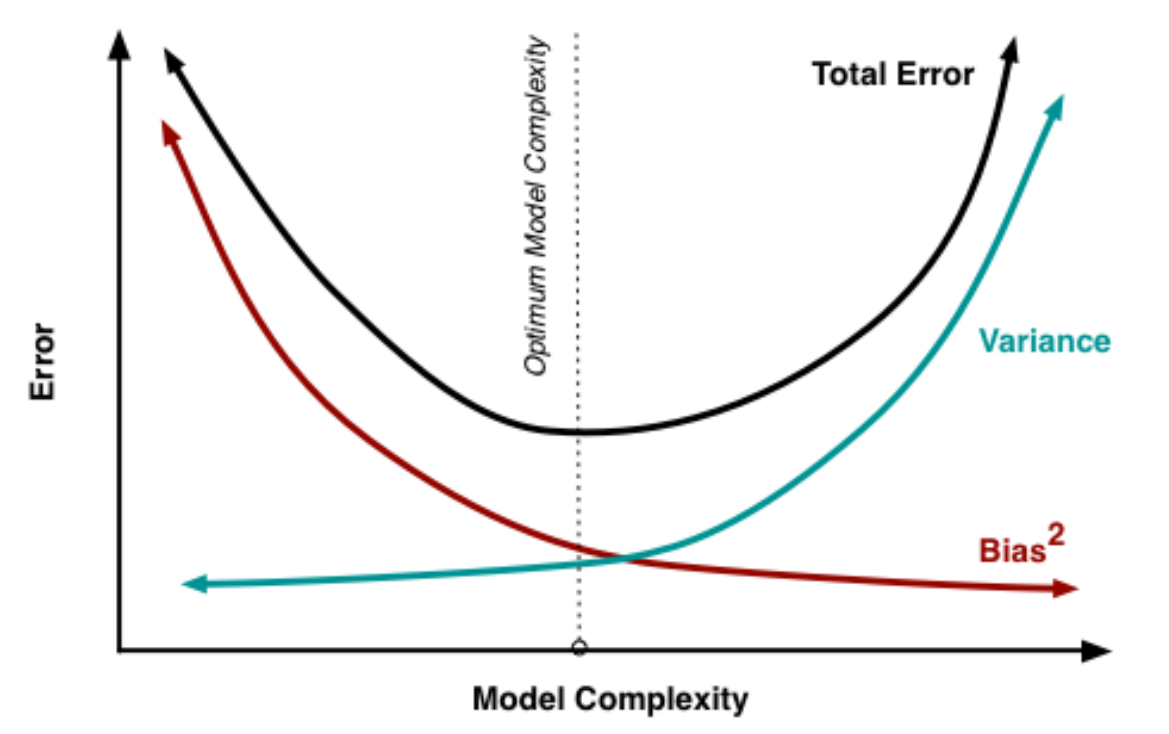
验证集和训练集的误差值都很大，偏差大，此时为欠拟合



训练集误差非常下，但验证集误差远大于训练集误差，此时为过拟合

模型复杂度、偏差、方差

- 选择合适复杂度的模型

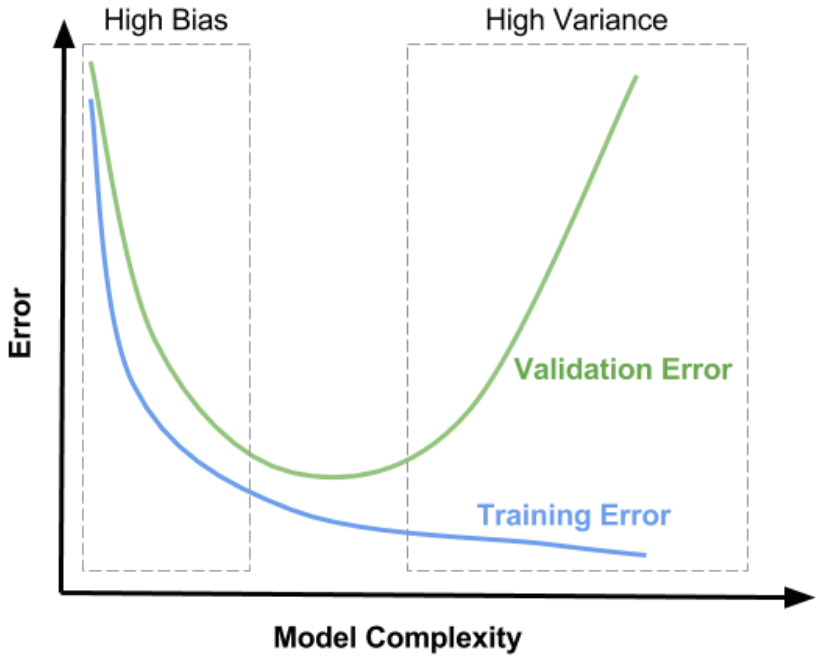


欠拟合和过拟合的外在表现

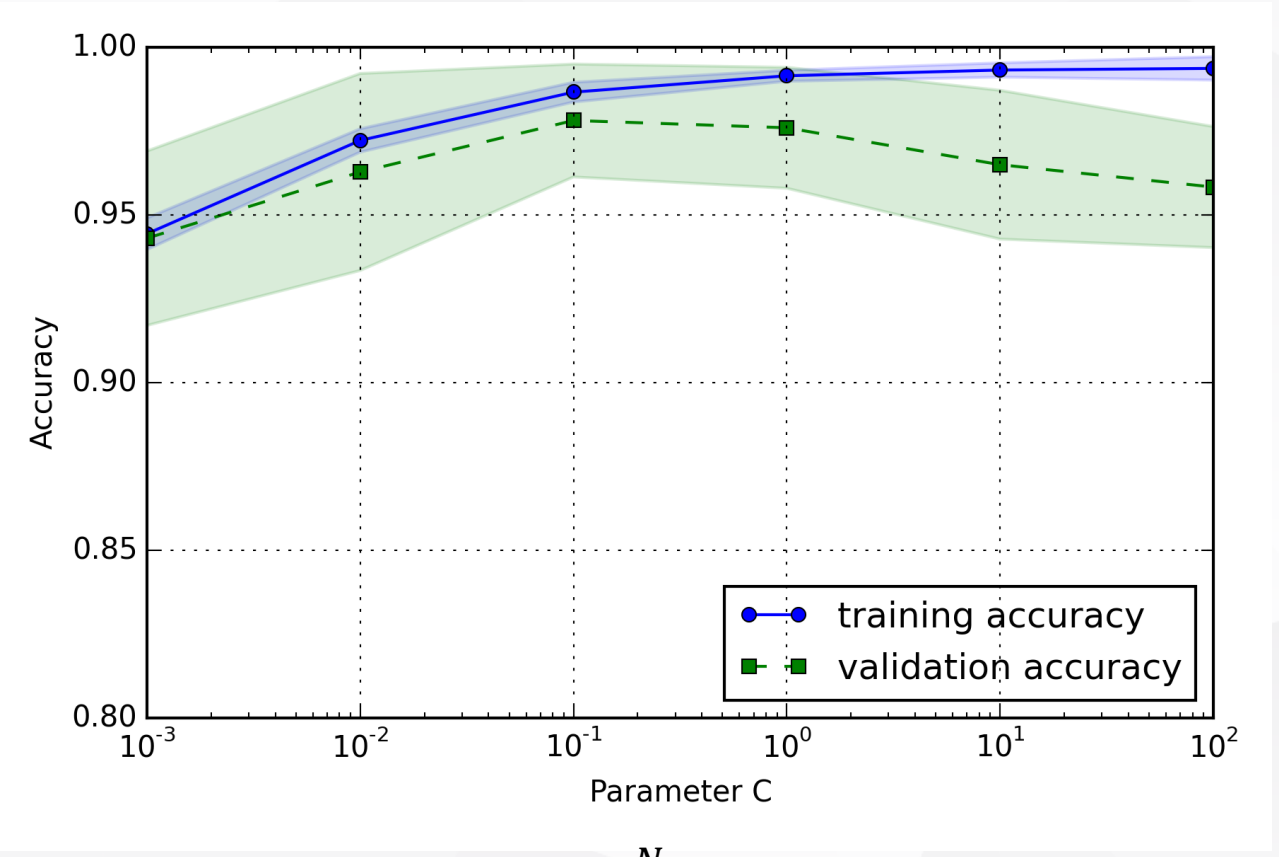
- 在实际样应用中，有时候我们很难计算模型的偏差与方差，只能通过外在表现，判断模型的拟合状态是欠拟合还是过拟合。

训练误差随着模型复杂度增加一直减小。

校验误差随着模型复杂度的变化先减小（欠拟合程度减轻）；
当模型复杂度超过一定值后，校验误差随模型复杂度增加而增大，此时模型进入过拟合状态。



模型复杂度



$$J(\mathbf{w}, b; C) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N} L_{Hinge}(y_i, f(\mathbf{x}_i; \mathbf{w}, b))$$

提高模型性能

- 欠拟合：当模型处于欠拟合状态时，根本的办法是增加模型复杂度。
  - 修改模型架构（增大模型规模）
  - 增加模型的迭代次数
  - 更多特征
  - 降低模型正则化水平（L2、L1、Dropout）
- 过拟合：当模型处于过拟合状态时，根本的办法是降低模型复杂度。
  - 修改模型架构（减小模型规模）
  - 及早停止迭代
  - 减少特征数量
  - 提高模型正则化水平
  - 扩大训练集：可以帮助解决方差问题，但对偏差通常没有明显影响

小结

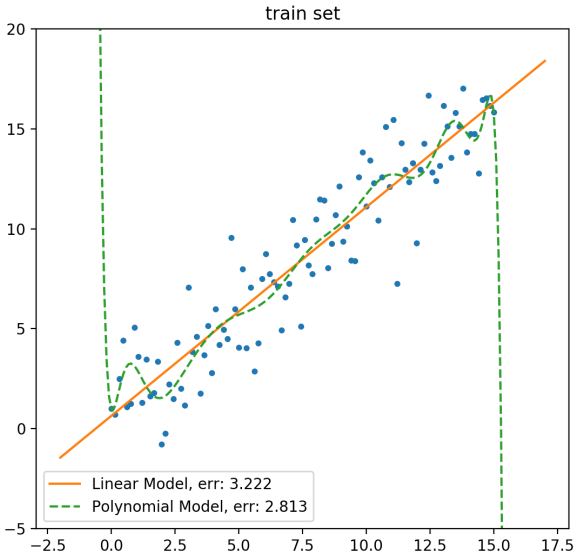
- 无免费午餐定理：模型的选取要以问题的特点为根据。
- 奥卡姆剃刀：在性能相同的情况下，应该选取更加简单的模型。
- 过于简单的模型会导致欠拟合，过于复杂的模型会导致过拟合。
- 从误差分解的角度看，欠拟合模型的偏差较大，过拟合模型的方差较大。

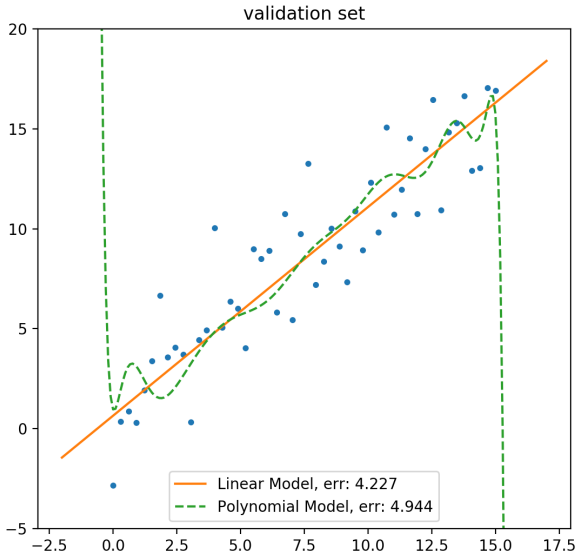
偏差-方差平衡

- 通常：简单的模型偏差高、方差低；复杂的模型偏差低、方差高
- 例：$y = x + x^{0.01} + \varepsilon$，$\varepsilon \sim \mathcal{N}(0,2)$，用线性模型和15阶多项式拟合

训练集上，线性模型的误差要明显高于多项式模型。

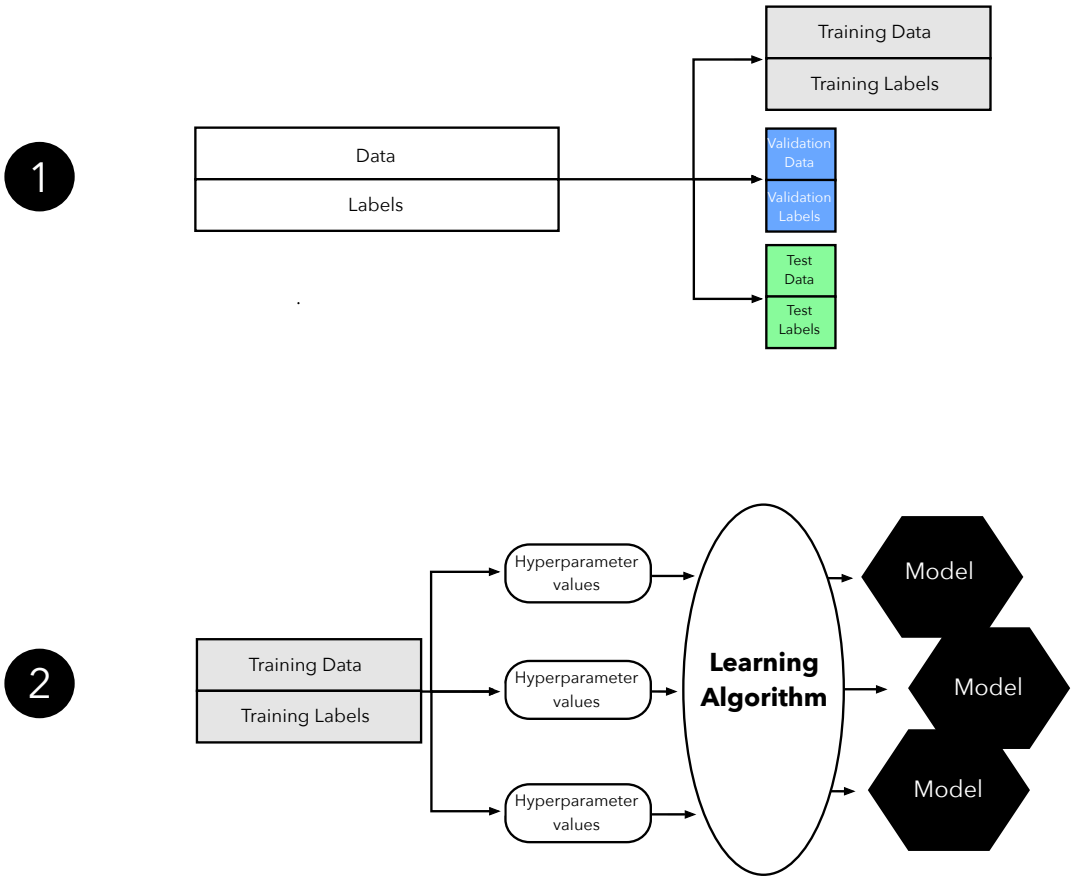
线性模型在训练集上欠拟合，偏差高于多项式模型的偏差。





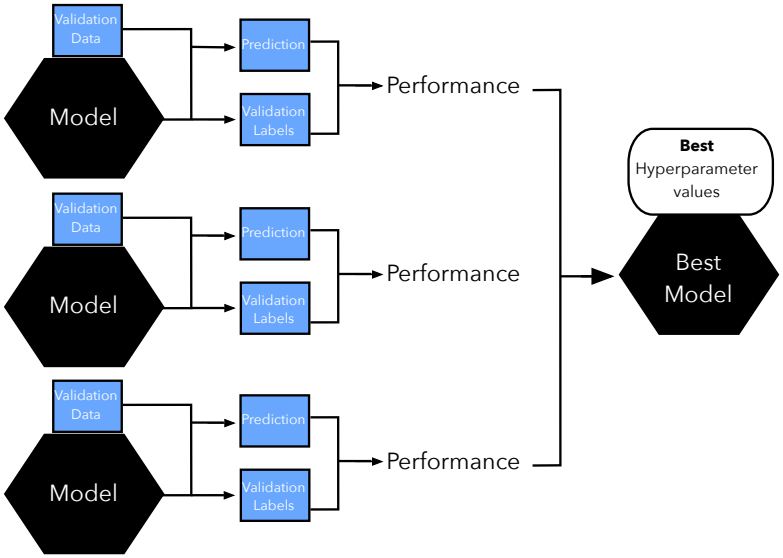
验证集上，线性模型的误差小于多项式模型的误差，且线性模型在训练集和验证集上的误差相对接近，泛化能力更好。而多项式模型在两个数据集上的误差差距很大。
多项式模型在训练集上过拟合，方差高于线性模型的方差。

模型校验——校验集

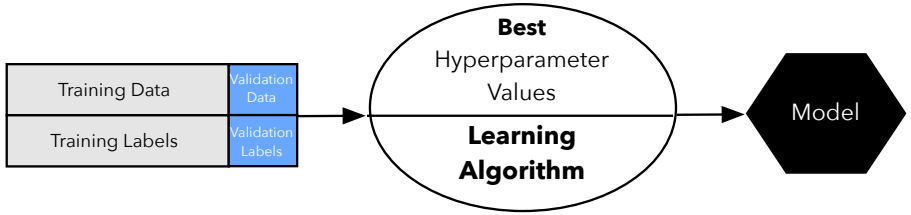

1
Data
Labels
Training Data
Training Labels
Validation Data
Validation Labels
Test Data
Test Labels
2
Training Data
Training Labels
Hyperparameter values
Hyperparameter values
Hyperparameter values
Learning Algorithm
Model
Model
Model

模型校验——校验集

3

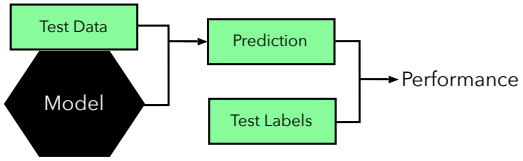

Validation Data
Model
Prediction
Validation Labels
Performance
Validation Data
Model
Prediction
Validation Labels
Performance
Validation Data
Model
Prediction
Validation Labels
Performance
Best Hyperparameter values
Best Model

4

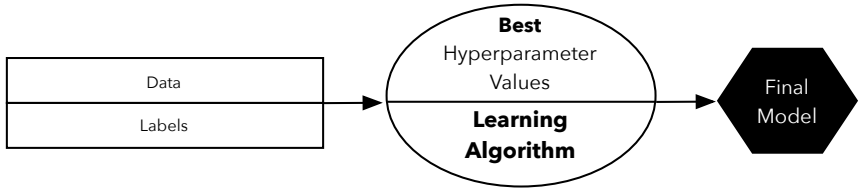

Training Data
Validation Data
Training Labels
Validation Labels
Best Hyperparameter Values
Learning Algorithm
Model

模型校验——校验集

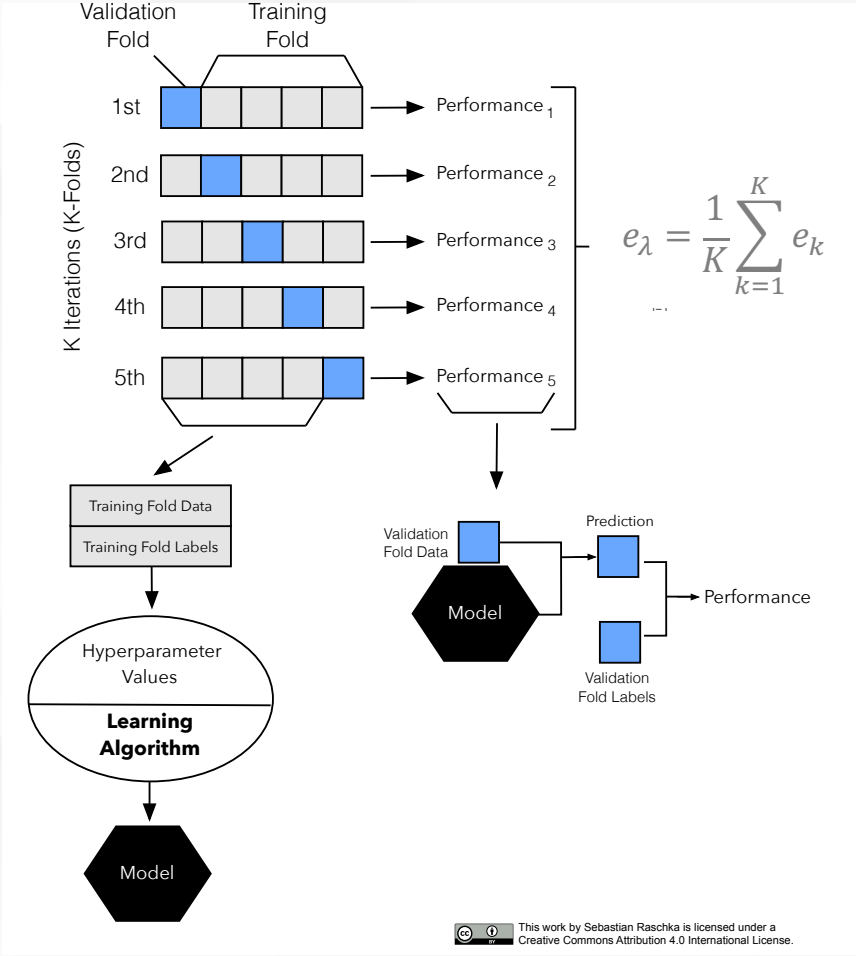
5



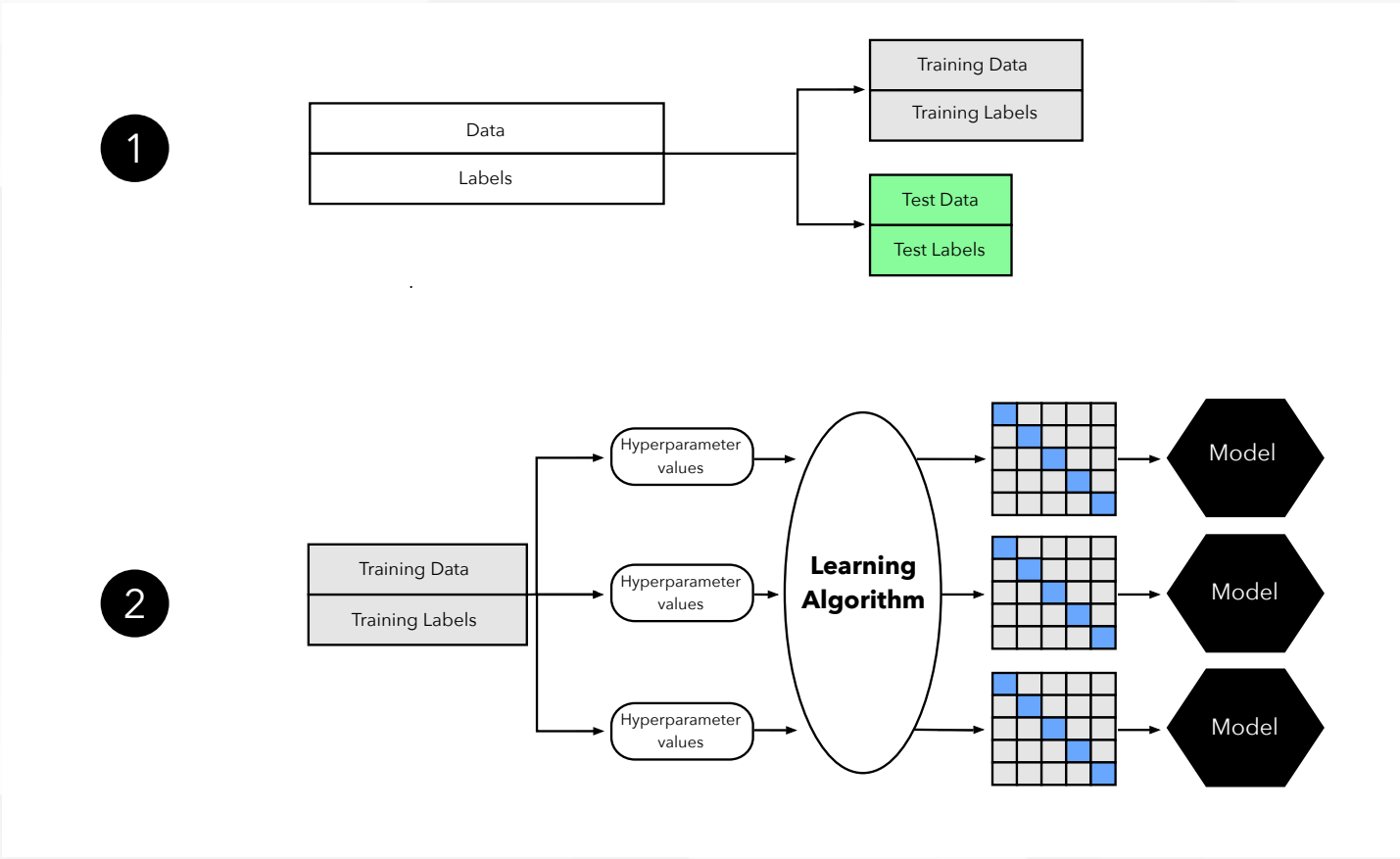
6



模型校验——K-fold Cross-Validation



K-fold Cross-Validation Pipeline I



K-fold Cross-Validation Pipeline II