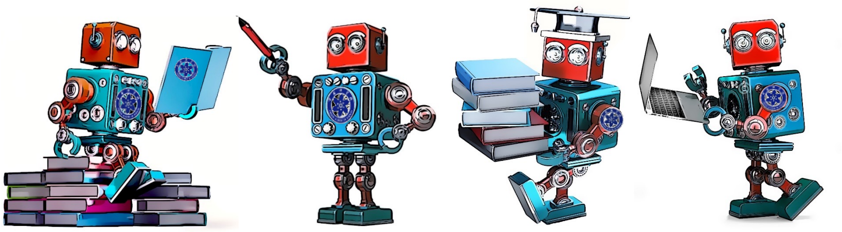
模式识别与机器学习

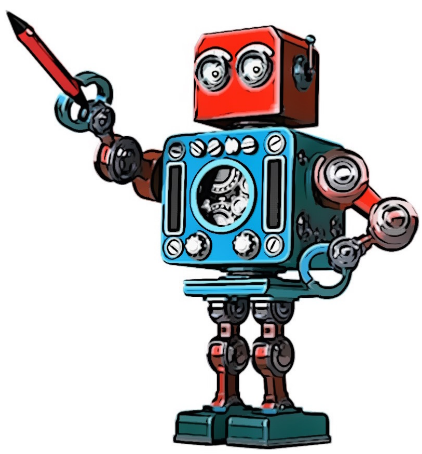
集成学习



苏荔
suli@ucas.ac.cn

1

- 简介
- 模型性能评价
- Resampling
- Bagging
- Boosting
- Stacking



2021/12/10 2

2

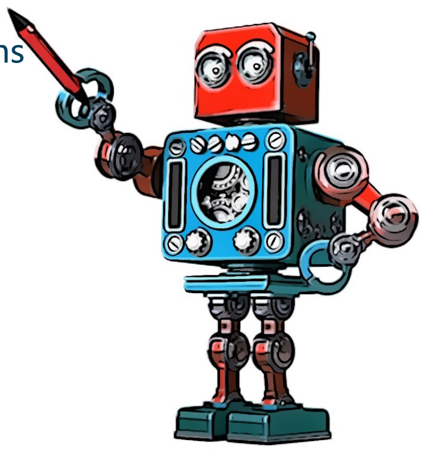
集成学习

- 我们已经开发了很多机器学习算法/代码。
- 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
  – 基模型最好变化多样，这样不同的基模型集成后形成互补。

3

- Introduction
- 模型性能评价
  - No Free Lunch Theorems
  - Occam剃刀原理
  - 偏差-方差折中
  - 校验集/交叉验证
- Resampling
- Bagging
- Boosting
- Stacking



4

4

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

https://en.wikipedia.org/wiki/Tom_M._Mitchell

5

No Free Lunch Theorem

- Wolpert, 1996

无噪声、无先验知识

"In a noise-free scenario where the loss function is the misclassification rate, if one is interested in off-training-set error, there are no a priori distinctions between learning algorithms."

- 没有任何学习算法可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

6

No Free Lunch Theorem

- 对于以下的各种误差定义，总体来看，所有算法都一样（无法选出最优的）：$\mathbb{E}(L|\mathcal{D})$, $\mathbb{E}(L|f,\mathcal{D})$, $\mathbb{E}(L|n)$, $\mathbb{E}(L|f,n)$
  - $\mathbb{E}$：期望
  - $\mathcal{D}$ = training set;
  - n = number of elements in training set;
  - f = 'target' input-output relationships;
  - h = hypothesis (the algorithm's guess for f made in response to $\mathcal{D}$); and
  - L = off-training-set 'loss' associated with f and h ('generalization error')
- 没有一种算法比随机乱猜的效果更好？
  All models are wrong， but some are useful.

7

No Free Lunch Theorem

- NFL定理有个重要前提：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“哪种学习算法更好”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
- 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。

8

No Free Lunch Theorem

- 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。
- 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。

- 没有任何先验知识时，理论上无法找到最优的模型。
  那么，是否能找到度量未知模式之间近似程度的最优方法？

9

丑小鸭定理

- 渡边慧(Watanabe)， 1969：“丑小鸭与白天鹅之间的区别和两只白天鹅之间的区别一样大” *
- 世界上不存在分类的客观标准，一切分类的标准都是主观的。
  - 鲸鱼的例子：
    - 生物学分类：鲸鱼属于哺乳类的偶蹄目，和牛是一类
    - 产业界分类：鲸和鱼同属于水产业，而不属于包括牛的畜牧业。

- 分类结果取决于选择什么特征作为分类标准，而特征的选择又依存于人的目的(隐含假设， implicit assumptions)

* Watanabe, Satosi, Knowing and Guessing: A Quantitative Study of Inference and Information. New York: Wiley. (1969). pp. 376–377.

10

10

奥卡姆剃刀（Occam's Razor）原理

- 公元 14 世纪，圣方济各会修士，Occam Philosophy Principle： “Entities” (or explanations) should not be multiplied beyond necessity.

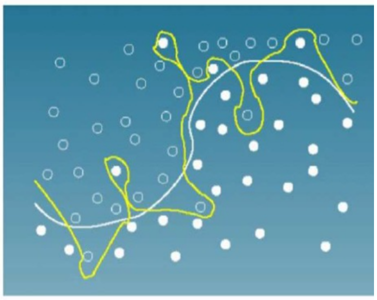
“如无必要，勿增实体”

- 在各种候选的假设中，应选择假设最少的假设：“大道至简”
- 对于PR/ML而言，必要性“necessary”可以用对训练集的拟合程度来度量：过拟合？欠拟合？

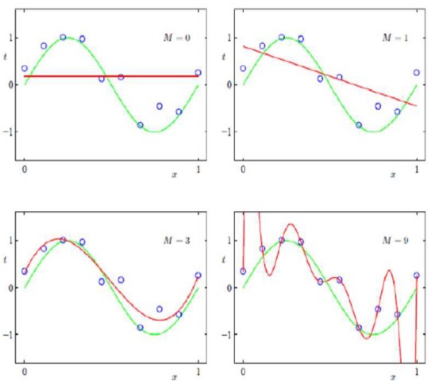
11

过拟合 vs 欠拟合

Overfitting-Classification



Overfitting-Regression



12

12

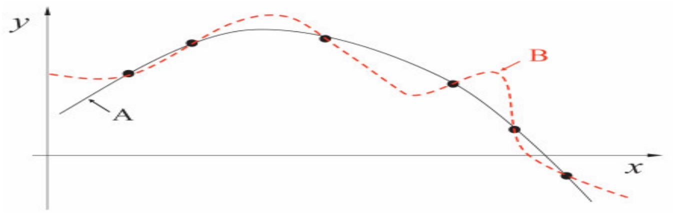
奥卡姆剃刀（Occam's Razor）原理

- 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，即输入特征和标签之间的关系。
  - 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）。

13

例子

- 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？

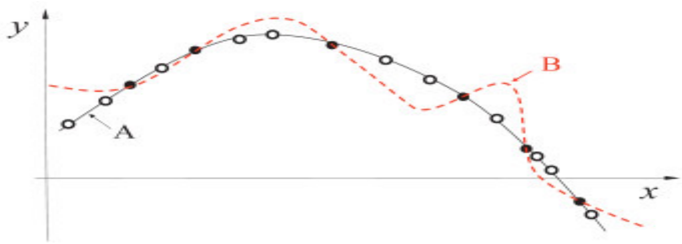


仅仅从这几个数据点来看，我们无法判断哪个更好，或者说，A和B一样好。

14

例子

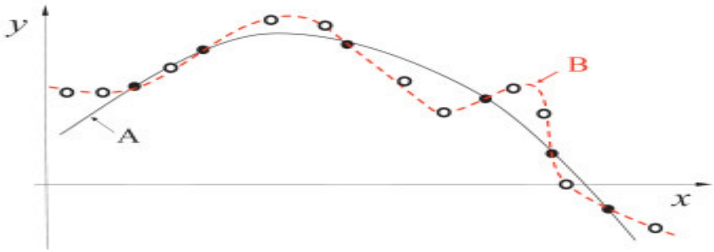
- 测试数据1（空心点）
  - A更好:



15

例子

- 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，是不可能绝对准确地把握的。

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问题都并不复杂，通常使用比较简单的方法就可以取得很好的效果。

16

奥卡姆剃刀原理

- WHY？
- Evolution bias: “strong selection pressure on our pattern recognition apparatuses to be computationally simple”
  - Fewer neurons
  - Less running time
  - Faster response

偏差和方差的折中

- 模型复杂度也可以从误差组成的角度一窥端倪。
- 三种误差来源：

$$Error(\hat{f}) = \underbrace{\left(f(x)-\bar{f}(x)\right)^2}_{\substack{\text{偏差的平方} \\ \text{Bias}}} + \underbrace{\mathbb{E}\left[\left(\hat{f}(x)-\bar{f}(x)\right)^2\right]}_{\substack{\text{方差} \\ \text{Variance}}} + \underbrace{\sigma_\varepsilon^2}_{\substack{\text{随机误差} \\ \text{Noise}}}$$

- 随机误差是不可消除的，与数据产生机制有关（如不同精度设备得到的数据随机误差不同）。
- 偏差和方差与欠拟合/过拟合联系在一起：偏差和方差之间的折中（Bias-Variance Tradeoff）。

补充：偏差-方差分解

- 以平方误差（L2损失）为例，令$\bar{f}(\mathbf{x}) = \mathbb{E}[\hat{f}(\mathbf{x})]$，
- 预测误差可分解为：偏差的平方 + 方差

$$
\begin{aligned}
&\boldsymbol{Error}(\hat{f}) \\
&= \mathbb{E}\left[\left(y - \hat{f}(\mathbf{x})\right)^2\right] = \mathbb{E}\left[\left(f(\mathbf{x}) + \varepsilon - \hat{f}(\mathbf{x})\right)^2\right] = \mathbb{E}\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2\right] + \sigma_\varepsilon^2 \\
&= \mathbb{E}\left[\left(\boxed{f(\mathbf{x}) - f(\mathbf{x})} + \boxed{f(\mathbf{x}) - \hat{f}(\mathbf{x})}\right)^2\right] + \sigma_\varepsilon^2 \qquad \varepsilon\text{与}\hat{f}\text{独立} \\
&= \mathbb{E}\left[\left(\boxed{f(\mathbf{x}) - f(\mathbf{x})}\right)^2\right] + \mathbb{E}\left[\left(\boxed{f(\mathbf{x}) - \hat{f}(\mathbf{x})}\right)^2\right] + 2\mathbb{E}[\boxed{f(\mathbf{x}) - f(\mathbf{x})}]\mathbb{E}[(\boxed{f(\mathbf{x}) - \hat{f}(\mathbf{x})}] + \sigma_\varepsilon^2 \\
&= \left(f(\mathbf{x}) - f(\mathbf{x})\right)^2 + \mathbb{E}\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2\right] + 2\mathbb{E}[f(\mathbf{x}) - f(\mathbf{x})]\left(f(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})]\right) + \sigma_\varepsilon^2 \\
&= \underbrace{\left(f(\mathbf{x}) - f(\mathbf{x})\right)^2}_{\text{偏差的平方}} + \underbrace{\mathbb{E}\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2\right]}_{\text{方差}} + \underbrace{\sigma_\varepsilon^2}_{\text{随机误差}}
\end{aligned}
$$

19

偏差

- 偏差：模型预测值$\hat{f}(\mathbf{x})$的期望与真实规律$f(\mathbf{x})$之间的差异，记为：$\boldsymbol{bias}(\hat{\boldsymbol{f}}) = \mathbb{E}[\hat{\boldsymbol{f}}(\mathbf{x})] - \boldsymbol{f}(\mathbf{x})$
- 期望如何得到？
  - 训练数据无穷多
  - 假设我们可以对整个学习流程重复多次
    - 由于每次收集到的样本稍有不同（每次训练数据集可视为总体数据独立同分布的样本。由于随机性，每次训练样本会有差异），从而每次得到的模型也稍有不同，预测结果也稍有不同。
    - 多次预测结果取平均为期望。

偏差来源于模型中的错误假设。偏差过高就意味着模型所代表的特征和标签之间的关系是错误的，对应欠拟合现象。

20

方差

- 方差：模型预测值的方差，记为：

$$Var(\hat{f}) = \mathbb{E}\left[\left(\hat{f}(\mathbf{x}) - \mathbb{E}[\hat{f}(\mathbf{x})]\right)^2\right]$$

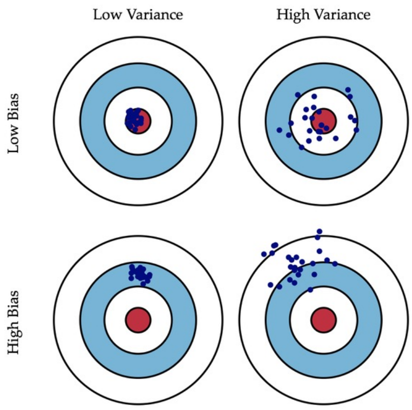
  – 描述的是通过学习拟合出来的结果自身的不稳定性。

- 方差来源于模型对训练数据波动的过度敏感。方差过高意味着模型对数据中的随机噪声也进行了建模，将本不属于“特征 – 标签”关系中的随机特性也纳入到模型之中，对应着过拟合现象。

21

偏差与方差

完美的模型算法　集成学习可降低模型的偏差或/和方差。



推荐阅读：《Understanding the Bias-Variance Tradeoff》
https://liam.page/2017/03/25/bias-variance-tradeoff/

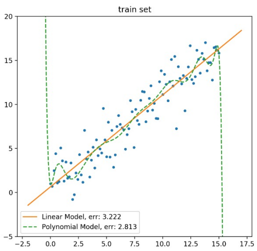
22

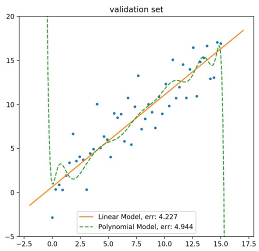
偏差-方差平衡

- 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。





校验集上，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。

23

学习曲线 （Learning Curve）

“match” or “alignment” of the model to the problem
- Bias: accuracy/quality of the match
- Variance: precision/specificity of the match

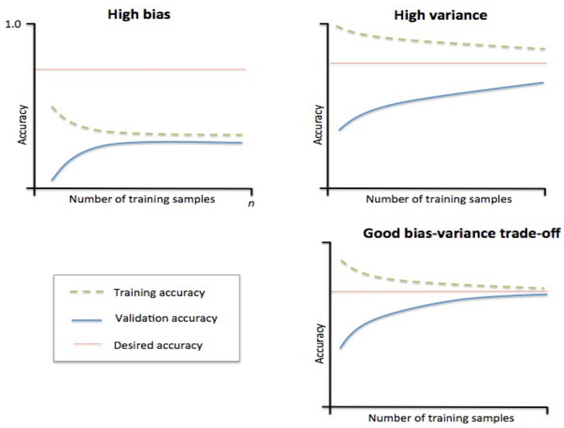
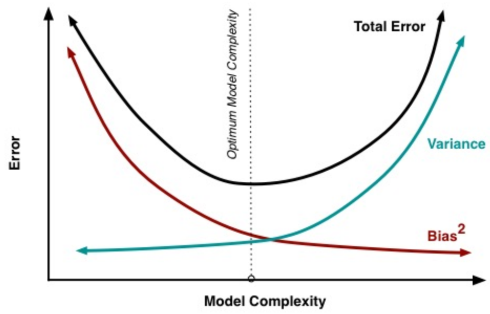


Image source: https://github.com/rasbt/python-machine-learning-book/blob/master/code/ch06/images/06_04.png

24

模型复杂度、偏差、方差

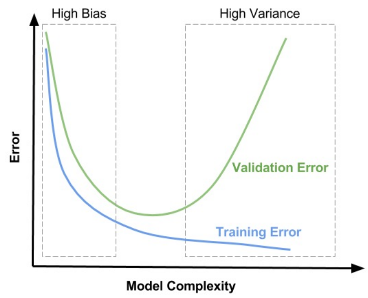
- 选择合适复杂度的模型



25

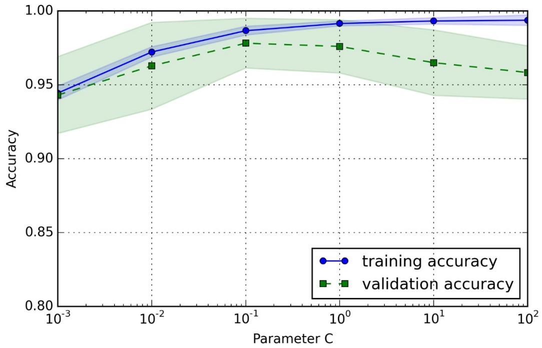
欠拟合和过拟合的外在表现

- 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（模型进入过拟合状态）。



26

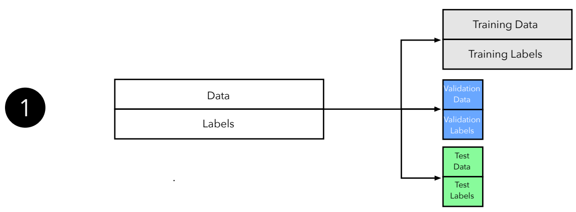
模型复杂度

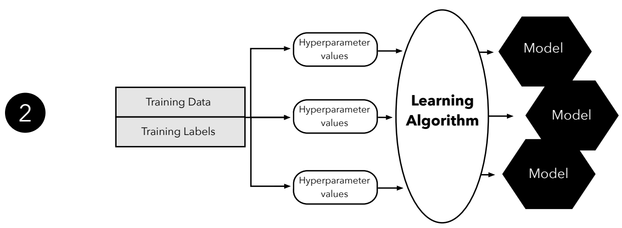


$J(\mathbf{w},b;C)=\frac{1}{2}\|\mathbf{w}\|_2^2+C\sum_{i=1}^{N}L_{Hinge}(y_i,f(\mathbf{x}_i;\mathbf{w},b))$c

27

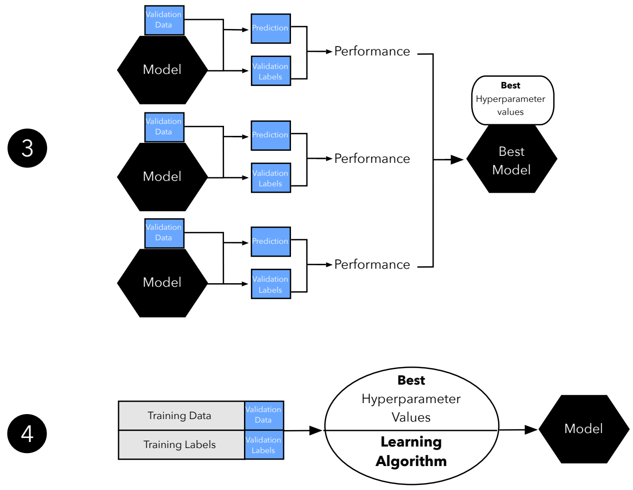
模型校验——校验集





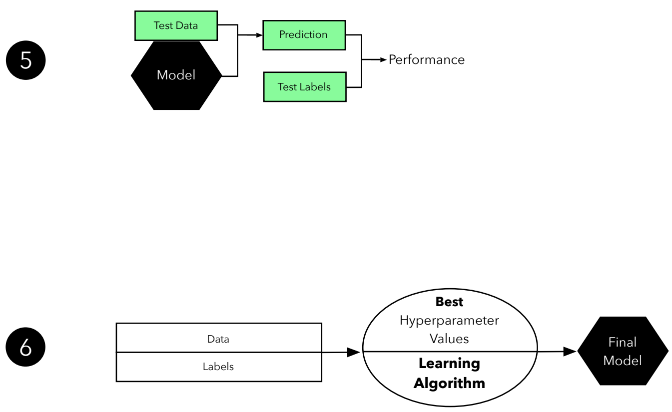
28

模型校验——校验集



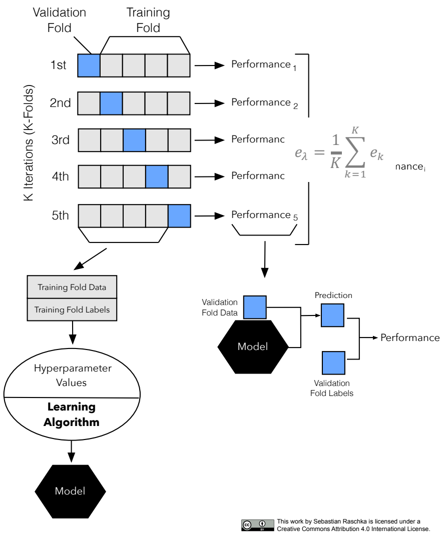
29

模型校验——校验集



30

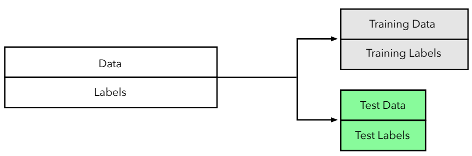
模型校验—K-fold Cross-Validation



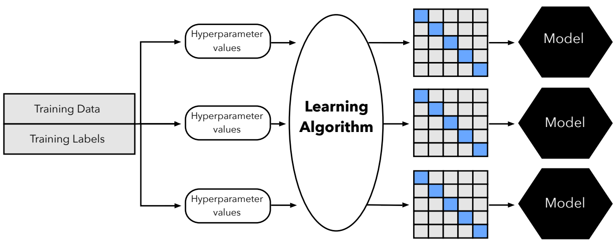
31

K-fold Cross-Validation Pipeline I

1

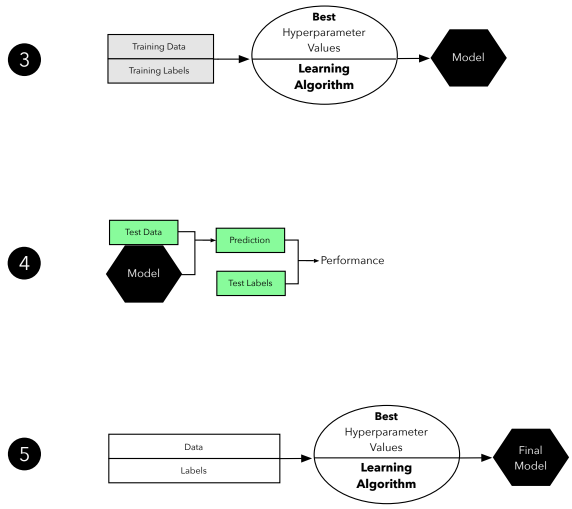


2



32

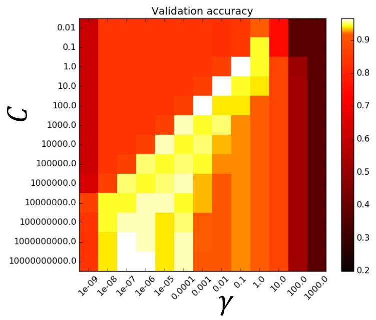
K-fold Cross-Validation Pipeline II



33

网格搜索（Grid Search）

- 穷举：在所有候选的参数组合中循环遍历，尝试每一种可能性，表现最好的参数作为最终结果



C：训练样本有多重要?
越大，越看重训练样本，模型越复杂。

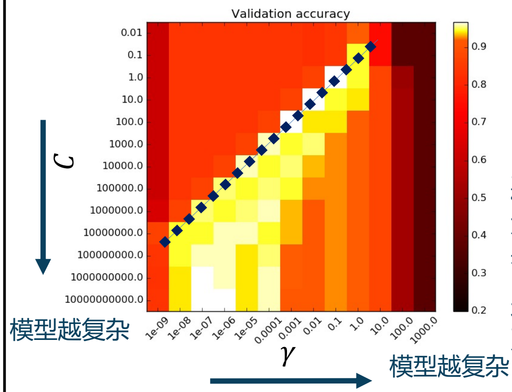
γ： RBF核宽度的倒数.
表示每个训练样本的影响范围。越大，RBF核宽度越小，每个训练样本/支持向量的范围越小，模型越复杂。

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

Source: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

34

Grid Search



对角线上可以找到好的模型：平滑模型（ γ较小）可以通过增加正确分类每个点（较大的C ）的重要性而变得更复杂，从而获得性能良好的模型。

当γ取一些中间值时，C非常大对应的模型性能几乎相同：不必通过强制更大的间隔来正则化，RBF核的半径本身就是一个很好的结构正则化器。

在实践中，较小的C对应的决策函数更简单，内存占用少、预测速度更快

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

Source: http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

35

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。
  - 增加模型的迭代次数
  - 更多特征
  - 降低模型正则化水平
- 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。
  - 及早停止迭代
  - 扩大训练集
  - 减少特征数量
  - 提高模型正则化水平

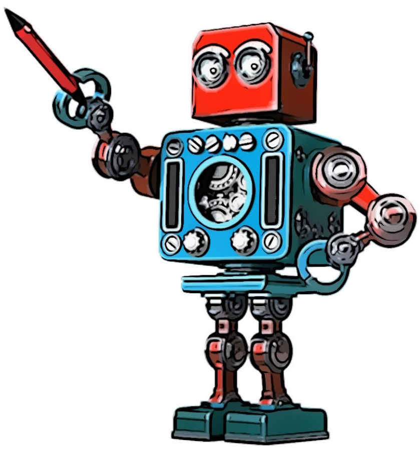
36

小结

- 无免费午餐定理：模型的优劣依赖于先验知识
- 丑小鸭定理：模式相似性依赖于所选择的特征
- 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。

37

- ■ 简介模型性能评价
- ■ **Resampling**
- ■ Bagging
  - 随机森林
- ■ Boosting
- ■ Stacking



2021/12/10 38

38

重采样（Resampling）

- How?
  - 从原始训练集中重采样一个子集
    - Jackknife： 无放回
    - Bootstrap： 有放回
      - 等同于给每个样本点赋予不同权重
- Why?
  - 对整体统计量生成更多新的估计
    - 通常能改进分类效果

39

Bootstrap

- 通过从原始的N个样本数据$\mathcal{D}=\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$进行$N$次有放回采样$N$个数据$\mathcal{D}'$，称为一个bootstrap样本。
  - 对原始数据进行有放回的随机采样，抽取的样本数目同原始样本数目一样。
  - 等价于给样本reweighting
- 如：若原始样本为$\mathcal{D}=\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_5\}$
- 则bootstrap样本可能为
  - $\mathcal{D}^1=\{\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_5,\mathbf{x}_4,\mathbf{x}_5\}$
  - $\mathcal{D}^2=\{\mathbf{x}_1,\mathbf{x}_3,\mathbf{x}_1,\mathbf{x}_4,\mathbf{x}_5\}$

40

Bootstrap

- 例如，假设一批产品随机抽出30个，使用寿命（天数）如下，用bootstrap的方法估计这批产品寿命95%的置信区间。

Data=(119, 120, 131, 209, 210, 337, 332, 287, 146, 129,
232, 169, 208, 253, 142, 105, 419, 179, 324, 287,
115, 132, 308, 356, 286, 221, 204, 105, 45, 245)

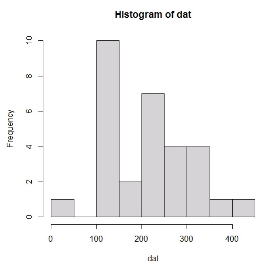


图1 初始数据的频数直方图

循环1000次，有放回的抽样，每次生成的30个新的伪样本，求mean；
结果为1000个bootstrap样本的mean值，求95%的置信区间（2.5%-97.5%）；
在初始样本足够大的情况下，bootstrap抽样能够无偏接近总体分布。

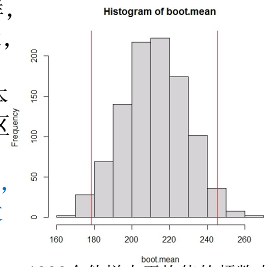
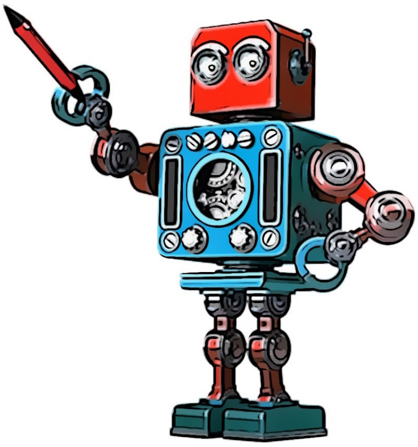


图2 Bootstrap 1000个伪样本平均值的频数直方图

Arcing

- Arcing: Adaptive Reweighting and CombinING
  - 通过重用或者选择性使用数据来改进分类器
- Bagging: Bootstrap AGGregatING
  - Independently bootstrap data sets
- Boosting
  - Dependently bootstrap data sets
- AdaBoost

- 简介模型性能评价
- Resampling
- Bagging
  - 随机森林
- Boosting
- Stacking



Bagging

- Breiman, 1996
- 对给定有N个样本的数据集$\mathcal{D}$进行Bootstrap采样，得到$\mathcal{D}^1$，在$\mathcal{D}^1$上训练模型f_1
- 上述过程重复M次，得到M个模型，则M个模型的平均（回归）/投票（分类）为：

$$f_{avg}(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M} f_m(\mathbf{x})$$

aggregating

- 可以证明：Bagging可以降低模型的方差。

Bagging可降低模型方差

- 令随机变量X的均值为μ，方差为σ^2，
- 则N个独立同分布的样本的均值$\bar{X}$为：$\bar{X} = \frac{1}{N}\sum_{i=1}^{N} X_i$
- 样本均值$\bar{X}$的期望为： $\mathbb{E}(\bar{X}) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}(X_i) = \mu$
  - 样本均值$\bar{X}$的期望和X的期望相等（无偏估计）
- 样本均值$\bar{X}$的方差为： $Var(\bar{X}) = \frac{1}{N^2}\sum_{i=1}^{N} Var(X_i) = \frac{\sigma^2}{N}$
  - 样本均值$\bar{X}$的方差$Var(\bar{X})$ 比X的方差σ^2小
  - 样本数N越大，方差$Var(\bar{X})$越小

45

Bagging可降低模型方差

- 在Bagging中，M次预测结果的均值$f_{avg}(\mathbf{x})$的方差比用原始训练样本单次训练的模型的预测结果的方差小，均值不变
  - Bagging可以降低模型方差
  - Bagging不改变模型偏差
- 注意：Bagging中每个模型不完全独立（训练样本有一部分相同），方差的减少没那么多，但也会减少
- 若f_m之间的相关性为ρ，则f_{avg}的方差为：$\rho\times\sigma^2 + (1-\rho)\times\frac{\sigma^2}{M}$

46

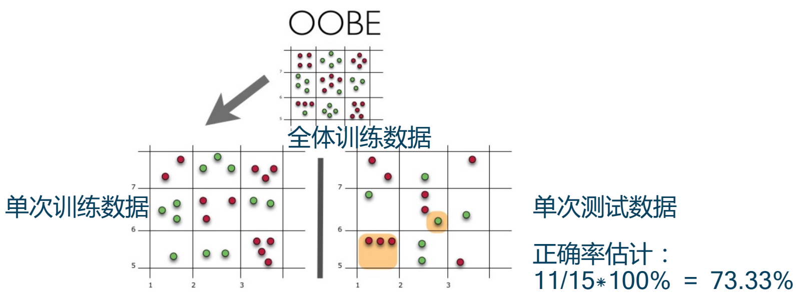
Bagging

- Bagging适合对偏差低、方差高的模型进行融合
  - 如决策树、神经网络
- 决策树很容易过拟合 → 偏差低、方差高
  - 如果每个训练样本为一个叶子结点，训练误差为0

47

补充：Out-of-bag Error（OOBE）

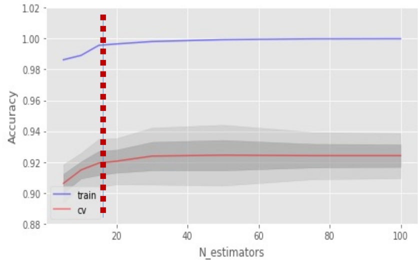
- 在Bagging中，每个基学习器只在原始数据集的一部分上训练，所以可以不用交叉验证，直接用包外样本上的误差（out-of-bag error）来估计它的泛化误差/测试误差。



50

补充：基学习器数目

- 在Bagging中，通常基学习器的数目越多，效果越好，但测试时间与训练时间也会随之增加。
- 当树的数量超过一个临界值之后，算法的效果并不会很显著地变好。所以参数基学习器数目n_estimators不是模型复杂度参数，无需通过交叉验证来确定。
- 参数值建议：
  - 对分类问题，可设置基学习器数目为$\sqrt{D}$，其中D 为特征数目；
  - 对回归问题，可设置基学习器数目为 $D/3$ 。



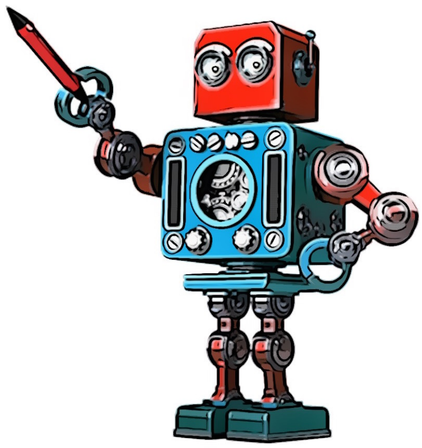
51

补充：随机森林（Random Forest）

- 由于仅训练数据有些不同，对决策树算法进行Bagging得到的多棵树高度相关，因此带来的方差减少有限
- 随机森林通过
  - 随机选择一部分特征
  - 随机选择一部分样本
- 降低树的相关性
- 随机森林在很多应用案例被证明有效，但牺牲了可解释性
  - 森林：多棵树
  - 随机：对样本和特征进行随机抽取

52

- Introduction
- 模型性能评价
- Bagging
- Boosting
  - 基本思想
  - AdaBoost
- Stacking



Boosting基本思想

- Boosting：将弱学习器组合成强分类器
  - 构造一个性能很高的预测（强学习器）是一件很困难的事情
  - 但构造一个性能一般的预测（弱学习器）并不难
    - 弱学习器：性能比随机猜测略好（如层数不深的决策树）
- Boosting学习框架
  - 学习第一个弱学习器ϕ_1
  - 学习第二个弱学习器ϕ_2，ϕ_2要能帮助ϕ_1（ϕ_2和ϕ_1互补）
  - …
  - 组合所有的弱学习器：$f(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m \phi_m(\mathbf{x})$

  弱学习器是按顺序学习的。

Boosting例子

- $D_1 = randomly\ select\ a\ subset\ of\ X$
- $D_2 = select\ from\ X/D_1,$

比例可以不同 $\{half\ correctly\ classified\ by\ h_1\} + \{half\ incorrectly\ classified\ by\ h_1\}$

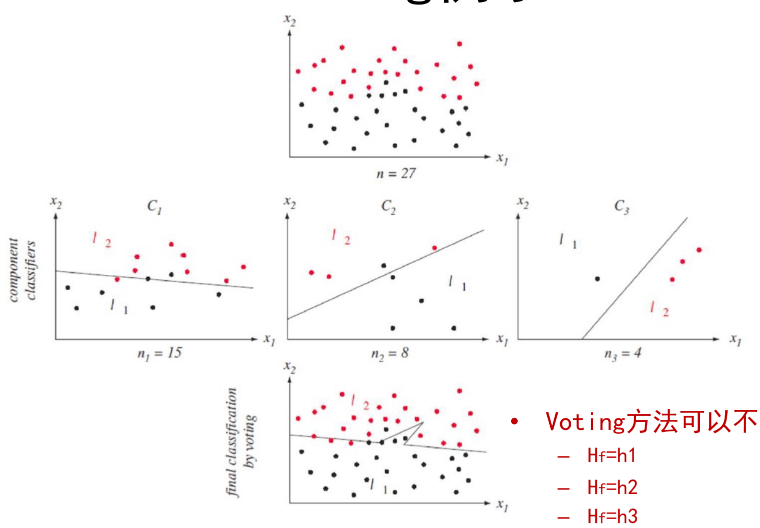
- $D_3 = \{x_i \in (X/D_1 \cup D_2)\ and\ h_1(x_i) \neq h_2(x_i)\}$
- The final classifier: Voting方法可以不同

$$h_{\text{final}}(\mathrm{x}) = \begin{cases} h_1(x); & if\ h_1(x) == h_2(x) \\ h_3(x); & otherwise \end{cases}$$

2021/12/10 55

55

Boosting例子



- Voting方法可以不同
  - Hf=h1
  - Hf=h2
  - Hf=h3

2021/12/10 56

56

各种Boosting

AdaBoost.M1, AdaBoost.MR,
FilterBoost, GentleBoost,
GradientBoost, MadaBoost,
LogitBoost, LPBoost,
MultiBoost, RealBoost,
RobustBoost, …

怎样得到互补的学习器?

- 在不同的训练集上训练学习器。
- 怎么得到不同的训练集?
  - 对原始训练集重采样
  - 对原始训练集重新加权
    - 在实际操作中可改变目标函数：

$$J(f,\lambda)=\sum_{i=1}^{N}L(y_i,f(\mathbf{x}_i))+\lambda R(f)$$

$$(\mathbf{x}_1,y_1,w_1)$$
$$\cdots$$
$$(\mathbf{x}_N,y_N,w_N)$$

$$J(f,\lambda)=\sum_{i=1}^{N}w_iL(y_i,f(\mathbf{x}_i))+\lambda R(f)$$

Adaboost基本思想

- Open problem [Kearns & Valiant, STOC' 89]:

"weakly learnable ?= strongly learnable"

- "YES！" [Schapire，1990]

"Weights of misclassified samples are increase in (t+1)th iteration."

59

59

AdaBoost的基本思想

- Freund & Schapire, 1995
- 在弱学习器ϕ_1失败的样本上学习第二个弱学习器ϕ_2

$$\varepsilon_1 = \sum_{i=1}^{N} w_{1,i} \mathbb{I}(y_i \neq \phi_1(\mathbf{x}_i)),$$

$\mathbb{I}(condition)$: 指示（Indicator）函数，满足条件值为1，否则为0

- 令弱学习器ϕ_1在其训练集上的误差为：

$$\sum_{i=1}^{N} w_{2,i} \mathbb{I}(y_i \neq \phi_1(\mathbf{x}_i)) = \frac{1}{2}$$

学习器ϕ_1在训练集2上的性能为随机猜测

- 样本重加权
  - 分对的样本，其权重减小
  - 分错的样本，其权重增大

60

Adaboost基本思想

- given training set $(x_1, y_1), \ldots, (x_m, y_m)$
  where $x_i \in X$, $y_i \in \{-1, +1\}$
- initialize $D_1(i) = 1/m \ (\forall i)$
- for $t = 1, \ldots, T$:
  - train weak classifier $h_t : X \to \{-1, +1\}$ with error
    $\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$ < 1/2　　//如果error=1/2，即学习器h1在训练集D2上的性能为随机猜测
  - $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ >0　　//如果error越小，则权重越大
  - update $\forall i$:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t y_i h_t = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t}, if\ yi = ht(xi) \\ e^{\alpha_t},\ \ if\ yi \neq h_t(xi) \end{cases}$$

  where Z_t = normalization factor

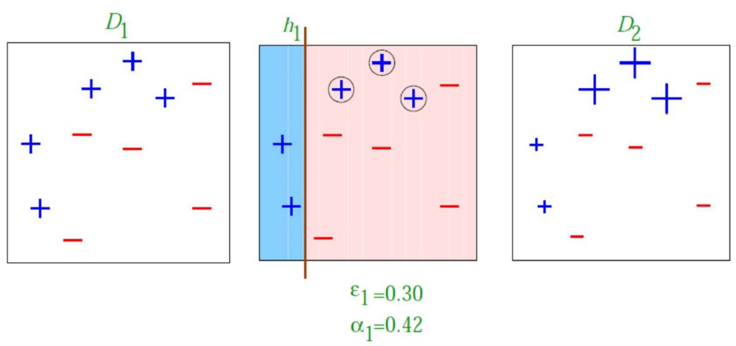
- $H_{\text{final}}(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$　　//Stronger classifier的权重较大

2021/12/10　　61

61

Adaboost例子

- Initial　　　Round1

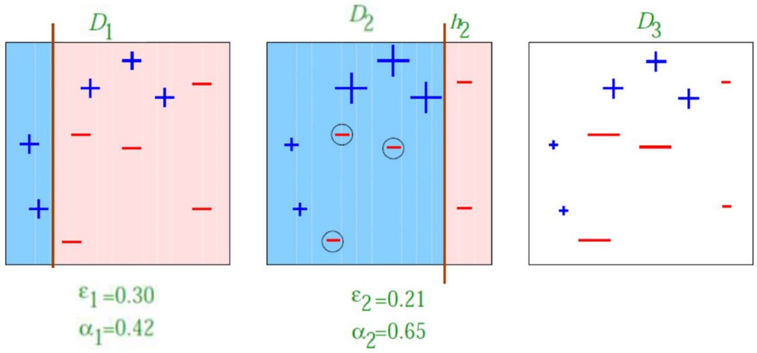


2021/12/10　　62

62

Adaboost例子

- Round 2

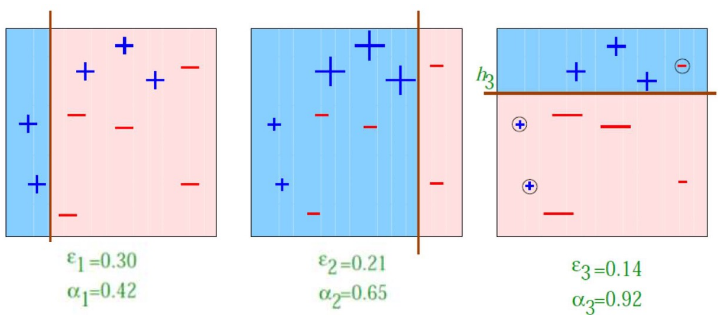


2021/12/10 63

63

Adaboost例子

- Round 3

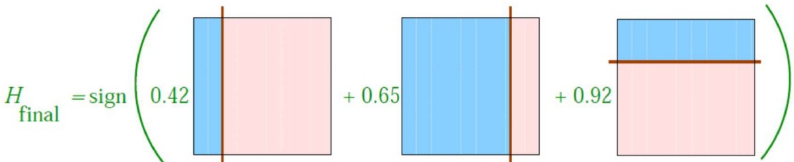


2021/12/10 64

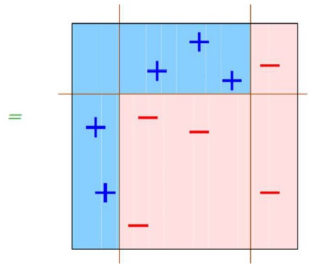
64

Adaboost例子

- 最终得到的结果



弱学习器：深度为1的决策树（树桩，stumps）



强学习器

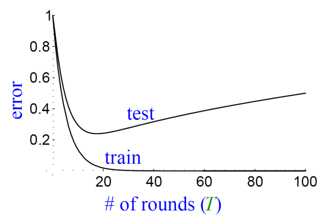
可以证明：随着弱学习器数目增多，训练误差越来越小。

65

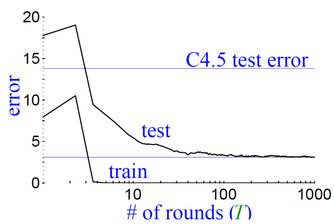
测试误差

- 训练误差随着M的增加而减小。测试误差呢？

理论上可能会Overfitting？



实际上测试误差并没有增加？



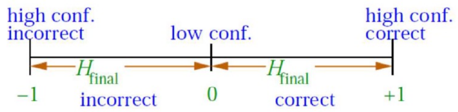
(boosting C4.5 on "letter" dataset)

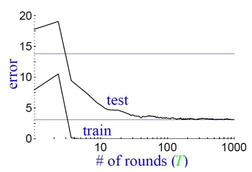
	# rounds		
	5	100	1000
train error	0.0	0.0	0.0
test error	8.4	3.3	3.1

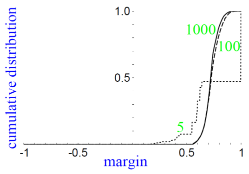
66

测试误差

- 上述训练误差（0-1损失）只考虑了分类是否正确，还应该考虑分类的置信度
- Margin的累积分布







	# rounds		
	5	100	1000
train error	0.0	0.0	0.0
test error	8.4	3.3	3.1
% margins $\leq$ 0.5	7.7	0.0	0.0
minimum margin	0.14	0.52	0.55

67

Adaboost优点

- 实现快速、简单
  - 参数少
- 灵活
  - weak classifier可采用很多简单算法
  - 构建weak classifier的先验知识（假设）限制少
- 通用性高
  - 不同模态数据
  - 多类别数据

2021/12/10 68

68

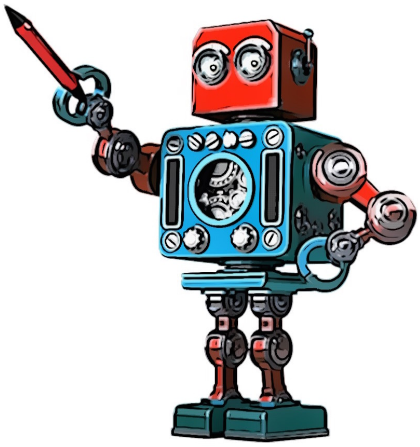
Adaboost缺点

- AdaBoost的性能取决于数据和弱学习器
  - 与理论一致，如果弱分类器过于复杂，AdaBoost可能会失败（overfitting）
  - 弱分类器太弱（underfitting/overfitting）
  - 根据经验，AdaBoost容易受到均匀噪声的影响

Adaboost用于人脸识别中的例子

- 中科院计算所，山世光研究员
- PPT另附

- Introduction
- 模型性能评价
- Bagging
- Boosting
- **Stacking**

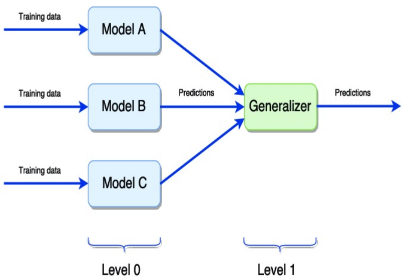


71

71

Stacking

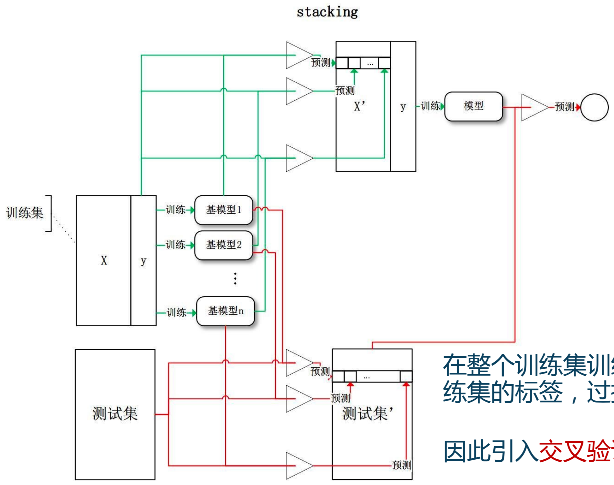
- Stacking是一种分层的结构
- 二层Stacking：
  - 将训练好的基模型对训练集进行预测
    - 新的训练集：第j个基模型对第i个训练样本的预测值将作为新的训练集中第i个样本的第j个特征值
    - 新的测试集：所有基模型的对测试集的预测
  - 在新的训练集上训练模型，在新的测试集上进行预测



Stacked Gereralization, David H. Wolpert, *Neural Networks*, Volume 5, Issue 2, Pages 241-259.

72

Stacking



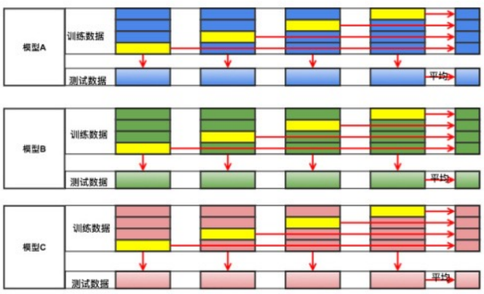
在整个训练集训练的模型反过来去预测训练集的标签，过拟合会非常非常严重 X

因此引入交叉验证

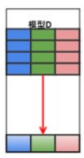
73

带交叉验证的Stacking

- 以3个基模型、4折交叉验证为例：



基模型



融合后的模型

74

小结

- No Free Lunch Theorem

没有最好的学习器

- Ugly Duckling Theorem

没有最优的特征

- Occam’ s razor

模型/描述/…越简单越好

- Resampling = Reweighting

75

75

- END

2021/12/11 76

76