

# 第十一章 集成学习 ——随机森林


卿来云

## 集成学习

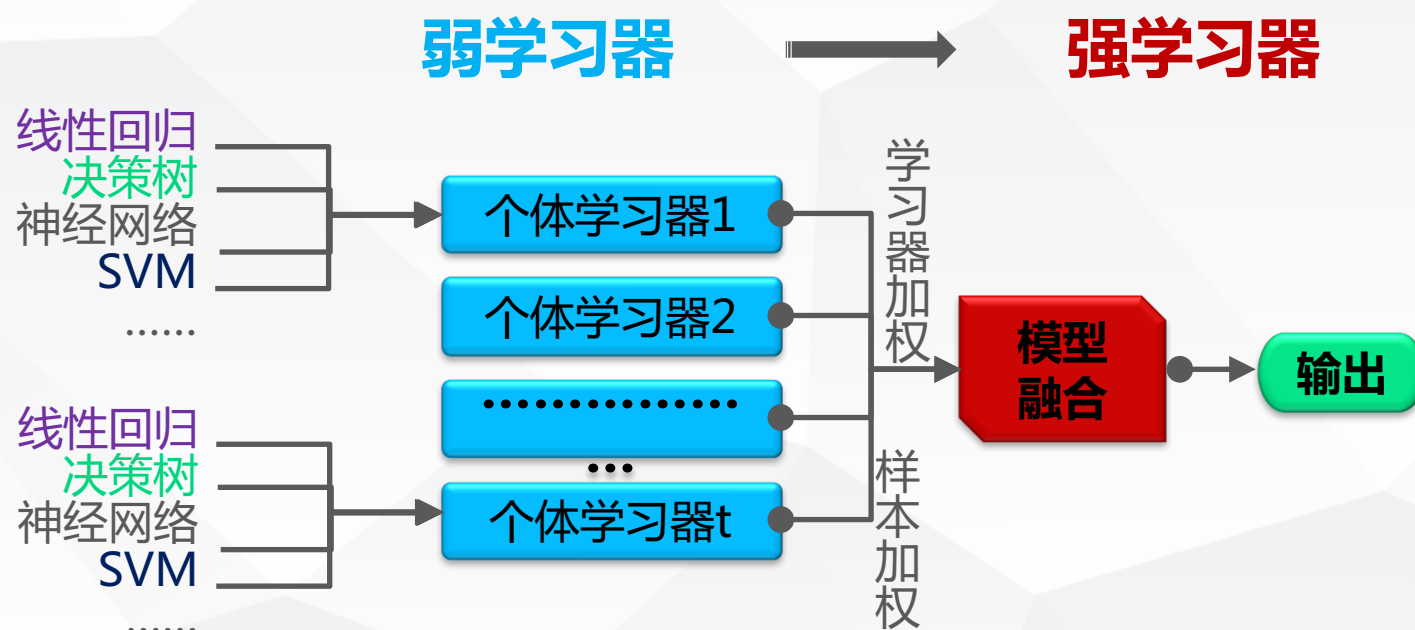
- 我们已经开发了很多机器学习算法/代码
- 单个模型的性能已经调到最优，很难再有改进
- 集成学习：用很少量的工作，组合多个基模型，使得系统总的性能提高
  - 基模型最好变化多样，这样不同的基模型集成后形成互补。



•三个臭皮匠，顶个诸葛亮



## 集成学习



将多个弱学习器进行融合，通过对样本加权、学习器加权，获得比单一学习器显著优越的泛化性能的强学习器

# Outline

- 模型性能评价
  - No Free Lunch Theorems
  - Occam剃刀原理
  - 偏差-方差折中
- Bagging
  - 随机森林
- Boosting
  - AdaBoost
  - Gradient Boosting Decision Tree (GBDT)
  - XGBoost
  - LightGBM
- Stacking

## ➤ Bagging

- 对给定有  $N$  个样本的数据集  $\mathcal{D}$  进行 **B**ootstrap 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：
- 可以证明：Bagging可以降低模型的方差。

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

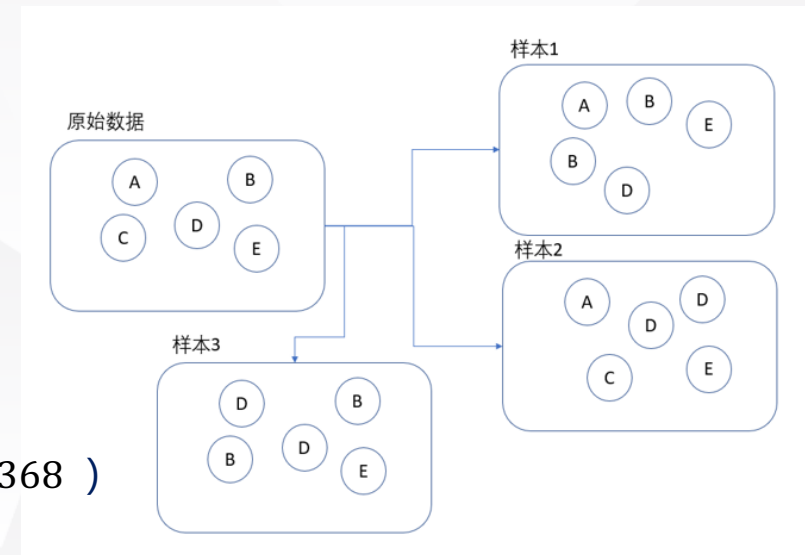
**aggregating**

# Bootstrap

- 通过从原始的 $N$ 个样本数据 $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  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。
- 如：若原始样本为 $\mathcal{D} = \{A, B, C, D, E\}$
- 则bootstrap样本可能为
  - $\mathcal{D}^1 = \{A, B, B, D, E\}$
  - $\mathcal{D}^2 = \{A, C, D, D, E\}$

一个样本不在采样集中出现的概率： $\left(1 - \frac{1}{N}\right)^N$ 。 ( $\lim_{N \rightarrow \infty} \left(1 - \frac{1}{N}\right)^N = 0.368$ )

原始训练集中约有： $1 - 0.368 = 63.2\%$ 的样本出现在采样集中。



## ➤ Bagging可降低模型方差

- 令随机变量 $X$ 的均值为 $\mu$ ，方差为 $\sigma^2$ ，
- 则 $N$ 个独立同分布的样本的样本均值 $\bar{X}$ 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： $Var(\bar{X}) = \frac{1}{N^2} \sum_{i=1}^N Var(X_i) = \frac{\sigma^2}{N}$ 
  - $Var$ 表示方差运算
  - 样本均值 $\bar{X}$ 的方差比 $X$ 的方差小（ $N$ 越大，样本数越多，方差越小）

推荐阅读：

1. 《为什么说bagging是减少variance，而boosting是减少bias?》

2. 使用sklearn进行集成学习——理论 <https://www.cnblogs.com/jasonfreak/p/5657196.html>



## ➤ 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 $\rho$ ，则 $f_{avg}$ 的方差为： $\rho \times \sigma^2 + (1 - \rho) \times \frac{\sigma^2}{M}$

推荐阅读：

1. [《为什么说bagging是减少variance，而boosting是减少bias?》](#)

2. [使用sklearn进行集成学习——理论](https://www.cnblogs.com/jasonfreak/p/5657196.html) <https://www.cnblogs.com/jasonfreak/p/5657196.html>

## 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

## ➤ Scikit-Learn中的Bagging

- Scikit-Learn中支持对任意基学习器的Bagging
  - 分类 : BaggingClassifier
  - 回归 : BaggingRegressor

```
class sklearn.ensemble.BaggingClassifier(base_estimator=None, n_estimators=10, max_samples=1.0,  
max_features=1.0, bootstrap=True, bootstrap_features=False, oob_score=False, warm_start=False, n  
_jobs=None, random_state=None, verbose=0)
```

```
class sklearn.ensemble.BaggingRegressor(base_estimator=None, n_estimators=10, max_samples=1.0,  
max_features=1.0, bootstrap=True, bootstrap_features=False, oob_score=False, warm_start=False, n_j  
obs=None, random_state=None, verbose=0)
```

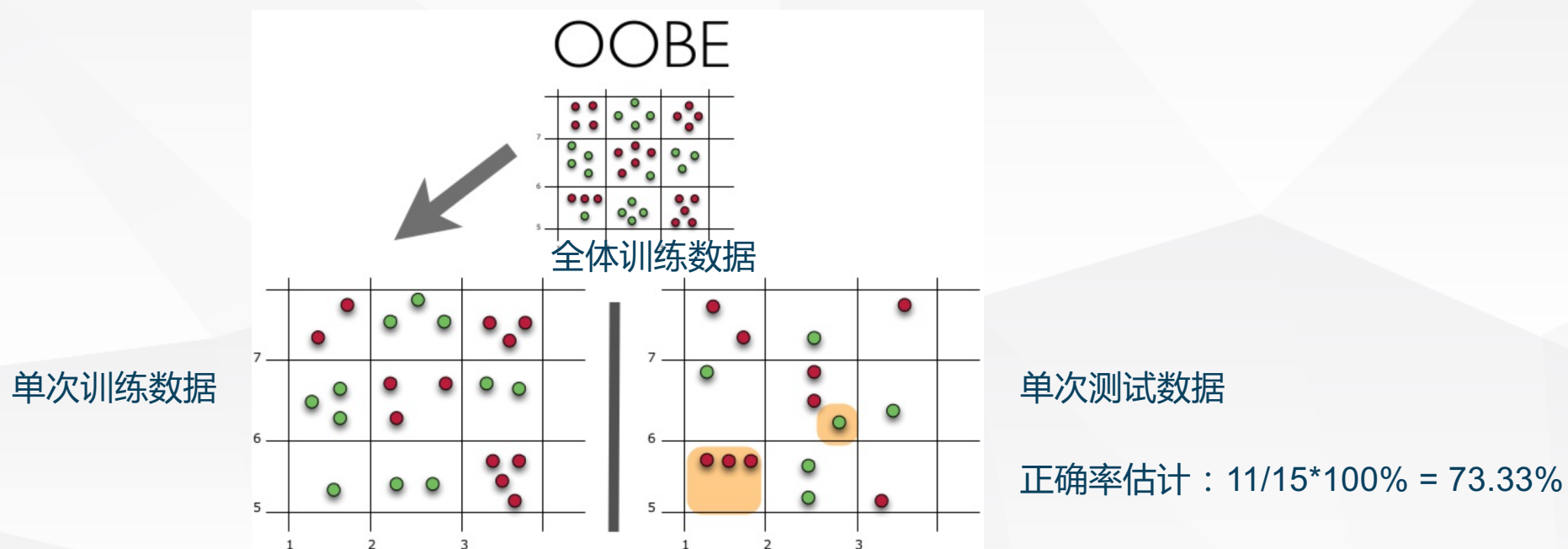
## ➤ BaggingClassifier的参数

```
class sklearn.ensemble.BaggingClassifier(base_estimator=None, n_estimators=10, max_samples=1.0,
max_features=1.0, bootstrap=True, bootstrap_features=False, oob_score=False, warm_start=False, n
_jobs=None, random_state=None, verbose=0)
```

参数	说明
<b>base_estimator</b>	基学习器，scikit-learn的分类器或回归器。如果没有给出，默认使用决策树（不推荐，不如RandomForest）。
<b>n_estimators</b>	基学习器的数目。通常基学习器越多，模型的方差越小。
<b>max_samples</b>	每个数据子集（用于训练基学习器）的样本数量。可以是浮点数（0.0至1.0，表示取样本占所有样本的比例），也可以是整数（表示样本的实际数量）。注意：如果输入了1而不是1.0，那么每个数据子集仅包含1个样本，会导致严重失误。
<b>max_features</b>	训练基学习器的特征数量。
<b>bootstrap</b>	在随机选取样本时是否是有放回
<b>bootstrap_features</b>	在随机选取特征时是否是有放回
<b>oob_score</b>	是否计算out-of-bag分数。每个基学习器只在原始数据集的一部分上训练，所以可以用剩下样本上的误差（out-of-bag error），来估计它的泛化误差/测试误差。
<b>warm_start</b>	如果是True，在下次使用fit方法时，向原有的模型再增加n_estimators个新的基学习器。

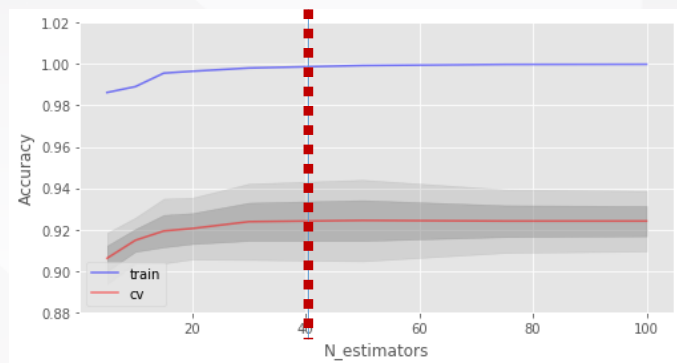
## Out-of-bag error (OOBE)

- 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。



## 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 $D/3$ 。

## ➤ 随机森林 ( Random Forest )

- 由于只是训练数据有一些不同，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上被证明有效，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

## ➤ Scikit-Learn中的随机森林

- Scikit-Learn中实现了两种包含随机树的森林
  - 随机森林 ( Random Forests )
  - 极度随机森林 ( Extremely Randomized Trees )
- 极度随机森林组合比随机森林更随机
  - 在分裂时，随机森林寻找特征最有判别力的阈值。
  - 极度随机森林中，随机选取每个候选特征的阈值，然后从这些随机选取的阈值中寻找最佳阈值。
  - 极度随机森林对方差的减少会更多一些，但偏差可能增大一点点。



## ➤ 随机森林超参数调优

- 随机森林模型参数众多，且涉及随机操作，有时在分类任务中很多时候不同类别的样本数目不均衡，在超参数调优时需慎重。
- 一般来说，先调含随机性的参数
  - 先初调“子采样率”（`subsample`）和“分裂时考虑的最大特征数”（`max_features`）
  - 再调叶节点最小样本数”（`min_samples_leaf`）和“分裂所需最小样本数”（`min_samples_split`）
- 再调无随机性的参数：
  - “最大深度”（`max_depth`）或“最大叶节点数”（`max_leaf_nodes`）

[推荐阅读：使用sklearn进行集成学习——实践](#)

## ➤ 案例：Otto商品分类

- 决策树
- 随机森林

The End