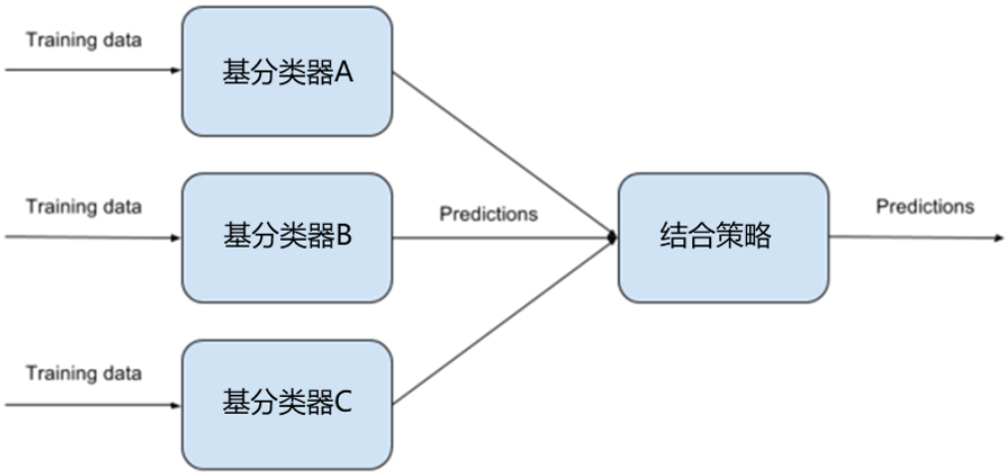
集成模型概述(一)

单个学习器要么容易欠拟合要么容易过拟合，为了获得泛化性能优良的学习器，可以训练多个个体学习器，通过一定的结合策略，最终形成一个强学习器。这种集成多个个体学习器的方法称为集成学习(ensemble learning)。 基本思想如下图所示：



从以上概念可以看出，集成学习主要围绕两个核心问题：

- **如何选取个体学习器？**
- **如何选择结合策略？**

1. 集成学习之个体学习器

个体学习器（又称为“基学习器”）的选择有两种方式：

- 集成中只包含同种类型的个体学习器，称为同质集成 。
- 集成中包含不同类型的个体学习器，为异质集成 。

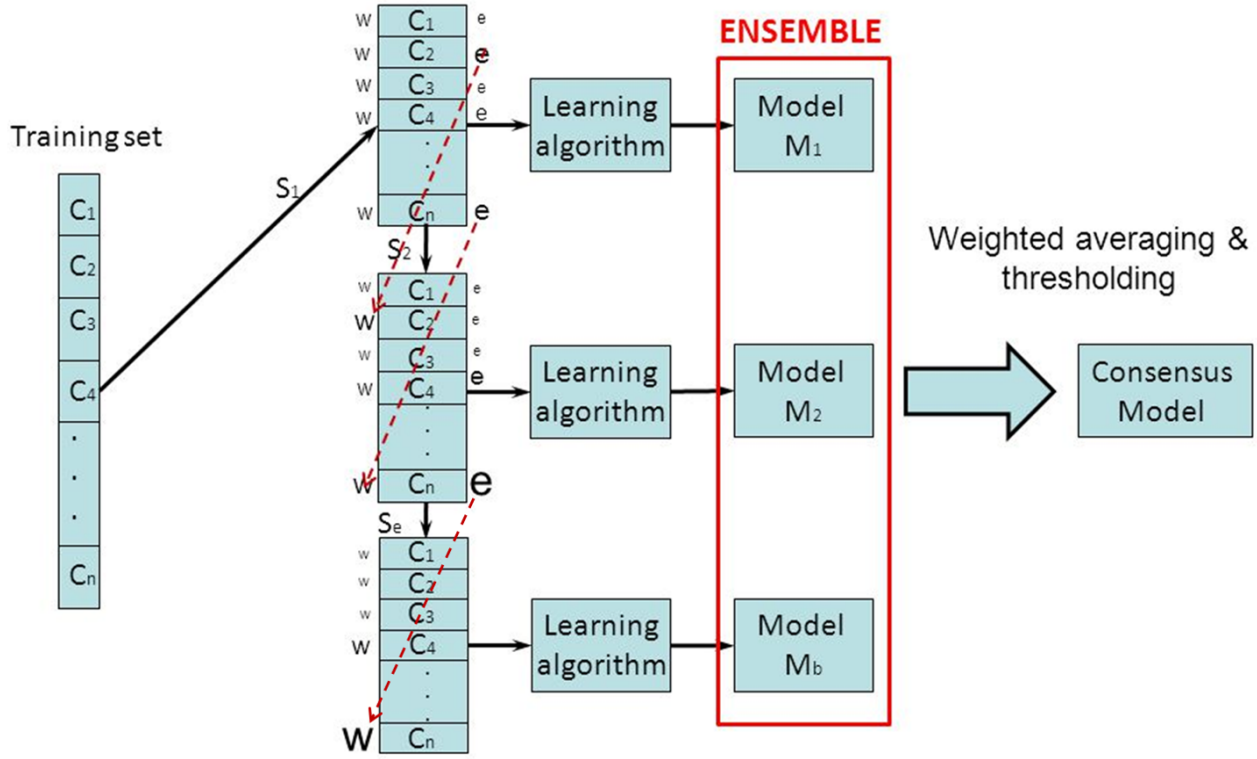
目前同质集成 的应用最广泛，而基学习器使用最多的模型是CART决策树和神经网络。

按照个体学习器之间是否存在依赖关系可以分为两类：

- 个体学习器之间存在强依赖关系，一系列个体学习器基本必须串行生成，代表是boosting系列算法。
- 个体学习器之间不存在强依赖关系 ，一系列个体学习器可以并行生成，代表是bagging系列算法。

1.1 boosting算法原理

boosting的算法原理如下所示：



Boosting算法的工作机制是：

（1）先从初始训练集训练出一个基学习器；

（2）再根据基学习器的表现对样本权重进行调整，增加基学习器误分类样本的权重（又称重采样）；

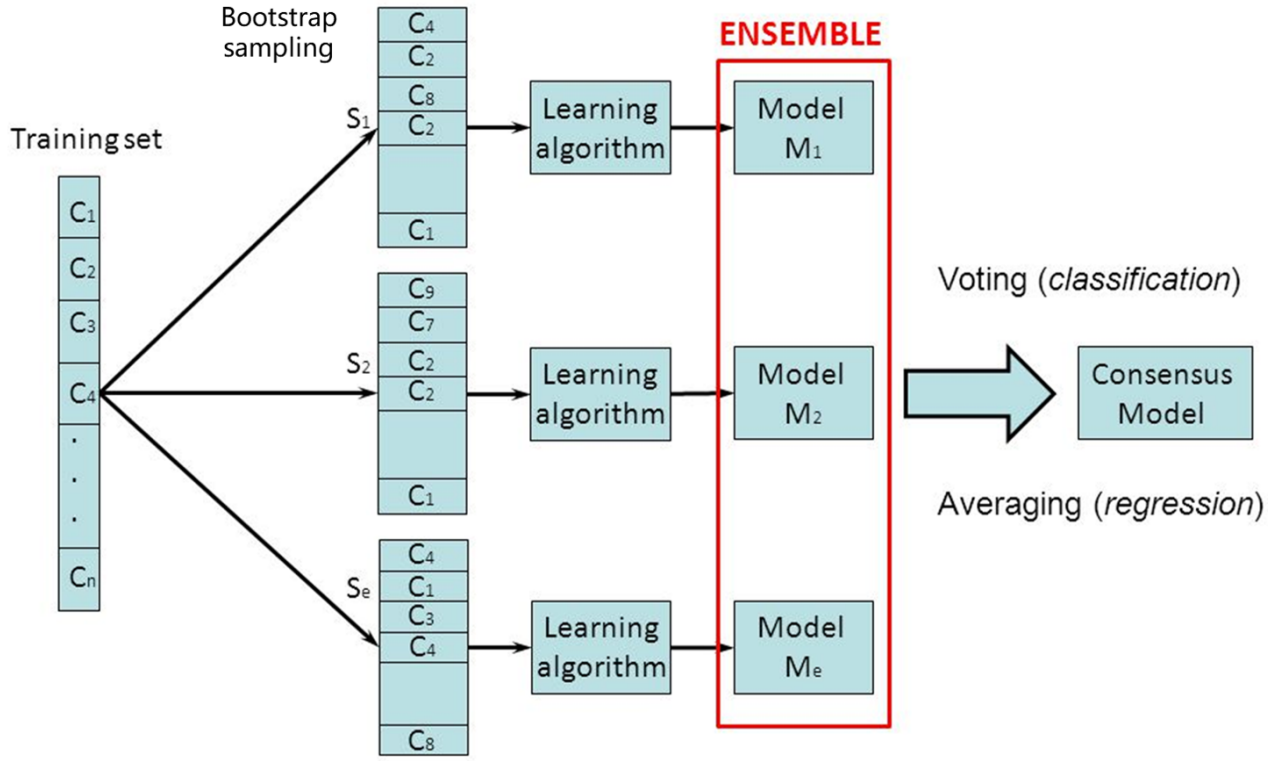
（3）基于调整后的样本分布来训练下一个基学习器；

（4）如此重复进行，直至基学习器数目达到事先指定的个数 T，将这 T 个基学习器通过集合策略进行整合，得到最终的强学习器。

Boosting系列算法里最著名算法主要有AdaBoost算法和提升树(boosting tree)系列算法。提升树系列算法里面应用最广泛的是梯度提升树(Gradient Boosting Tree)。后面我们会讲到这些优秀的算法。

1.2. 集成学习之Bagging 算法原理

Bagging的算法原理如下：



bagging算法的工作机制为：

（1）对训练集利用自助采样法进行T次随机采样，每次采样得到 m 个样本的采样集；

（2）对于这 T 个采样集，我们可以分别独立的训练出 T 个基学习器；

（3）再对这 T 个基学习器通过集合策略来得到最终的强学习器。

值得注意的是这里的随机采样采用的是自助采样法（Bootstap sampling），自助采样法是一种有放回的采样。即对于 m 个样本的原始训练集，我们每次先随机采集一个样本放入采样集，接着把该样本放回，这样采集 m 次，最终可以得到 m 个样本的采样集，由于是随机采样，这样每次的采样集是和原始训练集不同的，和其他采样集也是不同的。

对于一个样本，它每次被采集到的概率是 $\frac{1}{m}$。不被采集到的概率为 $1-\frac{1}{m}$。如果 m 次采样都没有被采集中的概率是 $(1-\frac{1}{m})^m$。则 $\lim\limits_{m\to+\infin}(1-\frac{1}{m})^m \to \frac{1}{e} \approx 0.368$，即当抽样的样本量足够大时，在bagging的每轮随机采样中，训练集中大约有36.8%的数据没有被采集中。对于这部分大约36.8%的没有被采样到的数据，我们常常称之为袋外数据（Out Of Bag, 简称OOB）。这些数据未参与训练集模型的拟合，可以用来检测模型的泛化能力。

bagging对于弱学习器最常用的一般也是决策树和神经网络。bagging的集合策略也比较简单，对于分类问题，通常使用相对多数投票法。对于回归问题，通常使用算术平均法。

2. 集成学习之结合策略

上面几节主要关注于学习器，下面就对集成学习之结合策略做一个总结。我们假定我得到的T个弱学习器是 $\{h_1,h_2,...h_T\}$

2.1 平均法

对于回归问题，通常使用的结合策略是平均法。

最简单的平均是算术平均，即：

$$H(x) = \frac{1}{T}\sum_{1}^{T} h_i(x)$$

也可以是每个个体学习器的加权平均，即 ：

$$H(x) = \sum_{i=1}^{T} w_i h_i(x)$$

其中 w_i 是个体学习器 h_i 的权重，$0 \geq w_i \geq 0,\quad \sum_{i=1}^{T} w_i = 1$。

2.2 投票法

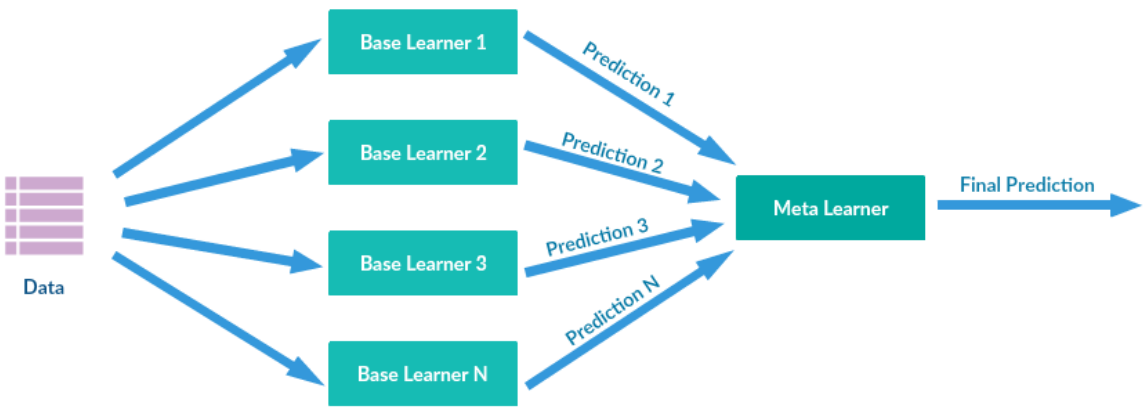
对于分类问题通常使用投票法。假设我们的预测类别是$\{c_1, c_2, \ldots c_K\}$，对于任意一个预测样本 x，我们的 T 个弱学习器的预测结果分别是 $(h_1(x), h_2(x) \ldots h_T(x))$。主要有以下三种：

- 相对多数投票法：也就是少数服从多数，即预测结果中票数最高的分类类别。如果不止一个类别获得最高票，则随机选择一个作为最终类别。
- 绝对多数投票法：即不光要求获得最高票，还要求票过半数。
- 加权投票法：每个弱学习器的分类票数要乘以一个权重，最终将各个类别的加权票数求和，最大的值对应的类别为最终类别。

2.3 Stacking

平均法和投票法仅是对弱学习器的结果做简单的逻辑处理，而stacking是再加上一层权重学习器（Meta Learner），基学习器（Base learner）的结果作为该权重学习器的输入，得到最终结果。

如下图所示为Stacking的工作原理：



其中基学习器（Base learner）称为初级学习器，用于结合的学习器（Meta Learner）称为次级学习器。对于测试集，我们首先用初级学习器预测一次，将其输入次级学习器预测，得到最终的预测结果。