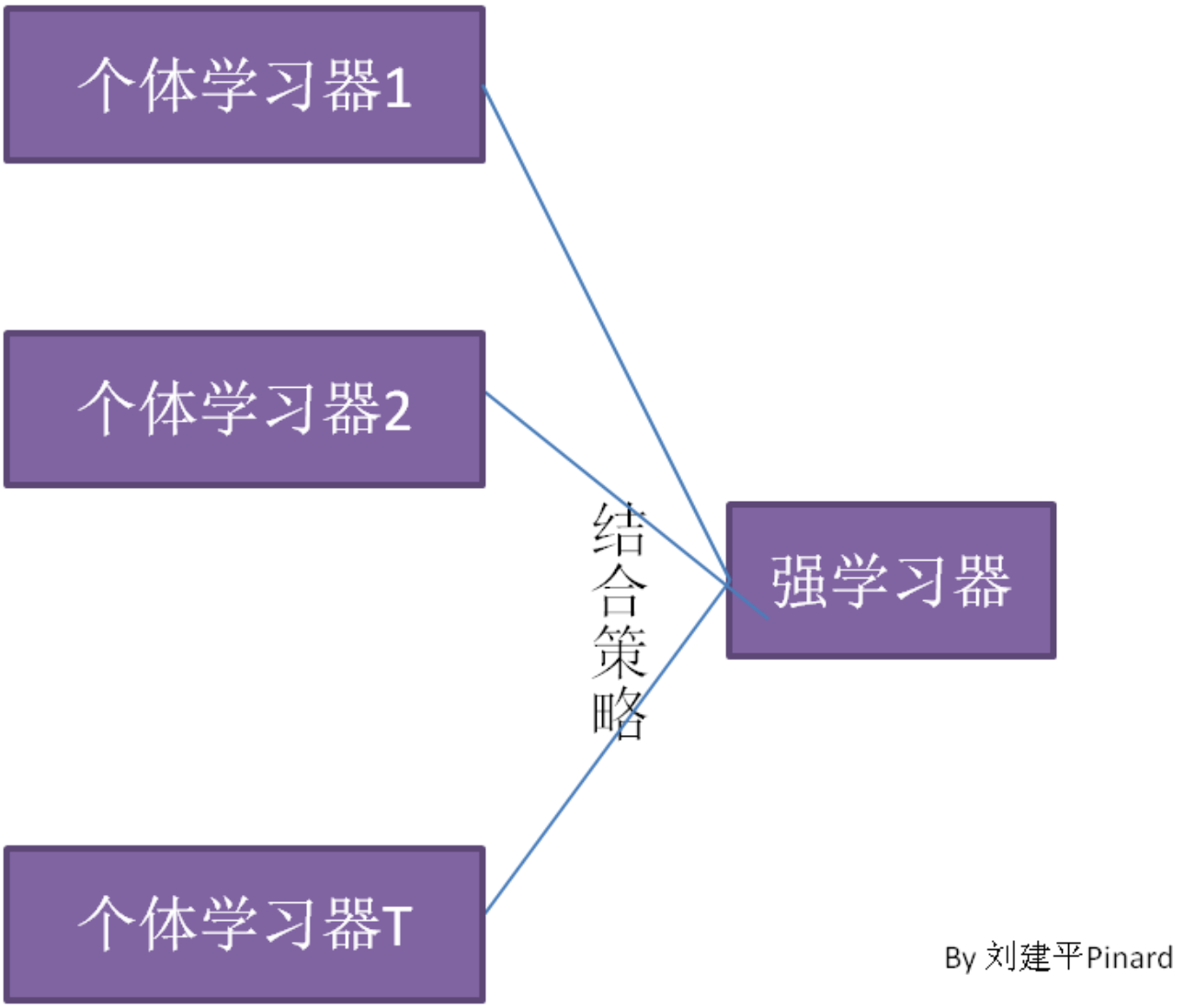
20180329_集成学习原理小结

集成学习(ensemble learning)可以说是现在非常火爆的机器学习方法了。它本身不是一个单独的机器学习算法，而是通过构建并结合多个机器学习器来完成学习任务。也就是我们常说的“博采众长”。集成学习可以用于分类问题集成，回归问题集成，特征选取集成，异常点检测集成等等，可以说所有的机器学习领域都可以看到集成学习的身影。本文就对集成学习的原理做一个总结。

1. 集成学习概述

从下图，我们可以对集成学习的思想做一个概括。对于训练集数据，我们通过训练若干个个体学习器，通过一定的结合策略，就可以最终形成一个强学习器，以达到博采众长的目的。



也就是说，集成学习有两个主要的问题需要解决，第一是如何得到若干个个体学习器，第二是如何选择一种结合策略，将这些个体学习器集合成一个强学习器。

2. 集成学习之个体学习器

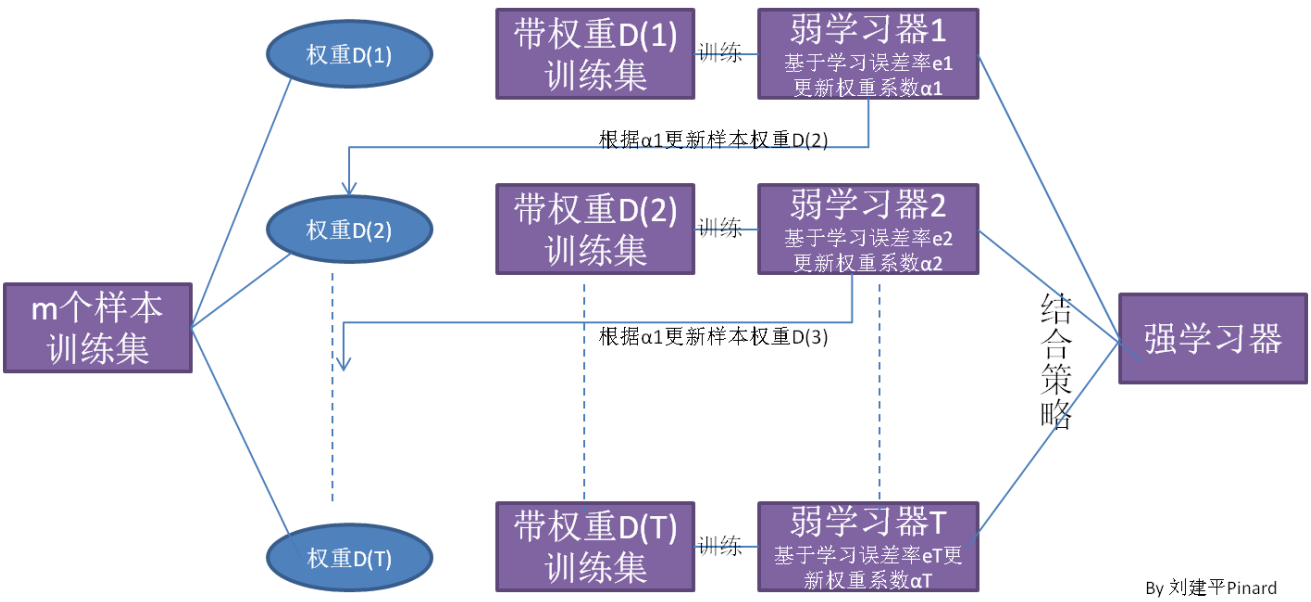
上一节我们讲到，集成学习的第一个问题就是如何得到若干个个体学习器。这里我们有两种选择。

第一种就是所有的个体学习器都是一个种类的，或者说是同质的。比如都是决策树个体学习器，或者都是神经网络个体学习器。第二种是所有的个体学习器不全是一个种类的，或者说是异质的。比如我们有一个分类问题，对训练集采用支持向量机个体学习器，逻辑回归个体学习器和朴素贝叶斯个体学习器来学习，再通过某种结合策略来确定最终的分类强学习器。

目前来说，同质个体学习器的应用是最广泛的，一般我们常说的集成学习的方法都是指的同质个体学习器。而同质个体学习器使用最多的模型是CART决策树和神经网络。同质个体学习器按照个体学习器之间是否存在依赖关系可以分为两类，第一个是个体学习器之间存在强依赖关系，一系列个体学习器基本都需要串行生成，代表算法是boosting系列算法，第二个是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，代表算法是bagging和随机森林（Random Forest）系列算法。下面就分别对这两类算法做一个概括总结。

3. 集成学习之boosting

boosting的算法原理我们可以用一张图做一个概括如下：

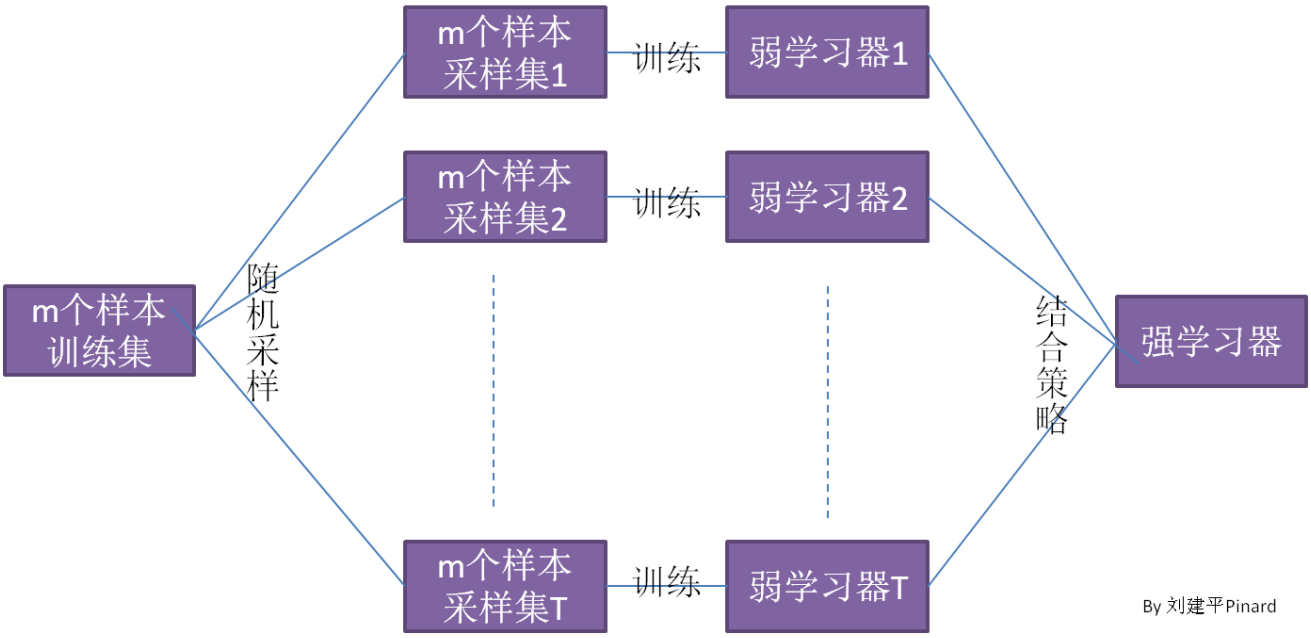


从图中可以看出，Boosting算法的工作机制是首先从训练集用初始权重训练出一个弱学习器1，根据弱学习的学习误差率表现来更新训练样本的权重，使得之前弱学习器1学习误差率高的训练样本点的权重变高，使得这些误差率高的点在后面的弱学习器2中得到更多的重视。然后基于调整权重后的训练集来训练弱学习器2.，如此重复进行，直到弱学习器数达到事先指定的数目T，最终将这T个弱学习器通过集合策略进行整合，得到最终的强学习器。

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)。AdaBoost和提升树算法的原理在后面的文章中会专门来讲。

4. 集成学习之bagging

Bagging的算法原理和 boosting不同，它的弱学习器之间没有依赖关系，可以并行生成，我们可以用一张图做一个概括如下：



从上图可以看出，bagging的个体弱学习器的训练集是通过随机采样得到的。通过T次的随机采样，我们就可以得到T个采样集，对于这T个采样集，我们可以分别独立的训练出T个弱学习器，再对这T个弱学习器通过集合策略来得到最终的强学习器。

对于这里的随机采样有必要做进一步的介绍，这里一般采用的是自助采样法（Bootstap sampling）,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，也就是说下次采样时该样本仍有可能被采集到，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，这样得到多个不同的弱学习器。

随机森林是bagging的一个特化进阶版，所谓的特化是因为随机森林的弱学习器都是决策树。所谓的进阶是随机森林在bagging的样本随机采样基础上，又加上了特征的随机选择，其基本思想没有脱离bagging的范畴。bagging和随机森林算法的原理在后面的文章中会专门来讲。

5. 集成学习之结合策略

在上面几节里面我们主要关注于学习器，提到了学习器的结合策略但没有细讲，本节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{h1,h2,...hT}{h1,h2,...hT}

5.1 平均法

对于数值类的回归预测问题，通常使用的结合策略是平均法，也就是说，对于若干和弱学习器的输出进行平均得到最终的预测输出。

最简单的平均是算术平均，也就是说最终预测是

H(x)=1T∑1Thi(x)H(x)=1T∑1Thi(x)

如果每个个体学习器有一个权重ww，则最终预测是

H(x)=∑i=1Twihi(x)H(x)=∑i=1Twihi(x)

其中wiwi是个体学习器hihi的权重，通常有

wi≥0,∑i=1Twi=1wi≥0,∑i=1Twi=1

5.2 投票法

对于分类问题的预测，我们通常使用的是投票法。假设我们的预测类别是{c1,c2,...cK}{c1,c2,...cK},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(h1(x),h2(x)...hT(x))(h1(x),h2(x)...hT(x))。

最简单的投票法是相对多数投票法，也就是我们常说的少数服从多数，也就是T个弱学习器的对样本x的预测结果中，数量最多的类别cici为最终的分类类别。如果不止一个类别获得最高票，则随机选择一个做最终类别。

稍微复杂的投票法是绝对多数投票法，也就是我们常说的要票过半数。在相对多数投票法的基础上，不光要求获得最高票，还要求票过半数。否则会拒绝预测。

更加复杂的是加权投票法，和加权平均法一样，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。

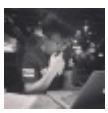
5.3 学习法

上两节的方法都是对弱学习器的结果做平均或者投票，相对比较简单，但是可能学习误差较大，于是就有了学习法这种方法，对于学习法，代表方法是stacking，当使用stacking的结合策略时， 我们不是对弱学习器的结果做简单的逻辑处理，而是再加上一层学习器，也就是说，我们将训练集弱学习器的学习结果作为输入，将训练集的输出作为输出，重新训练一个学习器来得到最终结果。

在这种情况下，我们将弱学习器称为初级学习器，将用于结合的学习器称为次级学习器。对于测试集，我们首先用初级学习器预测一次，得到次级学习器的输入样本，再用次级学习器预测一次，得到最终的预测结果。

以上就是集成学习原理的一个总结，后面会分别对Adaboost, 提升树，bagging和随机森林的算法原理做一个总结，敬请期待。

集成学习（Ensemble Learning)



PENG

在机器学习的有监督学习算法中，我们的目标是学习出一个稳定的且在各个方面表现都较好的模型，但实际情况往往不这么理想，有时我们只能得到多个有偏好的模型（弱监督模型，在某些方面表现的比较好）。集成学习就是组合这里的多个弱监督模型以期得到一个更好更全面的强监督模型，集成学习潜在的思想是即便某一个弱分类器得到了错误的预测，其他的弱分类器也可以将错误纠正回来。

集成学习在各个规模的数据集上都有很好的策略。

- **数据集大：划分成多个小数据集，学习多个模型进行组合**
- **数据集小：利用Bootstrap方法进行抽样，得到多个数据集，分别训练多个模型再进行组合**

Bagging

Bagging是bootstrap aggregating的简写。先说一下bootstrap，bootstrap也称为自助法，它是一种有放回的抽样方法，目的为了得到统计量的分布以及置信区间。具体步骤如下

- **采用重抽样方法（有放回抽样）从原始样本中抽取一定数量的样本**
- **根据抽出的样本计算想要得到的统计量T**
- **重复上述N次（一般大于1000），得到N个统计量T**
- **根据这N个统计量，即可计算出统计量的置信区间**

在Bagging方法中，利用bootstrap方法从整体数据集中采取有放回抽样得到N个数据集，在每个数据集上学习出一个模型，最后的预测结果利用N个模型的输出得到，具体地：分类问题采用N个模型预测投票的方式，回归问题采用N个模型预测平均的方式。

例如**随机森林（Random Forest）**就属于Bagging。随机森林简单地来说就是用随机的方式建立一个森林，森林由很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。

在我们学习每一棵决策树的时候就需要用到Bootstrap方法。在随机森林中，有两个随机采样的过程：对输入数据的**行（数据的数量）与列（数据的特征）**都进行采样。对于行采样，采用有放回的方式，若有N个数据，则采样出N个数据（可能有重复），这样在训练的时候每一棵树都不是全部的样本，相对而言不容易出现overfitting；接着进行列采样从M个feature中选择出m个（m<<M）。最近进行决策树的学习。

预测的时候，随机森林中的每一棵树的都对输入进行预测，最后进行投票，哪个类别多，输入样本就属于哪个类别。这就相当于前面说的，每一个分类器（每一棵树）都比较弱，但组合到一起（投票）就比较强了。

Boosting

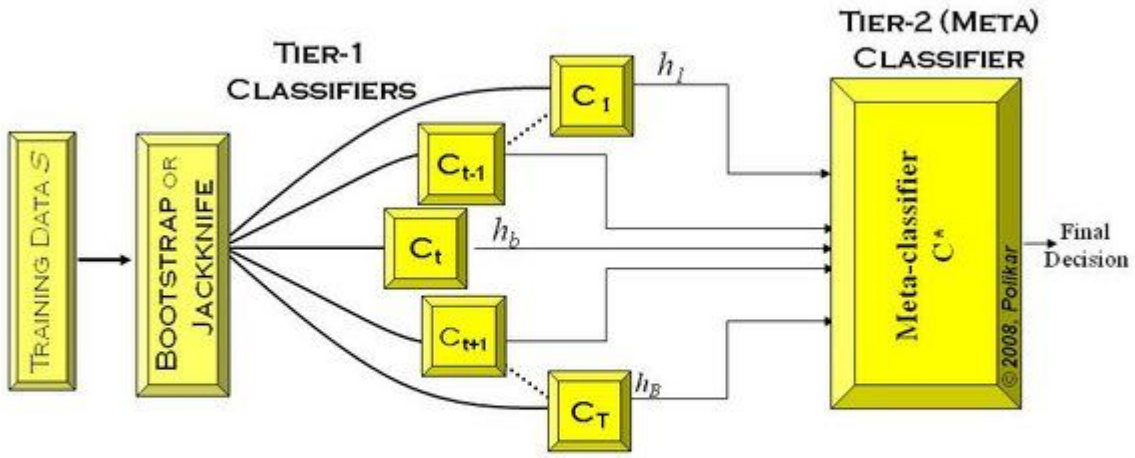
提升方法（Boosting）**是一种可以用来减小监督学习中偏差的机器学习算法。主要也是学习一系列弱分类器，并将其组合为一个强分类器。Boosting中有代表性的是**AdaBoost（Adaptive boosting）算法**：刚开始训练时对每一个训练例赋相等的权重，然后用该算法对训练集训练t轮，每次训练后，对训练失败的训练例赋以较大的权重，也就是让学习算法在每次学习以后更注意学错的样本，从而得到多个预测函数。具体可以参考《统计学习方法》。

之前提到过的**GBDT（Gradient Boost Decision Tree)**也是一种Boosting的方法，与AdaBoost不同，GBDT每一次的计算是为了减少上一次的残差，GBDT在残差减少（负梯度）的方向上建立一个新的模型。可以参考Gradient Boosting - 知乎专栏。

Stacking

Stacking方法是指训练一个模型用于组合其他各个模型。首先我们先训练多个不同的模型，然后把之前训练的各个模型的输出为输入来训练一个模型，以得到一个最终的输出。理论上，Stacking可以表示上面提到的两种Ensemble方法，只要我们采用合适的模型组合策略即可。但在实际中，我们通常使用logistic回归作为组合策略。

如下图，先在整个训练数据集上通过bootstrap抽样得到各个训练集合，得到一系列分类模型，称之为Tier 1分类器（可以采用交叉验证的方式学习），然后将输出用于训练Tier 2 分类器。

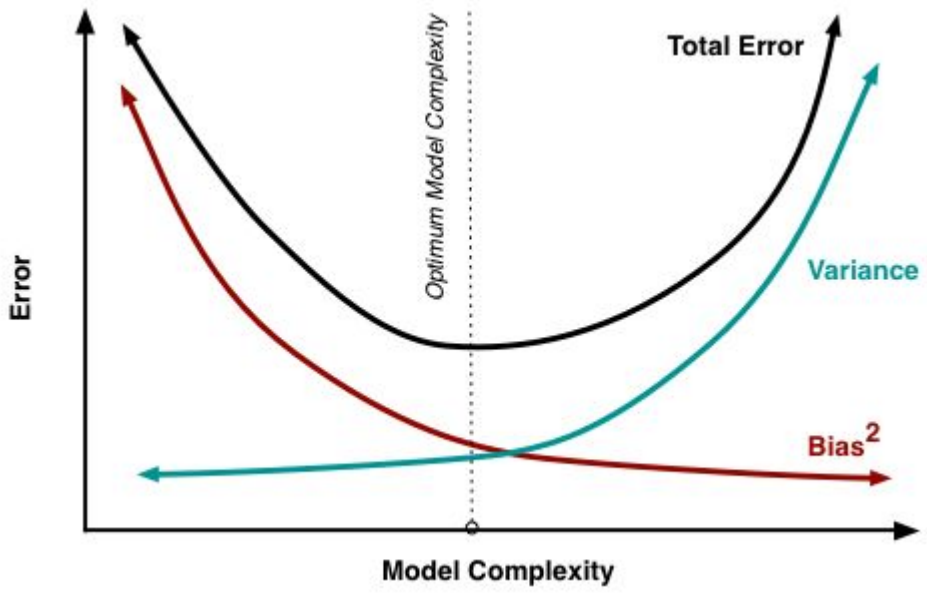


Bagging与Boosting

Bagging和Boosting采用的都是**采样-学习-组合**的方式，但在细节上有一些不同，如

- **Bagging中每个训练集互不相关，也就是每个基分类器互不相关，而Boosting中训练集要在上一轮的结果上进行调整，也使得其不能并行计算**
- **Bagging中预测函数是均匀平等的，但在Boosting中预测函数是加权的**

在算法学习的时候，通常在bias和variance之间要有一个权衡。bias与variance的关系如下图，因而模型要想达到最优的效果，必须要兼顾bias和variance，也就是要采取策略使得两者比较平衡。



从算法来看，Bagging关注的是多个基模型的投票组合，保证了模型的稳定，因而每一个基模型就要相对复杂一些以降低偏差（比如每一棵决策树都很深）；而Boosting采用的策略是在每一次学习中都减少上一轮的偏差，因而在保证了偏差的基础上就要将每一个基分类器简化使得方差更小。

参考

- 《统计学习方法》 李航
- Ensemble learning 概述
- 机器学习-组合算法总结
- 为什么xgboost/gbdt在调参时为什么树的深度很少就能达到很高的精度？ - 知乎