1.集成学习的概念

构建并通过多个学习器来完成学习任务的方式。也称为多分类器系统或者基于委员会的学习。

同质集成：只包含同类个体学习器。（称为基学习器）。异质集成：学习器由不同的学习算法生成。（称为组件学习器）

核心：如何产生“好而不同”的个体学习器。

根据个体学习器的生产方式，分为两类：

个体学习器之间存在强相关，必须串行生成的序列化方法。如 Boosting

个体学习器之间不存在依赖关系，可以同时生成的并行化方法。如 Bagging、随机森林。

原文链接：https://blog.csdn.net/m0_38019841/article/details/85100588

2.个体学习的概念

基于现有的学习算法从训练数据产生的一个模型。也称基学习器，组件学习器，弱学习器。

3.boosting，bagging，stacking

(1)bagging 集成方法有很多种，一种叫做 bagging，bagging 的思想是，我把我的数据做一点微小的调整，就得到了一个跟原来不一样的数据集，我就能多训练一个模型出来，模型的数量多了，解释力自然就增强了。比如说我原来有 100 个人的数据，其中有两个分别叫 Tony 和 Lily，我把 Tony 这条数据删掉，用 Lily 的数据来替换，这样就得到了一个跟原来不一样的全新的数据集，这个过程叫做 Bootstrap。

每一个 Bootstrap 数据集都能用来训练一次模型，所以我们重复这个过程，比如重复 1000 次，一次是 Tony 替代 Cici，一次是 Ivy 替代 Yuki，这样每一次都是不一样的数据，也就可以训练 1000 次，得到了 1000 个决策树，我们把这 1000 个决策树打包到一起作为我们最终的模型，这个打包就叫做 bagging。

一般我们会把 bagging 跟随机森林一起叠加使用，在数据点的处理上，我们使用 bagging 来创造许多组（比如说 1000 组）bootstrap 数据，对于每一组数据，我们使用随机森林来训练模型，最后再把所有模型的预测结果 bagging 起来。对于分类问题：由投票表决产生的分类结果；对于回归问题，由 k 个模型预测结果的均值作为最后预测的结果（所有模型的重要性相同）

2 boosting 第二种集成的方法是 boosting，boosting 跟 bagging 一样都属于集成的思想，本质上都是训练很多模型，用数量堆积出质量。还是举 1000 个 model，100 个 variable 的例子，bagging 是训练 1000 个等价的模型，比如说用随机森林，这些模型都是同样随机从 100 个里面选 10 个 variable 出来训练，每一个模型之间是同一级别的、互不干扰的。

但 boosting 的思路和 bagging 不同，boosting 里每一个模型都是基于上一个模型来进行优化，它的 核心思想 是训练 1000 个模型，每一个模型在上一个模型的基础上再好一点点，

比如说第一个模型的 RSS 是 10，这时候我们基于第一个模型定个小目标，先让 RSS 减

到9，这就是我们的第二个模型，第三个模型的RSS减到8.5…如此往复，得到1000个model，再综合这1000个model得到最终的模型。

3 stacking 第三种也是最后一种集成方法是stacking，stacking在字面上更好理解一点，就是堆积、堆砌。如果说bagging和boosting一般都是在决策树的范围内使用，stacking的运用范围会更广一点。例如对于同一个问题，假设还是预测一个人是不是柠檬精，我们首先用Logistic回归跑一遍，再用LDA跑一遍，再用SVM跑一遍，最后用决策树再跑一遍，然后我们用一种方法，比如说是majority polling或是权重加成把这些结果结合到一起，这就是一个stacking的过程

stacking 的一个使用场景是我们有很多专家小组，每个小组都训练出了一个自己的模型，当这些模型难以取舍的时候，就干脆一口气打包带走，用stacking把这些模型结合起来，这样谁也不得罪，而且通常也能取得较好的效果。另外在参加各种建模比赛的时候，为了追求一点点精度，我们可以多训练几个模型然后结合起来，有时候也能得到很好的效果。

Bagging 和 Boosting 的主要区别：

样本选择上: Bagging采取Bootstraping的是随机有放回的取样，Boosting的每一轮训练的样本是固定的，改变的是买个样的权重。

样本权重上：Bagging采取的是均匀取样，且每个样本的权重相同，Boosting根据错误率调整样本权重，错误率越大的样本权重会变大

预测函数上：Bagging所以的预测函数权值相同，Boosting中误差越小的预测函数其权值越大。

并行计算: Bagging 的各个预测函数可以并行生成;Boosting的各个预测函数必须按照顺序迭代生成.

将决策树与以上框架组合成新的算法

Bagging + 决策树 = 随机森林

AdaBoost + 决策树 = 提升树

gradient + 决策树 = GDBT

参考链接：https://mp.weixin.qq.com/s/0Qng1Z-9HKVirNul_eAAqw

https://www.cnblogs.com/onemorepoint/p/9264782.html

4.理解不同的结合策略（平均法，投票法，学习法）

学习器的结合能带来以下优点

●统计方面，由于学习任务的假设空间往往很大，可能有多个假设在训练集上达到同等性能，此时若单个学习器可能因为误选而导致泛化性能不佳，结合多个学习器则会减少风险。

●计算方面，通过多次运行之后进行结合，可降低陷入局部最小的风险。

●表示方面，通过结合多个学习器，相应的假设空间也有所扩大，有可能可以到达更好的效果。

①平均法

对于数值型输出，最常见的结合策略是使用平均法。

●简单平均法

●加权平均法

加权平均法是简单平均法的特例，被广泛运用于集成学习。但加权平均算法也存在一定的缺陷，因为加权平均法的权重一般是从训练数据中学习而得，现实任务中的训练样本通常不充分或存在噪声，这将使得学出的权重不完全可靠。尤其当集成规模较大时，要学习的权重较多，就容易导致过拟合。

如何选择平均法？

一般而言，在个体学习器性能相差较大时宜使用加权平均法，而在个体学习器性能相近时宜使用简单平均法。

②投票法

对于分类问题，最常见的结合策略就是投票法。

●绝对多数投票法，提供了“拒绝预测”的选项

●相对多数投票法

●加权投票法

需要注意的是，若基学习器不同，其类概率值不能直接进行比较，通常需要将其转化为类标记输出，然后在投票。

③学习法

当训练数据很多时，一种更为强大的结合策略是使用“学习法”，即通过另一个学习器来进行结合。Stacking 是学习法的典型代表。我们将个体学习器称为初级学习器，结合学习器称为次级学习器。

有研究表明，将初级学习器输出的类别概率输入次级学习器作为属性，用多响应线性回归（MLR）作为次级学习算法效果较好，在 MLR 中使用不同属性集更佳。

https://blog.csdn.net/m0_38019841/article/details/85100588

5.随机森林的思想

随机森林利用随机的方式将许多决策树组合成一个森林，每个决策树在分类的时候投票决定测试样本的最终类别。

两个随机的过程：随机选择样本，随机选择特征。

随机选择样本

给定一个训练样本集，数量为 N，我们使用有放回采样到 N 个样本，构成一个新的训练集。注意这里是有放回的采样，所以会采样到重复的样本。详细来说，就是采样 N 次，每次采样一个，放回，继续采样。即得到了 N 个样本。

随机选择特征

在随机森林中，我们不计算所有特征的增益，而是从总量为 M 的特征向量中，随机选择 m 个特征，其中 m 可以等于 sqrt(M)，然后计算 m 个特征的增益，选择最优特征（属性）【nformationGain（ID3） 或者 Gain Ratio（C4.5）】。注意，这里的随机选择特征是无放回的选择！

参考链接：https://blog.csdn.net/m0_38019841/article/details/85100588

6.随机森林的推广

Extremely Randomized Trees 区别：

对于每个决策树的训练集，RF 采用的是随机采样 bootstrap 来选择子集作为每个决策树的训练集，而 extra trees 一般不采用随机采样，即每个决策树采用原始训练集。

在选定了划分特征后，RF 的决策树会基于信息增益，基尼系数，均方差之类的原则，选择一个最优的特征值划分点，这和传统的决策树相同。但是 extra trees 比较的激进，他会随机的选择一个特征值来划分决策树。

Isolation Forest 特点：

iForest 用部分模型不隔离所有正常点的情况下效果很好，并且模型的建立仅使用很小的样本数量（子采样数量远远小于原始训练集的数量），因为 iForest 目标是异常点检测，只需要部分样本就可以将异常点区分出来；

iForest 中的树的建立是任意选择一个特征，然后在该特征中任意选择一个值作为划分左右子树的标准；

iForest 使用不放回的随机子采样策略；

7.随机森林的优缺点

（实践使用较少，对于优缺点的体验不直观深刻）

优点：

能够处理很高维度（feature 很多）的数据，并且不用做特征选择，对数据集的适应能力强：既能处理离散型数据，也能处理连续型数据，数据集无需规范化

随机选择样本导致的每次学习决策树使用不同训练集，所以可以一定程度上避免过拟合；

本身精度比大多数单个算法要好

在测试集上表现良好，由于两个随机性的引入，使得随机森林不容易陷入过拟合（样本随机，特征随机）

由于树的组合，使得随机森林可以处理非线性数据，本身属于非线性分类（拟合）模型

由于有袋外数据（OOB），可以在模型生成过程中取得真实误差的无偏估计，且不损失训练数据量

可以处理缺省值（单独作为一类），不用额外处理

由于每棵树可以独立、同时生成，容易做成并行化方法

由于实现简单、精度高、抗过拟合能力强，当面对非线性数据时，适于作为基准模型

缺点：

在某些噪音较大的分类或回归问题上会过拟合；

对于有不同级别的属性的数据，级别划分较多的属性会对随机森林产生更大的影响，所以随机森林在这种数据上产出的属性权值是不可信的。

原文链接：https://blog.csdn.net/m0_38019841/article/details/85100588

8.随机森林在 sklearn 中的参数解释

https://blog.csdn.net/lynn_001/article/details/85340412

https://blog.csdn.net/m0_38019841/article/details/85100588

https://blog.csdn.net/qq_29750461/article/details/81516008

https://blog.csdn.net/lynn_001/article/details/85337784

https://blog.csdn.net/qq_37334135/article/details/86766014

9.随机森林的应用场景

数据维度相对较低（几十维），对准确性有一定要求

随机森林在多数数据集上都有不错的表现，相对来说，出现预测效果很差的情况较少。可以作为模型的 baseline 作为参考。