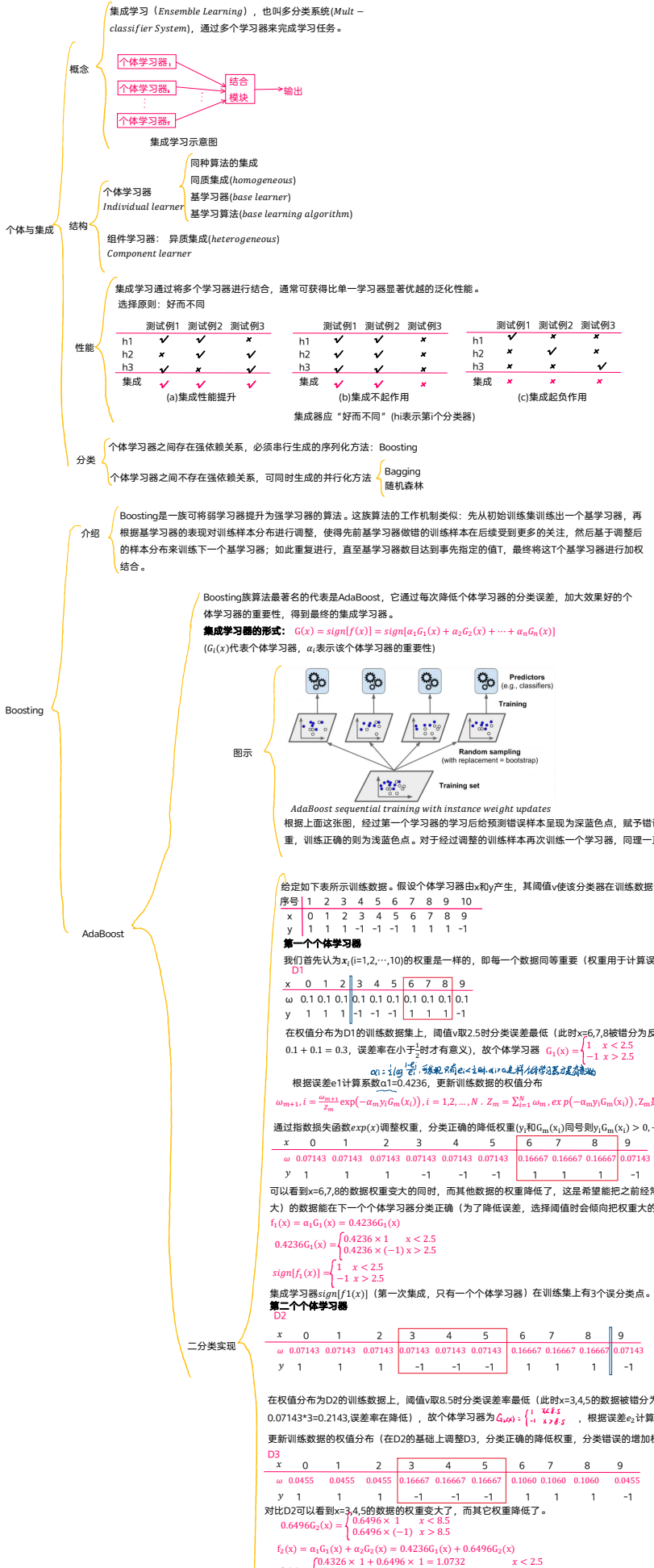


集成学习



对比D2可以看到x=3,4,5的数据的权重变大了，而其它权重降低了。

$$0.6496 G_2(x) = \begin{cases} 0.6496 \times 1 & x < 8.5 \\ 0.6496 \times (-1) & x > 8.5 \end{cases}$$

$$f_2(x) = \alpha_1 G_1(x) + \alpha_2 G_2(x) = 0.4236 G_1(x) + 0.6496 G_2(x)$$

$$f_2(x) = \begin{cases} 0.4326 \times 1 + 0.6496 \times 1 = 1.0732 & x < 2.5 \\ 0.4326 \times (-1) + 0.6496 \times 1 = 0.226 & 2.5 < x < 8.5 \\ 0.4326 \times (-1) + 0.6496 \times (-1) = -1.0732 & x > 8.5 \end{cases}$$

$$\text{sign}[f_2(x)] = \begin{cases} 1 & x < 8.5 \\ -1 & x > 8.5 \end{cases}$$

分类器 $\text{sign}[f_2(x)]$ 在训练数据集上有3个误分类点

第三个个体学习器

D3

x	0	1	2	3	4	5	6	7	8	9
w	0.0455	0.0455	0.0455	0.16667	0.16667	0.16667	0.1060	0.1060	0.1060	0.0455
y	1	1	1	-1	-1	-1	1	1	1	-1

在权重分布为D3的训练数据上，阈值v取5.5时分类误差率最低(x=0,1,2,9被分类错误 $\alpha_3 = 0.1820$ ，误差率又降低了)，故个体学习器为 $G_3(x)$ ， $\alpha_3=0.7514$ ，更新训练数据的权重分布

D4

x	0	1	2	3	4	5	6	7	8	9
w	0.125	0.125	0.125	0.102	0.102	0.102	0.065	0.065	0.065	0.125
y	1	1	1	-1	-1	-1	1	1	1	-1

$$f_3(x) = \alpha_1 G_1(x) + \alpha_2 G_2(x) + \alpha_3 G_3(x)$$

$$f_3(x) = \begin{cases} 0.3218 & x < 2.5 \\ -0.5254 & 2.5 < x < 5.5 \\ 0.9774 & 5.5 < x < 8.5 \\ -0.3218 & x > 8.5 \end{cases}$$

$$\text{sign}[f_3(x)] = \begin{cases} 1 & x < 2.5 \\ -1 & 2.5 < x < 5.5 \\ 1 & 5.5 < x < 8.5 \\ -1 & x > 8.5 \end{cases}$$

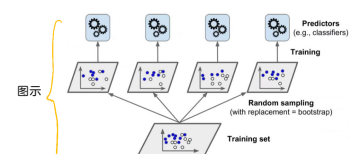
最终结果：

x	0	1	2	3	4	5	6	7	8	9
w	0.125	0.125	0.125	0.102	0.102	0.102	0.065	0.065	0.065	0.125
y	1	1	1	-1	-1	-1	1	1	1	-1

分类器 $\text{sign}[f_3(x)]$ 在训练数据集上有0个误分类点。

目的：欲得到泛化性能强的集成，集成中的个体学习器应尽可能相互独立；虽然独立在现实任务中无法做到，但可以设法使基学习器尽可能具有较大的差异。

思路：在训练集进行子抽样组成每个基模型，所需要的子训练集对所有基模型预测的结果进行综合，产生最终的预测结果。



bagging training set sampling and training

解决问题：为了获得好的集成，我们还希望个体学习器不能太差。如果采样出的每个子集都完全不同，则每个基学习器只用到了一部分训练数据，甚至不足以进行有效学习，这显然无法确保产出比较好的基学习器。为了解决这个问题，我们可以考虑使用相互有交叠的采样子集。

介绍：Bagging是并行式集成学习方法最著名的代表，它直接基于自助采样法在包含m个样本的数据集进行m次随机采样操作。最后我们可采集出T个含m个训练样本的采样集，然后基于每个采样集训练出一个基学习器，再将基学习器进行结合。结合时，Bagging通常对分类任务使用简单投票法，对回归任务使用简单平均法。

计算复杂度：假定基学习器的计算复杂度为 $O(m)$ ，则Bagging的复杂度大致为 $T(O(m) + O(s))$ 。考虑到采样与投票/平均过程的复杂度 $O(s)$ 很小，而T通常是一个不太大的常数，因此，训练一个Bagging集成与直接使用基学习算法训练一个学习器的复杂度同阶，这说明Bagging是一个很高效的集成学习算法。另外，与标准AdaBoost只适用于二分类任务不同，Bagging能不经修改地用于多分类、回归等任务。

包外估计：自助采样过程还给Bagging带来了另一个优点：由于每个基学习器只是用了初始训练集中约63.2%的样本，剩下36.8%的样本可用作验证集来对泛化性能进行“包外估计”(out-of-bag estimate)，简称oob。为此需记录每个基学习器所使用的训练样本。不妨令 D^i 表示第i个基学习器使用的训练样本集，令 $H^{oob}(x)$ 表示对样本x的包外预测，即仅考虑那些未使用x训练的基学习器在x上的预测，有

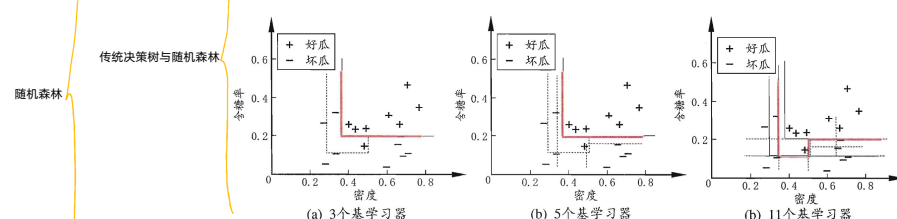
$$H^{oob}(x) = \arg \max_{y \in Y} \sum_{i=1}^T \mathbb{I}(h_i(x) = y) \cdot \mathbb{I}(x \in D_i^c)$$

则Bagging泛化误差的包外估计为

$$\epsilon^{oob} = \frac{1}{|D|} \sum_{(x,y) \in D} \mathbb{I}(H^{oob}(x) \neq y)$$

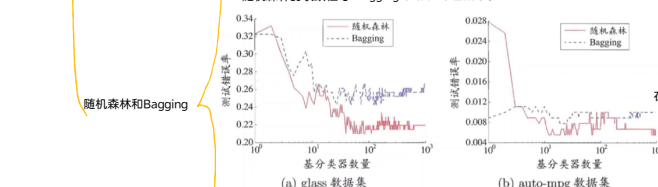
介绍：随机森林(Random Forest)是Bagging的一个扩展变体。RF在以决策树为基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。样本和节点判断的特征都随机选择，即添加样本扰动，又添加属性扰动。

具体来说，传统决策树在选择划分属性时是在当前结点的属性集合(假定有d个属性)中选择一个最优属性；而在RF中对基决策树的每个结点，先从该结点的属性集合中随机选择一个包含k个属性的子集，然后再从这个子集中选择一个最优属性用于划分。这里的参数k控制了随机性的引入程度：若令 $k=d$ ，则基决策树的构建与传统决策树相同；若令 $k=1$ ，则是随机选择一个属性用于划分；一般情况下，推荐值 $k = \log_2 d$



虽然随机森林对 Bagging 只做了小改动，但是与 Bagging 中基学习器的“多样性”仅通过样本扰动(通过对初始训练集采样)而来不同，随机森林中基学习器的多样性不仅来自样本扰动，还来自属性扰动，这就使得最终集成的泛化性能可通过个体学习器之间差异的增加而进一步提升。

随机森林的收敛性与 Bagging 相似如下图所示：



在两个UCI数据集上，集成规模对随机森林与Bagging的影响

随机森林的起始性能往往相对较低，特别是在集成中只包含一个基学习器时。这很容易理解，因为通过引入属性扰动，随机森林中个体学习器的性能往往有所降低。然而，随着个体学习器数目的增加，随机森林通常会收敛到更低的泛化误差。另外，随机森林的训练效率常优于Bagging，因为在个体决策树的构建过程中，Bagging使用的是“确定型”决策树，在选择划分属性时要对结点的所有属性进行考察，而随机森林使用的“随机型”决策树只需考察一个属性子集。

Bagging与随机森林

结合策略

(a) glass 数据集

(b) auto-mpg 数据集

随机森林的起始性能往往相对较低，特别是在集成中只包含一个基学习器时。这很容易理解，因为通过引入属性扰动，随机森林中个体学习器的性能往往有所降低。然而，随着个体学习器数目的增加，随机森林通常会收敛到更低的泛化误差。另外，随机森林的训练效率常优于Bagging，因为在个体决策树的构建过程中，Bagging使用的是“确定型”决策树，在选择划分属性时要对结点的所有属性进行考察，而随机森林使用的“随机型”决策树则只需考察一个属性子集。

平均法

简单平均法: $H(x) = \frac{1}{T} \sum_{t=1}^T h_t(x)$

加权平均法

$H(x) = \sum_{t=1}^T w_t h_t(x)$

其中 w_t 是个体学习器 h_t 的权重，通常要求 $w_t \geq 0, \sum_{t=1}^T w_t = 1$

必须使用非负权重才能确保集成性能由于单一最佳个体学习器，因此在集成学习中一般对学习器的权重施以非负约束。

性能比较

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，尤其是对规模比较大的集成来说，要学习的权重比较多，容易导致过拟合。因此，实验和应用均显示出，加权平均法未必一定优于简单平均法；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。

投票法

绝对多数投票法

若某标记得票超过半数，则预测为该标记，否则拒绝预测

标准的绝对多数投票法提供了“拒绝预测”选项，这在可靠性要求较高的学习任务中是一个很好的机制。但若学习任务要求必须提供预测结果则绝对多数投票法将退化为相对多数投票法。因此，在不允许拒绝预测的任务中，绝对多数、相对多数投票法统称为“多数投票法”。

相对多数投票法

预测为得票最多的标记，若同时有多个标记获得最高票，则随机中选择一个。

加权投票法

$H(x) = \text{cargy} \max_{c_j} \sum_{t=1}^T w_t h_t^j(x)$

与加权平均法类似， w_t 是 h_t 的权重，通常 $w_t \geq 0, \sum_{t=1}^T w_t = 1$

软投票与硬投票

在现实任务中，不同类型的个体学习器可能产生不同类型的 $h_t(x)$ 值，常见的有：
类标记： $h_t^j(x) \in \{0, 1\}$ ，若 h_t 将样本 x 预测为类别 c_j ，则取值为1，否则为0。使用类标记的投票亦称“硬投票”（hard voting）。
类概率： $h_t^j(x) \in [0, 1]$ ，相当于对后验概率 $P(c_j|x)$ 的一个估计。使类概率的投票亦称“软投票”（soft voting）。

5个不同的个体学习器对于一个样本的分类投票结果有：

模型编号	被分到A的概率	被分到B的概率
1	99%	1%
2	49%	51%
3	40%	60%
4	90%	10%
5	30%	70%

硬投票

1) 少数服从多数：
A得两票，B得三票，最终预测结果为B
2) 但在很多情况下，少数服从多数并不是最合理的方式，应考虑加入权重，由此引出soft集成。

软投票

要求集合的每一个模型都能估计概率：
A - $(0.99+0.49+0.4+0.9+0.3)/5 = 0.616$
B - $(0.01+0.51+0.6+0.1+0.7)/5 = 0.384$
最终预测结果为A

能够估计概率的模型

1) 逻辑回归，本身就是基于概率模型的
2) KNN，k个近邻中数量最多的那个类的数量除以k就是概率
3) 决策树，叶子节点中类数量最大的类，概率就是数量最大的那个类除以所有叶子节点的类
4) SVC，SVC本身是没有考虑概率的，它是寻找一个margin的最大值。但是也是可以计算过，不过需要消耗大量的计算资源

定义

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。这里我们把个体学习器称为初级学习器，用于结合的学习器称为次级学习器或元学习器（meta-learner）。

思路

Stacking的策略是把训练样本集分为两部分。一部分用来训练初级学习器，用初级学习器的输出作为次级学习器的输入，由此得到最终的预测结果。

学习法Stacking

结构

Aggregating predictions using a blending predictor

训练初级学习器

Train

Subset 1

Subset 2

Split

Training set

Training the first layer

训练次级学习器

Layer 3

Layer 2

Layer 1

Predictions in a multilayer stacking ensemble

New instance

评价

每个模型具有非常多的超参数使得stacking复杂很多，这种复杂性会导致其容易产生过拟合。比起这种更加复杂的网络，往往会选择使用神经网络。