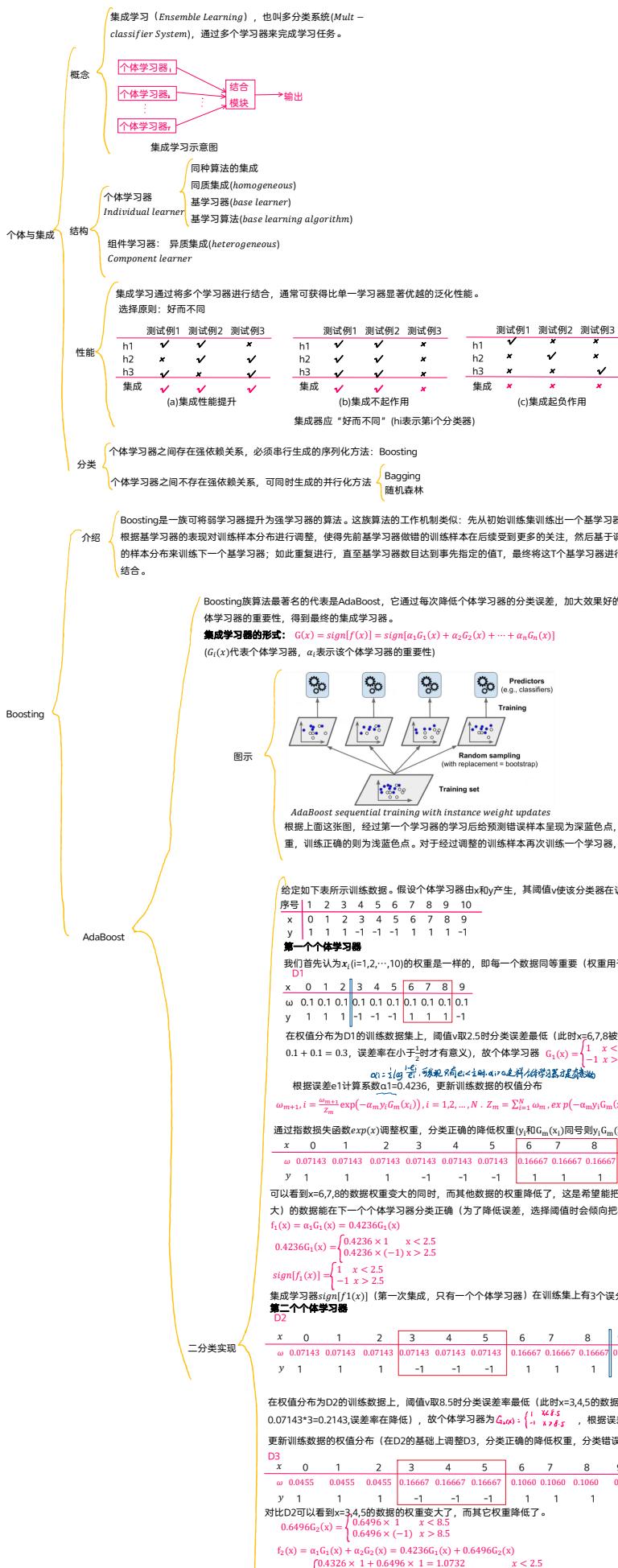


# 集成学习



对比如D2可以看到到x=3,4,5时数据的权重变大了，而其它权重降低了。

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

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

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

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

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

在权值分布为D3的训练数据上，阈值 $v=5.5$ 时分类误差率最低( $x=0,1,2,9$ 被分类错误 $e_3 = 0.1820$ ，误差率又降低了)，故个体学习器为 $\alpha_3G_3(x) = \begin{cases} 1 & x < 8.5 \\ -1 & x > 8.5 \end{cases}$ ， $\alpha_3=0.7514$ ，更新训练数据的权值分布

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

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

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

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

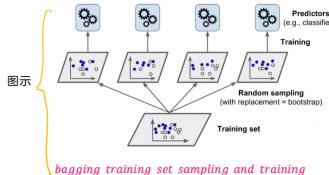
最终结果：

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。



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能不修改地用于多分类、回归等任务。

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 $D_i^t$ 表示 $i$ 实际使用的训练样本集，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

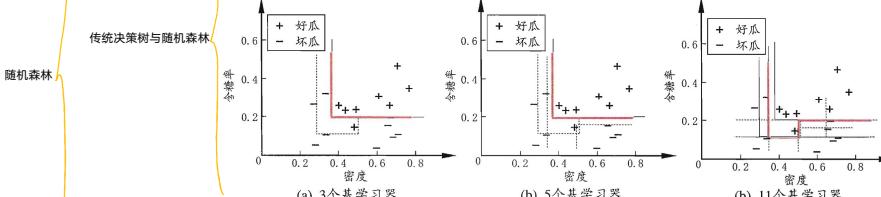
$$H^{oob}(x) = \arg \max_{y \in Y} \sum_{t=1}^T I\{h_t(x) = y\} \cdot I\{x \notin D_i^t\}$$

则Bagging泛化误差的包外估计为

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

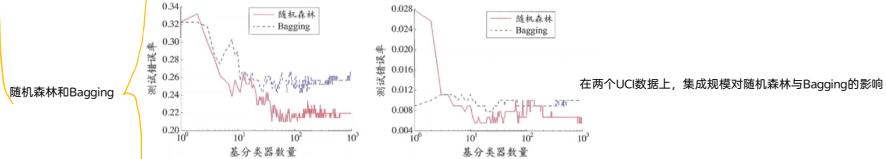
随机森林(Random Forest)是Bagging的一个扩展变体。RF在以决策树为基学习器构建Bagging集成的基础上，进一步在决策树的训练过程中引入了随机属性选择。样本和节点判断的特征都随机选择，即添加样本扰动，又添加属性扰动。

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度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：



随机森林的起始性能往往相对较差，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使用的是一种“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。

## Bagging与随机森林

