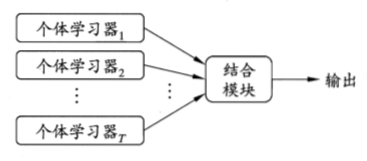
集成学习

1、个体与集成

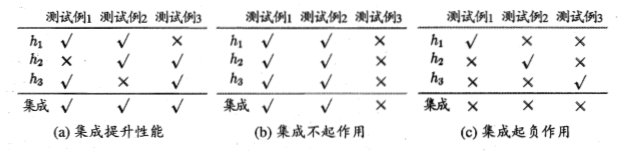
集成学习通过构建并结合多个学习器来完成学习任务。集成学习的一般结构：先产生一组“个体学习器”，再用某种策略将它们结合起来。个体学习器通常由一个现有的学习算法从训练数据产生，如C4.5决策树算法、BP神经网络算法等，此时集成中只包含同种类型的个体学习器（同质的-基学习器）、也可包含不同类型的个体学习器（异质的-组件学习器）。



集成学习通过将多个学习器进行结合，常可获得比单一学习器显著优越的泛化性能。这对“弱学习器”尤为明显，因此集成学习的很多理论研究都是针对弱学习器进行的，但需注意的是，虽然理论上来说，使用弱学习器集成足以获得好的性能，但在实践中，出于种种考虑，例如希望使用较少的个体学习器或重用关于常见学习器的一些经验等，人们往往会使用比较强的学习器。

在一般经验中，如果把好坏不等的东西掺到一起，那么通常结果会是比最坏的要好一些，比最好的要怀一些。集成学习把多个学习器结合起来，如何获得比最好的单一学习器更好的性能呢？

考虑一个简单的例子：在二分类任务中，假定三个分类器在三个测试样本上的表现如图所示：符号分别表示分类正确、分类错误，集成学习的结果通过投票法产生，即“少数服从多数”。



这个简单的例子表明：要获得好的集成，个体学习器应“好而不同”，即个体学习器要有一定的“准确性”，即学习器不能太坏，并且要有“多样性”，即学习器间具有差异。

考虑二分类问题和真实函数，假定基分类器的错误率为，则对每个基分类器有：

假定集成通过简单投票法结合T个分类器（一般是奇数），若有超过半数的基分类器正确，则集成分类器就正确：

假设基分类器的错误率相互独立（理想情况吧？），则由Hoeffding不等式可知，集成的错误率为：

从上式可以看出，随着集成中个体分类器数目T的增大，集成的错误率将指数级下降，最终趋向于零。

事实上，个体学习器的“准确性”和“多样性”本身存在冲突。一般的，准确性很高之后，要增加多样性就需要牺牲准确性。事实上，如何产生并结合“好而不同”的个体学习器，恰是集成学习研究的核心。

根据个体学习器的生成方式，目前的集成学习方法大致可分为两大类：

个体学习器间存在强依赖关系、必须串行生成的序列化方法（Boosting）

个体学习器间不存在强依赖关系、可同时生成的并行化方法（Bagging和随机森林）

2、Boosting

啊

3、Bagging与随机森林

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。给定一个训练数据集，一种可能的做法是对训练样本进行采样，产生若干个不同的子集，再从每个数据子集中训练出一个基学习器。这样，由于训练数据不同，我们获得的基学习器可望具有比较大的差异。然而，为了获得好的集成，我们同时还希望个体学习器不能太差。如果采用出的每个子集都完全不同，则每个基学习器只用到了一小部分训练数据，甚至不足以进行有效学习，这显然无法确保产生出较好的基学习器。为解决这个问题，我们可考虑使用相互有交叠的采样子集。

3.1、Bagging

Bagging是并行式集成学习方法最著名的代表。给定包含m个样本的数据集，我们先随机取出一个样本放入采样集中，再把该样本放回初始数据集，使得下次采样时该样本仍然有可能被选中，这样，经过m次随机采样操作，我们得到含m个样本的采样集，初始训练集中有的样本在采样集里多次出现，有的则从未出现。初始训练集中约有63.2%的样本出现在采样集中。

照这样，我们可采样出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，再将这些基学习器进行结合。这就是Bagging的基本流程。在对预测输出进行结合时，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。若分类预测时，出现两个类收到同样票数的情形，则最简单的做法是随机选择一个，也可进一步考察学习器投票的置信度来确定最终胜者。Bagging的算法描述如图所示：

|  |
| --- |
| 输入：训练集 |
| 基学习算法： |
| 训练轮数： |
| 过程： |
| 1、for do |
| 2、 |
| 3、end for |
| 输出： |

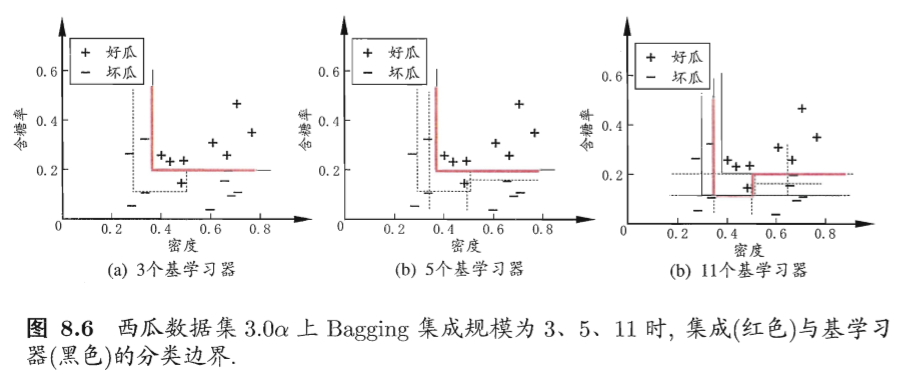
？？？

值得一提的是，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使用了初始训练集中约63.2%的样本，剩下约36.8%的样本可用作验证集来对泛化性能进行“包外估计”。为此，需要记录每个基学习器所使用的训练样本。不妨令表示实际使用的训练集样本，令表示对样本的包外预测，即仅考虑那些未使用x训练的基学习器在x上的预测，则有：

事实上，包外样本还有很多其他用途。例如当基学习器是决策树是，可使用包外样本来辅助剪枝，或用于估计决策树中各结点的后验概率以辅助对零训练样本结点的处理；当基学习器是神经网络时，可使用包外样本来辅助早期停止以减小过拟合风险。

从偏差-方差分解的角度看，Bagging主要关注降低方差，因此它在不剪枝决策树、神经网络等易受样本扰动的学习器上效用更为明显。我们以基于信息增益划分的决策树为基学习器，运行Bagging算法，不同规模的集成及其基学习器所对应的分类边界如图所示。

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **密度** | **含糖率** | **好瓜** |
| 1 | 0.697 | 0.460 | 是 |
| 2 | 0.774 | 0.376 | 是 |
| 3 | 0.634 | 0.264 | 是 |
| 4 | 0.608 | 0.318 | 是 |
| 5 | 0.556 | 0.215 | 是 |
| 6 | 0.403 | 0.237 | 是 |
| 7 | 0.481 | 0.149 | 是 |
| 8 | 0.437 | 0.211 | 是 |
| 9 | 0.666 | 0.091 | 否 |
| 10 | 0.243 | 0.267 | 否 |
| 11 | 0.245 | 0.057 | 否 |
| 12 | 0.343 | 0.099 | 否 |
| 13 | 0.639 | 0.161 | 否 |
| 14 | 0.657 | 0.198 | 否 |
| 15 | 0.360 | 0.370 | 否 |
| 16 | 0.593 | 0.042 | 否 |
| 17 | 0.719 | 0.103 | 否 |

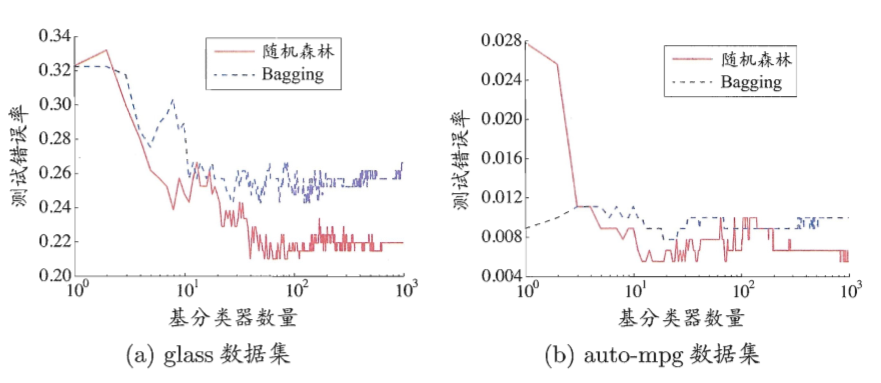


3.2、随机森林

随机森林是Bagging的一个扩展变体。随机森林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。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；而在随机森林中，对基决策树的每个结点，先从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，则基决策树的构建与传统决策树相同；若令，则是随机选择一个属性用于划分；一般情况下，推荐值。

随机森林简单、容易实现、计算开销小、令人惊奇的是，它在很多现实任务中展现出强大的性能，被誉为“代表集成学习技术水平的方法”。可以看出，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差异度的增加而进一步提升。

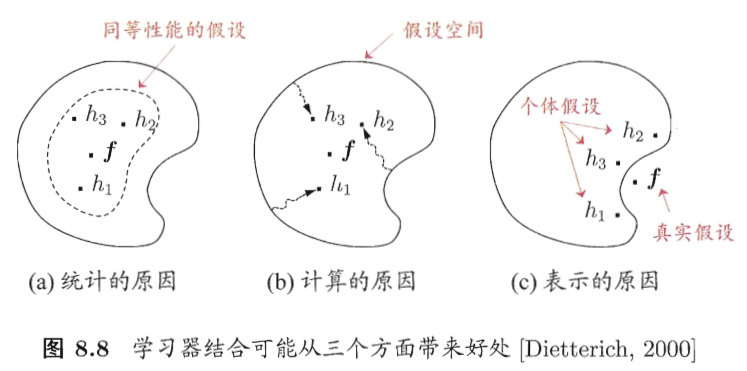
随机森林的收敛性与Bagging相似。随机森林的起始性能往往相对较差，特别是在继承中只包含一个基学习器时。这很容易理解，因为通过引入属性扰动，随机森林中个体学习器的性能往往会有所降低。然而，随着个体学习器数目的增加，随机森林通常会收敛到更低的泛化误差。值得一提的是，随机森林的训练效率常优于Bagging，因为在个体决策树的构建过程中，Bagging使用的是“确定型“决策树，在选择划分属性时要对结点的所有属性进行考察，而随机森林使用的”随机型“决策树则只考察一个属性子集。



4、结合策略

学习器结合可能会从三个方面带来好处：

1. 从统计的方面来看，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，此时若使用单学习器可能误选而导致泛化性能不佳，结合多个学习器则会较小这一风险；
2. 从计算的方面来看，学习算法往往会陷入局部极小，有的局部极小点所对应的泛化性能可能很糟糕，而通过多次运行之后进行结合，可降低陷入糟糕局部极小点的风险
3. 从表示的方面来看，某些学习任务的真实假设可能不在当前学习算法所考虑的假设空间中，此时若使用单学习器则肯定无效，而通过结合多个学习器，由于相应的假设空间有所扩大，有可能学的更好的近似。



4.1、平均法

对数值型输出，最常见的结合策略是使用平均法。

* 简单平均法
* 加权平均法

其中，是个体学习器的权重，通常要求，.

显然，简单平均法是加权平均法令的特例。加权平均法在1993年正式将其用于集成学习。它在集成学习中具有特别的意义，集成学习中的各种结合方法都可视为其特例或变体。事实上，加权平均法可认为是集成学习研究的基本出发点，对给定的基学习器，不同的集成学习方法可视为通过不同的方式来确定加权平均法中的基学习器权重。

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。尤其对规模比较大的集成来说，要学习的权重比较多，较容易导致过拟合。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；而在个体性能相近时，宜使用简单平均法。

4.2、投票法

对分类任务来说，学习器将从类别标记集合中预测出一个标记，最常见的结合策略是使用投票法。为便于讨论，我们将在样本x上的预测输出表示为一个N维向量，其中是在类别标记上的输出。

* 绝对多数投票法

即若某标记得票过半数，则预测为该标记；否则拒绝预测。

* 相对多数投票法

即预测为得票最多的标记，若同时又多个标记获得最高票，则从中随机选取一个。

* 加权投票法

与加权平均法类似，是的权重，通常要求，。

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“多数投票法”。

4.3、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。

5、多样性

啊