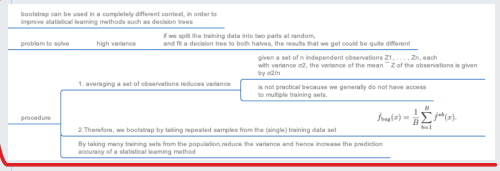
Ensemble learning II - Bagging

General Bagging

basic idea

Bagging [Breiman, 1996a] 是并行式集成学习方法最著名的代表. 从名字即可看出, 它直接基于我们在 2.2.3 节介绍过的自助采样法 (bootstrap sampling). 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. 由式(2.1)可知, 初始训练集中约有 63.2% 的样本出现在采样集中.



bias

search space is larger than decision tree whereas search strategy are aggregation of multiple decision tree

algorithm

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 8.5 所示.

即每个基学习器使用相同权重的投票、平均.

$\mathcal{D}_{bs}$ 是自助采样产生的样本分布.

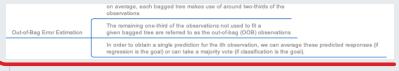
输入: 训练集 $D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_m, y_m)\}$;
基学习算法 $\mathfrak{L}$;
训练轮数 T.

过程:
1: **for** $t = 1, 2, \ldots, T$ **do**
2: $h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$
3: **end for**

输出: $H(\boldsymbol{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{I}(h_t(\boldsymbol{x}) = y)$

图 8.5 Bagging 算法

Natual out-of-bag validation set



Pros

Efficiency

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.

Reduce variance

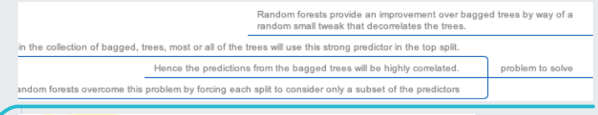
从偏差–方差分解的角度看, Bagging 主要关注降低方差, 因此它在不剪枝决策树、神经网络等易受样本扰动的学习器上效用更为明显. 我们以基于信息

Interpretation

Variable importance measures

Cons

Trees may be somehow correlated



random forest

Basic idea

随机森林(Random Forest, 简称 RF) [Breiman, 2001a] 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. 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 $k = \log_2 d$ [Breiman, 2001a].

Pros

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之间差异度的增加而进一步提升.

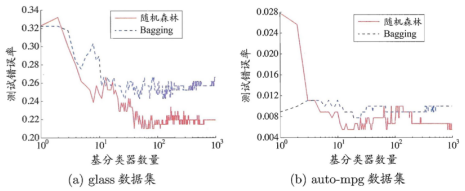
predictors. This may sound crazy, but it has a clever rationale. Suppose that there is one very strong predictor in the data set, along with a number of other moderately strong predictors. Then in the collection of bagged trees, most or all of the trees will use this strong predictor in the top split. Consequently, all of the bagged trees will look quite similar to each other.

why random

Using a small value of m in building a random forest will typically be helpful when we have a large number of correlated predictors. We applied

Demo

1.



(a) glass 数据集 (b) auto-mpg 数据集

图 8.7 在两个 UCI 数据上, 集成规模对随机森林与 Bagging 的影响

2.

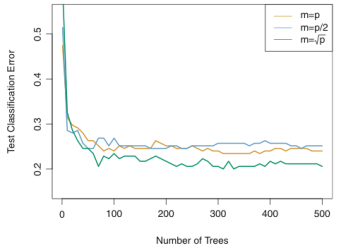


FIGURE 8.10. *Results from random forests for the 15-class gene expression data set with p = 500 predictors. The test error is displayed as a function of the number of trees. Each colored line corresponds to a different value of m, the number of predictors available for splitting at each interior tree node. Random forests (m < p) lead to a slight improvement over bagging (m = p). A single classification tree has an error rate of 45.7%.*