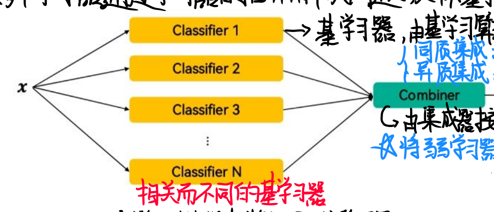
集成学习定义

通过训练多个学习器通过学习器的合并解决问题，又称基于委员会学习|多分类器系统



→ 基学习器，由基学习算法在训练数据得到（决策树，SVM，MLP等）
{ 同质集成：由一种基学习算法产生同质基学习器
{ 异质集成：多种学习算法训练出不同种类学习器

⤷ 由集成器按一定集成策略生成，使算法具有更强泛化能力

✿ 将弱学习器变成强学习器

采用加权平均可以得到比简单平均更好的泛化能力。但是，也有一些研究者认为，对权值进行优化将会导致过配(overfiting)，从而使得集成的泛化能力降低，因此，他们建议使用简单平均。

在分类问题上，集成的输出往往由个体学习器投票决定，投票法是一种很早开始就获得广泛使用的方法。(Random forest, Adaboost)

相关而不同的基学习器

参数、训练样本、特征、初始值不同

✿ 基学习器集成结果优于最优学习器与平均结果

{ 串行集成 ⇒ 串行生成基分类器 Boosting (Adaboost, GBDT, XGBoost)
{ 并行集成 ⇒ 并行生成基分类器 Bagging (Random Forest)

并行集成学习

利用基学习器之间的独立性，结合相互独立基分类器以显著减小方差

以一个二分类任务为例，输出分别为+1和-1，假设真实的函数为$f(x)$，训练T个基分类器$h_i(t)$，$i = 1,\ldots,T$。对于每一个基分类器，假设弱分类时的分类错误的概率$\epsilon < \frac{1}{2}$，存在：

$$P(h_i(x) \neq f(x)) = \epsilon$$

结合T个分类器，根据Voting的方法来决定最后的分类结果$H(x)$，用符号函数表示，可得

$$H(x) = sign(\sum_{i=1}^{T} h_i(x))$$

集成后的H之后在至少一半的基分类器预测错误时才会犯错。

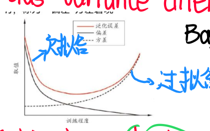
因此，根据霍夫丁不等式(Hoeffding)，集成后模型的泛化误差为（基学习器正确数小于T/2则出错）

$$P(H(x) \neq f(x)) = \sum_{k=0}^{[T/2]} \binom{T}{k} (1-\epsilon)^k \epsilon^{T-k} \leq \exp(-\frac{1}{2}T(2\epsilon - 1)^2)$$

集成器的规模越大，也就是T越大，集成模型的犯错的概率会显著性的呈指数下降，并且随着T趋向无穷大而趋于0。

✿ 基分类器由相同的训练数据产生并非完成独立同分布，因此很难得到完全独立的基分类器，但可通过随机方法减少他们之间的相关性，并通过集成实现更好的泛化能力

bias-variance dilemma



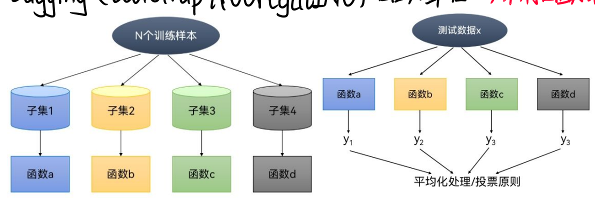
Bagging 可显著降低方差，使学习器更稳定

复杂模型方差大，平均化处理可有效降低

Bagging (Bootstrap AGGregatING) 自助聚合

⇒ 采用自助采样生成不同基分类器 独立性

✿ 生成不同训练样本同时又可服从类似分布



包外样本

m个样本集 i 每次被选中概率 $\frac{1}{m}$，选了m次，而 $\lim_{m\to\infty}(1-\frac{1}{m})^m = \frac{1}{e}$

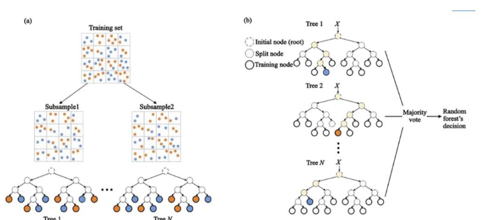
✿ 即对任一个基分类器，训练时原始训练数据中有约36.8%未使用 ⇒ 包外样本 ⇒ 验证基分类器性能

随机森林

在以决策树为基学习器构建Bagging集成基础上，进一步在决策树的训练的训练过程中引入随机属性

✓ 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。



➤ 测验： Bagging和Boosting有什么不同

1）样本选择上：
Bagging：训练集是在原始集中有放回选取的，从原始集中选出的各轮训练集之间是独立的。
Boosting：每一轮的训练集不变，只是训练集中每个样例在分类器中的权重发生变化。而权值是根据上一轮的分类结果进行调整。
2）样例权重：
Bagging：使用均匀取样，每个样例的权重相等
Boosting：根据错误率不断调整样例的权值，错误率越大则权重越大。
3）预测函数：
Bagging：所有预测函数的权重相等。
Boosting：每个弱分类器都有相应的权重，对于分类误差小的分类器会有更大的权重。
4）并行计算：
Bagging：各个预测函数可以并行生成。
Boosting：各个预测函数只能顺序生成，因为后一个模型参数需要前一轮模型的结果。

串行集成方法

Boosting 将弱学习器（错误率略小于50%的分类器）提升为强学习器（错误率可为0%）的算法

✿ 第一个分类器 $f_1(x)$ ⟶ 寻找 $f_2(x)$ 帮助 $f_1(x)$ 希望 $f_2(x)$ 与 $f_1(x)$ 互补 →…→ 组合所有分类器 ✿ 顺序学习

如何获得不同分类器 ⟶ 使用不同训练数据集

如何获得不同训练集

重新采样形成新集合 重新赋权改变样本分布形成新集合

$L(f) = \sum_n \mathbb{I}(f(x^n) \neq \hat{y}^n) \Rightarrow L(f) = \sum_n u^n \mathbb{I}(f(x^n) \neq \hat{y}^n)$

实际操作中，只要改代价/目标函数

Adaboost 自动调节权重

➤ 核心思想：在使得f1训练失败的训练集上训练f2
➤ 如何找到使f1训练失败的训练集？

ε1：f1在训练集上的错误率

$$\epsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_1} \qquad Z_1 = \sum_n u_1^n \qquad \epsilon_1 < 0.5$$

更新u_1^n到u_2^n样本的权重，使

$$\frac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_2} = 0.5$$

对于新的权重，f1的性能是随机的

基于新权重u_2^n，训练f2

$(x^1, \hat{y}^1, u^1)$ $u^1 = 1$ ✓	$u^1 = 1/\sqrt{3}$ ✓	
$(x^2, \hat{y}^2, u^2)$ $u^2 = 1$ ✗	$u^2 = \sqrt{3}$ ✗	
$(x^3, \hat{y}^3, u^3)$ $u^3 = 1$ ✓	$u^3 = 1/\sqrt{3}$ ✓	
$(x^4, \hat{y}^4, u^4)$ $u^4 = 1$ ✓	$u^4 = 1/\sqrt{3}$ ✓	

$\varepsilon_1 = 0.25$ → $f_1(x)$ → 0.5 → $\varepsilon_2 < 0.5$ → $f_2(x)$

{ x^n 被 f_1 错误分类 $u_2^n = u_1^n \times d_1$ 增强
{ x^n 被 f_1 正确分类 $u_2^n = u_1^n / d_1$ 减弱

⇒ 基于 u_2^n 权重训练 f_2 分类器

下求 d_1

新权重下的训练数据

d_1的值是什么？

$$\epsilon_1 = \frac{\sum_n u_1^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_1}$$

$$\frac{\sum_n u_2^n \delta(f_1(x^n) \neq \hat{y}^n)}{Z_2} = 0.5$$

$$= \sum_{f_1(x^n) \neq \hat{y}_n} u_1^n d_1$$

$Z_1 = \sum_n u_1^n$　$Z_2 = \sum_n u_2^n$

若$f_1(x^n) \neq \hat{y}_n$，则$u_2^n = u_1^n d_1$
若$f_1(x^n) = \hat{y}_n$，则$u_2^n = u_1^n / d_1$

$$\frac{\sum_{f_1(x^n) \neq \hat{y}_n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}_n} u_1^n / d_1}{\sum_{f_1(x^n) \neq \hat{y}_n} u_1^n d_1} = 2$$

$$Z_2 = \sum_{f_1(x^n) \neq \hat{y}_n} u_2^n + \sum_{f_1(x^n) = \hat{y}_n} u_2^n = \sum_{f_1(x^n) \neq \hat{y}_n} u_1^n d_1 + \sum_{f_1(x^n) = \hat{y}_n} u_1^n / d_1$$

$$\frac{\sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1}{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1} = 1$$

$$\sum_{f_1(x^n) = \hat{y}^n} u_1^n / d_1 = \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n d_1$$

$$\frac{1}{d_1} \underbrace{\sum_{f_1(x^n) = \hat{y}^n} u_1^n}_{Z_1(1-\epsilon_1)} = d_1 \underbrace{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n}_{Z_1 \epsilon_1}$$

错误率：$\epsilon_1 = \frac{\sum_{f_1(x^n) \neq \hat{y}^n} u_1^n}{Z_1}$　则$Z_1\epsilon_1 = \sum_{f_1(x^n) \neq \hat{y}^n} u_1^n$

$Z_1(1-\epsilon_1)/d_1 = Z_1 \epsilon_1 d_1$

$d_1 = \sqrt{(1-\epsilon_1)/\epsilon_1} > 1$

✿ Boosting 一般形式

Boosting的工作流程可写为如下更一般的情形

✓ 初始函数：$g_0(x) = 0$
✓ 从t=1到T：
　✓ 寻找一个函数$f_t(x)$和α_t来提高函数$g_{t-1}(x)$的性能
　　• $g_{t-1}(x) = \Sigma_{i=1}^{t-1} \alpha_i f_i(x)$
　✓ $g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$
✓ 输出：$H(x) = sign(g_T(x))$

➤ g(x)的学习目标：使如下损失函数最小

$$L(g) = \sum_n l(\hat{y}^n, g(x^n)) = \sum_n exp(-\hat{y}^n g(x^n))$$

算法流程

➤ 给定如下训练数据：

$\{(x^1, \hat{y}^1, u_1^1), \cdots, (x^n, \hat{y}^n, u_1^n), \cdots, (x^N, \hat{y}^N, u_1^N)\}$

$\hat{y} = \pm 1$（二分类），$u_1^n = 1$（相等权重）

• 对于t=1,⋯,T:
　• 用权重$\{u_t^1, \ldots, u_t^N\}$来训练弱分类器$f_t(x)$
　• ϵ_t为$f_t(x)$在权重$\{u_t^1, \ldots, u_t^N\}$上训练的错误率
　• 对于$n = 1, \ldots, N$:
　　• 如果x_n被$f_t(x)$错误分类：$u_{t+1}^n = u_t^n * d_t = u_t^n * exp(\alpha_t)$
　　• Else:
　　$u_{t+1}^n = u_t^n / d_t = u_t^n * exp(-\alpha_t)$
　　$u_{t+1}^n \leftarrow u_t^n \times exp(-\hat{y}^n f_t(x^n)\alpha_t)$

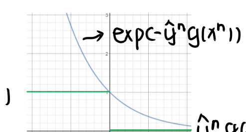
$d_t = \sqrt{(1-\epsilon_t)/\epsilon_t}$

$\alpha_t = ln\sqrt{(1-\epsilon_t)/\epsilon_t}$

整合得到的函数 $f_1(x), \cdots, f_t(x), \cdots, f_T(x)$ 误差 ε_t 越小投票权重越大

$H(x) = sign(\sum_{t=1}^{T} \alpha_t f_t(x))$

{ $\alpha_t = \ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$
{ $u_{t+1}^n = u_t^n \times exp(-\hat{y}^n f_t(x^n)\alpha_t)$



错误率讨论

最终分类器 $H(x) = sign(\underbrace{\sum_{t=1}^{T} \alpha_t f_t(x)}_{g(x)})$，$\alpha_t = \ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$

训练数据错误率：$\frac{1}{N}\sum_n \mathbb{I}(H(x^n) \neq \hat{y}_n) = \frac{1}{N}\sum_n \mathbb{I}(\hat{y}_n g(x^n) < 0) \leq \frac{1}{N}\sum_n exp(-\hat{y}^n g(x^n)) = \frac{1}{N} Z_{T+1}$

➤ Z_t：训练f_t的训练数据权重的总和，即：$Z_{T+1} = \sum_n u_{T+1}^n$

证明：

$$\left.\begin{aligned} u_1^n &= 1 \\ u_{t+1}^n &= u_t^n \times exp(-\hat{y}^n f_t(x^n)\alpha_t) \end{aligned}\right\} \quad u_{T+1}^n = \prod_{t=1}^{T} exp(-\hat{y}^n f_t(x^n)\alpha_t)$$

$$Z_{T+1} = \sum_n \prod_{t=1}^{T} exp(-\hat{y}^n f_t(x^n)\alpha_t) = \sum_n exp\left(-\hat{y}^n \underbrace{\sum_{t=1}^{T} f_t(x^n)\alpha_t}_{g(x)}\right)$$

$Z_1 = N$（初始权重都相同=1）

$Z_{t+1} = Z_t \epsilon_t e^{\alpha_t} + Z_t(1-\epsilon_t)e^{-\alpha_t}$

$= Z_t \epsilon_t \sqrt{(1-\epsilon_t)/\epsilon_t} + Z_t(1-\epsilon_t)\sqrt{\epsilon_t/(1-\epsilon_t)}$

$= Z_t \times 2\sqrt{(1-\epsilon_t)\epsilon_t}$

$Z_t \epsilon_t = \sum_{f_t(x^n) \neq \hat{y}^n} u_t^n$

$Z_t(1-\epsilon_t) = \sum_{f_t(x^n) = \hat{y}^n} u_t^n$

$$Z_{T+1} = N \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}$$

• 训练数据错误率 $\leq \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)}$

随着T的增加，变得越来越小

总结

优点：

• 模型简单容易操作，不用重新设计模型
• 分类错误率上界随着训练增加而稳定下降（理论可证）
• 在一定程度上不存在过拟合

缺点：

• α值是局部最优
• 对噪声比较敏感
• 模型的可解释性较差

Gradient Boosting 损失函数负梯度拟合残差

Boosting 目标：找到 $g(x)$ 使 $L(g) = \sum_n l(g(x^n), \hat{y}^n) = \sum_n exp(-\hat{y}^n g(x^n))$ 最小

若已有 $g(x) = g_{t-1}(x)$ 更新 $g(x)$ 以达成上述目标，梯度下降思想

$$g_t(x) = g_{t-1}(x) - \eta \boxed{\frac{\partial L(g)}{\partial g(x)}}\Big|_{g(x)=g_{t-1}(x)}$$

$-\sum_n exp(-\hat{y}^n g_{t-1}(x^n))\hat{y}^n$

＊boosting：$g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$

✿ $f_t(x)$ 与 $\sum_n exp(-\hat{y}^n g_t(x^n))(\hat{y}^n)$ 方向相同

希望有一个 $f_t(x)$ 使函数 $\sum_n \underbrace{exp(-\hat{y}^n g_{t-1}(x^n))}_{样本权重u_t^n} \boxed{(\hat{y}^n) f_t(x^n)}$ 最小化误差 同一符号

所以 $u_t^n = exp(-\hat{y}^n g_{t-1}(x^n)) = exp(-\hat{y}^n \sum_{i=1}^{t-1} \alpha_i f_i(x^n)) = \prod_{i=1}^{t-1} exp(-\hat{y}^n \alpha_i f_i(x^n))$

➤ 找到一个g(x)，使 $L(g) = \sum_n exp(-\hat{y}^n g(x^n))$

$g_t(x) = g_{t-1}(x) + \alpha_t f_t(x)$

α_t是一个类似于学习率的参数

固定$f_t(x)$找出使$L(g_t)$最小的α_t

$$L(g) = \sum_n exp(-\hat{y}^n(g_{t-1}(x) + \alpha_t f_t(x)))$$

$$= \sum_n exp(-\hat{y}^n g_{t-1}(x)) exp(-\hat{y}^n \alpha_t f_t(x))$$

$$= \sum_{\hat{y}^n \neq f_t(x)} exp(-\hat{y}^n g_{t-1}(x^n)) exp(\alpha_t) + \sum_{\hat{y}^n = f_t(x)} exp(-\hat{y}^n g_{t-1}(x^n)) exp(-\alpha_t)$$

找到一个α_t，使

$$\frac{\partial L(g)}{\partial \alpha_t} = 0$$

$\alpha_t = ln\sqrt{(1-\epsilon_t)/\epsilon_t}$

Adaboost!

明德 厚学 求

48

XGBoost 代价函数引入正则式