#任务一 随机森林算法梳理

04 曹轲烨

集成学习概念:

集成学习通过组合策略利用多个"个体学习器"来完成学习任务

集成方式:

- 同质集成: 集成中只包含同种类型的个体学习器(基学习器)
- 异质集成: 集成中包含不同类型的个体学习器(组件学习器)

集成学习的重要思想: 要获得好的集成效果,个体学习期要具有一定的准确性和多样性

集成学习的分类

- 个体学习器存在强依赖关系,必须串行 **Boost**
- 个体学习器可同时生成并行化 **Random Forest**

##Boosting

算法总体流程:

1. 初始化训练数据（N个样本）的权值分布：每一个训练的样本点被赋权重：1/N
2. 训练弱分类器.如果某个样本已经被准确地分类，那么在构造下一个训练集中，它的权重就被降低；相反，如果某个样本点没有被准确地分类，那么它的权重就得到提高。然后，更新权值后的样本集被用于训练下一个分类器，整个训练过程如此迭代地进行下去
3. 将各个训练得到的弱分类器组合成强分类器。各个弱分类器的训练过程结束后，分类误差率小 的弱分类器的话语权较大，其在最终的分类函数中起着较大的决定作用，而分类误差率大的弱分类器的话语权较小，其在最终的分类函数中起着较小的决定作用。

第一步：初始化权值

样本 $T=\{(x_1,y_1)\cdots(x_N,y_N)\}$，对每个样本赋予同样的权值 $w_i=1/N$

$D_1=(w_{11},w_{12},\cdots,w_{1N})$，$w_{1i}=1/N$，$i=1,2\cdots,N$

↳ 第一次迭代每个样本的权值

第二步：迭代训练多个基分类器 $G_m(x)$

使用具有权值分布 D_m 的样本进行学习，得到 $G_m(x)=\begin{cases}1\\-1\end{cases}$

过程

① 基于初始参数，通过分类器得到结果 $y_m(x)$

② 根据误差函数计算误差（统计分错样本对应数据的权值）

根据：$\epsilon_m=\sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n)\neq t_n)$

③ 根据 ϵ_m 计算当前分类器在最终分类器中的权值 α_m

$\alpha_m=\frac{1}{2}\ln\frac{1-\epsilon_m}{\epsilon_m}$ → 当 ϵ_m↓，α_m↑，误差越小的分类器在最终分类器的重要程度越大

④ 更新训练样本的权值分布 →
- ① 被误分的样本的权值增大
- ② 正确分的权值下降

$D_{m+1}=(w_{m+1,1},w_{m+1,2},\cdots,w_{m+1,N})$

$w_{m+1}=\frac{w_m}{z_m}\exp(-\alpha_m y_i G_m(x_i))$ →
- ① 若分对 $y_i G_m(x_i)=1$
- ② 若分错 $y_i G_m(x_i)=-1$

其中 z_m 是归一化参数

$z_m=\sum_{i=1}^{N} w_m\cdot\exp(-\alpha_m y_i G_m(x_i))$

第三步：多个分类器训练完后进行组合最终分类器 $G(x)$

$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$

$G(x)=\mathrm{sign}(f(x))$

##Bagging bagging是并行式集成学习,采用自助采样法, 通过对数据集进行采样,利用不同的数据子集可训练出具有差异性的基学习器.

数据生成**:** 给定包含m个样本的数据集,随机采样一个样本放到采样集中,在将样本放回数据集,经过n次随机采样最后得到包含n个样本的*数据子集* 基学习器学习**:** 通过自助采样得到T个数据子集,基于每个数据子集训练出一个基学习器 组合基学习器**:** 对分类任务进行简单投票 对回归任务使用随机选择

##随机森林(RF) 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随机属性选择

具体执行**:** 在RF中对决策树的每个节点,先从该节点的属性集合(d个)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. 当k = d,就退化成了传统决策树 当k = 1,则退化成了随机选择属性 推荐 $k=log_2d$

##组合策略

(一)平均法**:**

- 简单平均法 $$H(x)=\frac{1}{T}\sum_{i=1}^Th_i(x)$$
- 加权平均法: $$H(x)=\sum*{t=1}^Tw_ih_i(x)$$ *其中$w_i>0,*\sum{i=1}^Tw_i = 1$

(二)投票法**:**

对于分类问题h_i将在标记集合${{c_1,c_2,...,c_N}}$预测一个标签,假设h_i在样本x上的预测为N维度向量$(h_i^1(x),(h_i^2(x)...(h_i^N(x))$其中$h_i^j(x)$表示$h_i$在类别标记$c_j$上的输出

- 绝对数投票法

$$ H(x)=\left{ \begin{array}{rcl} c_j & & {\sum*{i=1}^Th_i^j(x)>0.5\sum*{k=1}^N\sum_{i=1}^Th_i^k(x)}\ reject & & {otherwise}\ \end{array} \right. $$ 若标记得票过半,就预测为该标记否则拒绝.

- 相对多数投票法 $$H(x)=C*{arg}max\sum*{i=1}^Th_i^j(x)$$ 预测为得票数最多的标记,若同时有多个标记获得最高票,则从中随机选取. $~$
- 加权投票法 $$H(x)=C*{arg}max\sum*{i=1}^Tw_ih_i^j(x)$$ 其中$w_i>0,\sum_{i=1}^Tw_i = 1$

在实际任务中,不同类型个体学习器会产生不同类型的$h_i^j(x)$值如*类标记*和*类概率*

- 类标记: 硬投票 若h_i预测x正确,$h_i^j(x) = 1$,否则$h_i^j(x) = 0$
- 类概率: 软投票 每一个预测是对后验概率$P{(c_j|x)}$的估计

(三)学习法**:**

个体学习器 -> 初级学习器 用于结合的学习器 -> 次级学习器

- Strcking Stacking的主要思想是训练几个简单的次级学习器,将它们进行K折交叉验证输出预测结果,然后将每个模型输出的预测结果合并为新的特征来训练初级学习器

图片:thttps://blog.csdn.net/qq_18916311/article/details/78557722

如图所示

选用初级学习期${{M_1,...,M_n}}$ 对每一个初级学习期进行5折交叉验证,各得到 ${P*{itrain1}...P*{itrain5}}$ 拼成 MT_i作为次级学习器的一部分训练数据 ${P*{itest1},,P*{itest5}}$ 平均得到P_{imean}

最终将${MT_1...MT_n}$拼成次级学习器的数据集 最终将${P*{1mean}...P*{nmean}}$拼成次级学习器的测试集 最后可利用LR训练