§8 集成学习

1. 个体与集成

弱分类器(weak learner)：分类准确率只稍微好于随机猜测的分类器

将若干个学习器组合后产生一个新学习器。具有泛化性能、准确性

$H(x)=\operatorname{sign}\left(\sum_{t=1}^{T} h_t(x)\right)$

- boosting：个体学习器间存在强依赖关系，必须串行生成序列化方法
- bagging, random forest：个体学习器间不存在强依赖关系，可同时生成并行化方法

为什么需要ensemble learning?

(1)弱分类器间存在一定差异性，导致错误率不同。若多个弱分类器合并后，可得到合成错误率的整体降低率。

(2)对于大/小数据集，可分别进行划分和有放回操作产生不同的数据子集

然后根据子集训练不同的分类器，最后合并成一个大分类器
(3) 若数据量过于庞大，可训练多个模型，再进行模型融合
(4) 对于多个异构特征集的进行融合，可考虑每个数据集构建一个分类模型，然后将多个模型融合。

2. Boosting

原理：先从初始训练集训练一个基学习器，再根据表现进行调整，使做错的训练样本多关注，然后基于调整后的样本分布来训练下一个基学习器，直至基学习器数目达到指定值T，将T进行加权结合。

根据损失函数的梯度——梯度提升 (Gradient boosting)

Adaboost：越错误预测，样本权重升高，使迭代过程直至error rate 小或达到一定的迭代次数。

$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$　　$G(x)=\text{sign}(f(x))=\text{sign}\left[\sum_{m=1}^{M}\alpha_m G_m(x)\right]$

损失函数 $loss=\frac{1}{n}\sum_{i=1}^{n}I(G(x_i)\neq y_i)\leq\frac{1}{n}\sum_{i=1}^{n}e^{-y_i f(x)}$

Adaboost过程：

(1) 假设训练data $T=\{(x_1,y_1),(x_2,y_2)\dots(x_n,y_n)\}$

(2) 初始化训练data 权重分布 $D_1=(W_{11},W_{12}\dots W_{1n})$

(3) 用具有权值分布 D_m 的训练data，基本分类器：$G_m(X): X\to\{-1,+1\}$

(4) 计算 $G_m(X)$ 在训练集上的分类误差 $\varepsilon_m=P(G_m(x_i)\neq y_i)=\sum_{i=1}^{n}W_{mi}\cdot I(G_m(x_i)\neq y_i)$

(5) 计算 $G_m(X)$ 模型的权重系数 α_m：$\alpha_m=\frac{1}{2}\log_2\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)$

(6) 权重训练data的权值分布：$D_{m+1}=(W_{m+1,1}\dots)$　$W_{m+1,i}=\frac{W_{mi}}{Z_m}e^{-\alpha_m y_i G_m(x_i)}$

(7) Z_m 规范化因子 $Z_m=\sum_{i=1}^{n}W_{mi}e^{-\alpha_m y_i G_m(x_i)}$

(8) 构建基本分类器的线性组合．　$f(x)=\sum_{m=1}^{M}\alpha_m G_m(x)$

(9) 最终分类器 $G(x)=\text{sign}(f(X))=\text{sign}\left(\sum_{m=1}^{M}\alpha_m G_m(X)\right)$

Adaboost 只做二分类

3. bagging. random forest

bagging: 可用于分类、回归。自助采样 (out-of-bag estimate)，有放回抽样

bagging降低方差 bias-variance

RF: bagging + 随机属性选择。

RF步骤: (1) n棵中用bootstrap有放回重采样选出n个样本。

(2) 使用子数据集训练决策树，从所有属性中取k个，再选择最佳分割属性作为节点来迭代创建决策树

(3) 重复m次，建立m棵决策树 (4) 通过投票表决决定属于哪一类

RF优点: 并行化速度快，随机划分特征训练效果高。给出各特征重要性

引表，时随机抽样，模型方差小泛化能力强，实现简单，对部分特征缺失不敏感

缺点: 噪声数据容易过拟合，取值比较多划分特征对RF决策影响大

4 结合策略

(1) 平均法 简单、加权平均法

(2) 投票法 绝对多数(过半)，相对多数(得票最多标记，若同时多个标记获最高票，从中随机取一个)，加权。

(3) 学习法。stacking。BMA(Bayes Model Averaging)

5 GBDT

要求的学习器必须是回归CART模型，要求每轮迭代样本损失尽可能小，方向为负梯度

GBDT: DT(Regression Decision Tree)，GB(Gradient Boosting)，Shrinkage(衰减)

Loss function: $L(y, F(x)) = \frac{1}{2}(y - F(x))^2$ $L(y, F(x)) = |y - F(x)|$

最优解: $F^*(x) = \arg\min_F L(y, F(x))$

GBDT 回归与分类算法区别：

回归

平方差损失函数　绝对误差损失函数

$L(y, F_m(x)) = \frac{1}{2}(y - F_m(x))^2$　$L(y, F_m(x)) = |y - F_m(x)|$　损失函数

$\alpha_{im} = y_i - F_{m-1}(x)$　$\alpha_{im} = sign|y_i - F_{m-1}(x)|$　负梯度

均值　中值　初始值

分类

对数损失函数（二分类）　对数损失函数（多分类）　损失函数 $P_k(x) = \frac{\exp(f_k(x))}{\sum_{l=1} \exp(f_l(x))}$

$L(y, F_m(x)) = -\{y \ln(p_m) + (1-y)\ln(1-p_m)\}$　$L(y, F_m(x)) = -\sum_{k=1}^{k} y_k \ln p_k(x)$

$p_m = \frac{1}{1+e^{-F_m(x)}}$

$\alpha_{im} = y_i - p_m$　$\alpha_{iml} = y_{il} - p_{ml}(x)$　负梯度

ln（正样本数／负样本数）　0　初始值

GBDT 优点：

可处理离散／连续值．少调参也效果不错 robust

缺点：

弱学习器之间存在关联关系．难以并行训练．速度慢

bagging．boosting 区别

	bagging	boosting
样本选择	有放回随机采样	训练集不变．样例权重／y 变化．根据上一轮调整
样例权重	随机均等权重	根据错误率不断调整权重．error ↑ 权重 ↑
预测函数	所有权重相等	误差小的分类权重更大
并行计算	并行生成各基模型	顺序生成．需上一个结果
bias-var	↓var 方差	↓bias 偏度
	需降低过拟合、强学习	弱分类．欠拟合