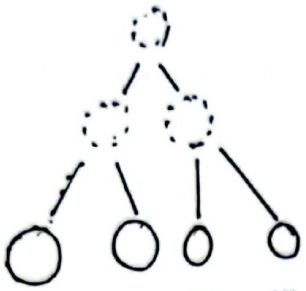
回归树



w_1 w_2 w_3 w_4 节点值

I_1 I_2 I_3 I_4 节点样本集合

$I_j = \{i \mid q(x_i) = j\}$ 节点样本集合和样本关系

$W_{q(x_i)} = T(\theta; x_i)$ x_i 对应的值.

模型表达

$$y_i^{(t)} = \sum_t^T f_t(x_i) = \underbrace{\sum_t^{T-1} f_t(x_i)}_{y_i^{(T-1)}} + f_T(x_i)$$

前向分布算法：贪心，逐个优化. 5.1

$$y_i^{(t)} = y_i^{(t-1)} + f_t(x_i)$$

$$= y_i^{(t-1)} + \underbrace{W_{q(x_i)}^t}$$

目标函数.

$$\sum_{i=1}^N L(y_i, \hat{y}_i^{(t)}) + \sum_t \Omega(f_t) = obj^{(t)}$$

优化目标.

$$(w_1, w_2, \cdots w_m) = \arg\min \sum_{i=1}^N (y_i, \hat{y}_i^{(t)}) + \sum_t^T \Omega(f_t).$$

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^T w_j^2 \quad T: \text{叶子节点个数}.$$

original by Alkaid

目标函数

$$\sum_{i=1}^{N} L(y_i, \hat{y}_i^{(t)}) + \sum_{j=1}^{t} \Omega(f_j)$$

非连续.

$$= \sum_{j=1}^{t-1} \Omega(f_j) + \Omega(f_t)$$

$$= \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

$$\sum_{i=1}^{N} L(y_i, \hat{y}_i^{(t)})$$

$$= \sum_{j=1}^{T} \sum_{i \in I_j} L(y_i, \hat{y}_i^{(t)})$$

$\therefore$ 目标函数 $\sum_{j=1}^{T} \left[\sum_{i \in I_j} L(y_i, \hat{y}_i^{(t)}) + \frac{1}{2}\lambda w_j^2 \right] + \gamma T$

$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + w_j$

x_0 $x - x_0$

$\therefore L(y_i, \hat{y}_i^{(t)}) \approx L(y_i, \hat{y}_i^{(t-1)}) + L'(y_i, \hat{y}_i^{(t-1)}) w_j + \frac{1}{2} L''(y_i, \hat{y}_i^{(t-1)}) w_j^2$

$\therefore obj^{(t)} = \gamma T + \sum_{j=1}^{T} \frac{1}{2}\lambda w_j^2 + \sum_{j=1}^{T} \sum_{i \in I_j} L'(y_i, \hat{y}_i^{(t-1)}) w_j + \frac{1}{2} L''(y_i, \hat{y}_i^{(t-1)}) w_j^2$.

$$= \gamma T + \sum_{j=1}^{T} \left[w_j \cdot \underbrace{\sum_{i \in I_j} L'(y_i, \hat{y}_i^{(t-1)})}_{g_i} + \frac{1}{2} w_j^2 \cdot \underbrace{\sum_{i \in I_j} L''(y_i, \hat{y}_i^{(t-1)})}_{h_i} + \frac{1}{2}\lambda w_j^2 \right]$$

G_j H_j

$$= \gamma T + \sum_{j=1}^{T} w_j G_j + \frac{1}{2} w_j^2 (\lambda + H_j)$$

original by Alkaid

$\therefore (w_1, w_2, \cdots, w_j) = \min obj^{(t)}$

$w_1^* = \arg\min (w_1 G_1 + \frac{1}{2} w_1^2 H_1)$

$w_2^* = \arg\min (w_2 G_2 + \frac{1}{2} w_2^2 H_2)$

$\therefore w_j^* = -\frac{G_j}{H_j + \lambda}$

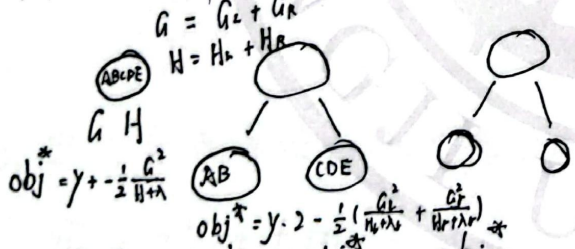
代入目标函数.

$obj^{(t)} = \gamma T + (-\frac{1}{2}) \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda}$

确定树的结构.

obj^*

精确贪心算法



增益 $gain = obj_{后}^{*} - obj_{前}^{*}$.

$$= \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma.$$

1. if $gain < 0$, 或 $len(l_j) = 1$,
   stop.
2. or if $gain <$ threshold,
   stop.
3. max_depth or T.

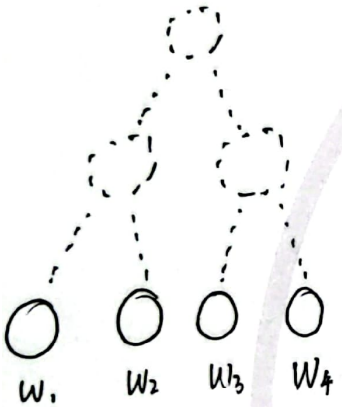
original by Alkaid

XGBoost 梯度提升回归树

集成算法，加法模型：

由许多棵回归树共同进行预测。

1. 单棵回归树



2. 模型表达

$$y_i^{(T)} = \underbrace{\sum_t^{T-1} f_t(x_i)}_{y_i^{(T-1)}} + f_T(x_i) = \sum_t^{T} f_t(x_i)$$

T棵树预测值之和为最终

3. 前向分布算法：逐个优化，优化第 t 棵树时，假定前 t-1 棵已

$$y_i^{(t)} = y_i^{(t-1)} + f_t(x_i)$$

4. 目标函数

$$\sum_{i=1}^{N} L(y_i, y_i^{(t)}) + \sum_t^T \Omega(f_t) = obj^{(t)}$$

XGBoost.

回归树. CART树. 二叉.

使用方式 sklearn 中 API. 或者 xgboost

目前先学 sklearn 中的 API.

XGBoost 三大版块.

参数　集成算法　弱评估器　其他过程

集成算法: 多个弱评估器

装袋法: bagging. 每棵树之间(弱评估器)互不干扰, 最后求均值.

提升法: boosting. 逐次迭代. 1->2->3. 基于前面的树的结果生成新的树, 不断有很多棵这样的树

梯度提升回归树. GBDT

单颗树中每个叶子节点都包含一些样本;
待预测的样本在这颗树中最后会落到某一个叶子节点上,
该样本的预测值即为该叶子节点的样本均值.

输入 x_i, 在第 k 棵树上的结果表示为 $h_k(x_i)$, γ_k 表示树在森林中的权重

则 $\hat{y}_i = \sum_k \gamma_k h_k(x_i)$, $h_k(x_i)$ 为该叶子上所有样本的均值.

但 XGB 作为 GBDT 改进, 有所不同.

XGB 中每颗树上每个叶子节点上的值不再是均值, 而是以更复杂计算后的回归取值, 称为叶子权重. 记为 f_k. ∴ 输入 x_i, 第 K 棵树得分. 共计 K 棵树. ∴ $\sum_K f_k(x_i) = \hat{y}_i^{(k)}$

∴在 XGB 中, $\hat{y}_i^{(k)} = \sum_k^K f_k(x_i)$.

K表示集成算法中生成树的数目, 是一个超参数.

sk: n_estimators 默认=10. 表示评估器数量. …… ①
xgb: num_round 默认 10.

sk: silent default: True
xgb: silent de: False 是否打印每次训练结果 …… ②
可监控建模流程.

W_j 第 j 个节点的值预测

I_j 第 j 个节点包含的样本的集合.

x_i 第 i 个样本

$q(x_i)$ 第 i 个样本落入的节点的序号

$W_{q(x_i)}$ 第 i 个样本的预测值.

T 回归树的总数

$y_i^{(T)}$ 模型对第 i 个样本的预测值

$y_i^{(t)}$ 包含 t 棵树的模型对第 i 个样本的预测值

$f_t(x_i)$ 第 t 棵树对第 i 个样本的预测值.

N 样本总数

$J_{\&}(y_i, y_i^{(t)})$ 包含 t 棵树的模型对 i 样本的预测值与真实值误差. $J_{\&}(y_i, y_i^{(t)}) = (y_i - y_i^{(t)})^2$.

$\Omega(f_t)$ 第 t 棵树的复杂度