GBDT

GBDT
- 决策树cart算法
  - 回归树
  - 分类树
- 提升树

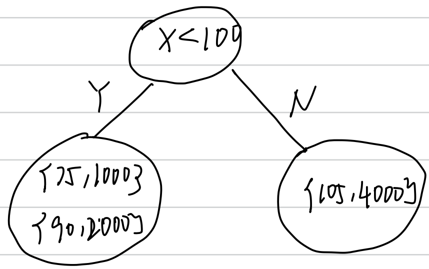
1.回归树

1.1. $f(x)=\sum_{m=1}^{M} C_m I(x \in R_m)$

m：叶子节点中的序号，第m个叶子节点、
Cm：每个叶子节点对应的树

x：房屋面积
y：租金

$D=\{(75,1000),(90,2000),(105,4000)\}$



$R_1=\{75,90\}$　　$R_2=\{105\}$
C_1　　C_2

注：每个叶子节点，都会输出一个预测值
预测值一般是该叶子所含训练样本在该点上输出的均值

1.2. 如何构建树：

(1) 树的深度：
- 确定叶子节点个数 or 树的深度
- 子节点包含样本数
- 给定精度 (loss)

(2) 划分节点、如何选　　找到损失最小的树的划分情况

(3) 叶子节点代表的Cm如何定　　Cm为平均值（每个叶子节点）

划分条件 — Cm — 损失

1.3、损失函数

$$\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2$$ 按样本方式遍历（同样也可以按节点）

按样本遍历

按节点遍历

优化求解

$$J=\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2$$

$$=\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{m=1}^{M}C_m I(x_i\in R_m)-y_i\right)^2$$

$$=\underbrace{\frac{1}{n}\sum_{m=1}^{M}\sum_{x_i\in R_m}(C_m-y_i)^2}$$ 对于第m个节点，遍历其中所有的样本

$$C_m^*=\min_{C_m}\frac{1}{n}\sum_{m=1}^{M}\sum_{x_i\in R_m}(C_m-y_i)^2$$ （求导令导数=0，从而求解 C_m^*）

$$\frac{\partial J}{\partial C_m}=\frac{\partial \frac{1}{n}\sum_{m=1}^{M}\sum_{x_i\in R_m}(C_m-y_i)^2}{\partial C_m}=\frac{\partial \sum_{x_i\in R_m}(C_m-y_i)^2}{\partial C_m}$$ （C_m确定后，$\sum_{m=1}^{M}$无）

$$=2\sum_{x_i\in R_m}(C_m-y_i)$$

设叶子节点 R_m 包含样本 N_m 个

$$=N_mC_m-\sum_{x_i\in R_m}y_i=0$$

$$C_m=\frac{\sum_{x_i\in R_m}y_i}{N_m}$$ => 每个叶子节点的含有样本的均值

即当每个叶子节点的 C_m 取值为该节点所有样本 y_i 的均值时，损失最小

2、分类树

· 二叉树，使用基尼指数作为损失函数

基尼指数：（K类，样本点属于第k类的概率为 p_k）

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2$$

对于二分类问题，样本点属于第1类的概率为p，概率分布的基尼指数为

$$Gini(p)=2p(1-p)$$

对于给定样本集合D

$$Gini(D)=1-\sum_{k=1}^{K}\left(\frac{|C_k|}{|D|}\right)^2$$

Gini 很小 —— 样本基本都属于同一类别

$$Gini(D,A)=\frac{|D_1|}{|D|}Gini(D_1)+Gini(D_2)\frac{|D_2|}{|D|}$$

回归树、分类树本质都是构建二叉树；计算损失不同 $\begin{cases}\text{回归 — MSE}\\ \text{分类 — 基尼系数}\end{cases}$

3. GBDT：提升树被认为是统计学习中性能最好的方法之一

$$f_M(x)=\sum_{m=1}^{M}T(x;\Theta_m)$$

M为树的个数

Θ_m：决策树的参数

$T(x;\Theta_m)$ 决策树

模型：加法模型

损失函数：
- 回归问题 —— MSE
- 分类问题
  - 二分类　　指数损失
  - 多分类　　softmax
- 一般决策：自定义损失函数

优化方法：前向分步算法

提升树算法采用前向分步算法。首先确定初始提升树 $f_0(x)=0$，第 m 步的模型是

$$f_m(x)=f_{m-1}(x)+T(x;\Theta_m) \tag{8.25}$$

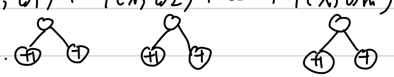
其中，$f_{m-1}(x)$ 为当前模型，通过经验风险极小化确定下一棵决策树的参数 Θ_m：

$$\hat{\Theta}_m=\arg\min_{\Theta_m}\sum_{i=1}^{N}L(y_i,f_{m-1}(x_i)+T(x_i;\Theta_m)) \tag{8.26}$$

3.1. 二分类问题的提升树

相当于把Adaboost中的G(x)换成二叉树

$$f(x)=T(x;\Theta_1)+T(x;\Theta_2)+\cdots+T(x;\Theta_m)$$



相当于Adaboost的特殊情况
① 基分类器G(x)限制为二分类树
② 权重α_m置为1

使用指数损失函数更新权重 $e^{-yf(x)}$

3.2. 回归问题的提升树

损失函数：平方误差根 $L(y,f(x))=(y-f(x))^2$

$$=[y-f_{m-1}(x)-T(x;\Theta_m)]^2$$

$$=[r-T(x;\Theta_m)]^2$$

(残差 = 真实值 − 预测值)

前向分步算法：利用残差数据构建训练样本，拟合残差后新样本

$$\hat{\theta}_m = \arg\min_{\theta_m} \sum_{i=1}^{N} L\left(y^{(i)}, f_{m-1}(x^{(i)}) + T(x^{(i)};\theta_m)\right)^2$$

$$= \arg\min_{\theta_m} \sum_{i=1}^{N} \left(r_m^{(i)} - T(x^{(i)};\theta_m)\right)^2$$

用残差训练当前学习器

3.3. GBDT (梯度提升树)

分类 — 指数损失f　　回归 — MSE — 残差　　通解?

目标: 每增加一棵, 损失减小

$$L(y^{(i)}, f_m(x^{(i)})) < L(y^{(i)}, f_{m-1}(x^{(i)}))$$

即 $L(y^{(i)}, f_m(x^{(i)})) - L(y^{(i)}, f_{m-1}(x^{(i)})) > 0$ 满足此式说明损失越来越小

一阶泰勒: $f(x) = f(x_0) + f'(x_0)(x - x_0)$

$L(y, f_m(x))$ 中只有 $f_m(x)$ 是未知量

$$L(y, f_m(x)) \approx L(y, f_{m-1}(x)) + \frac{\partial L(y, f_m(x))}{\partial f_m(x)}\bigg|_{f_m(x)=f_{m-1}(x)} \left(f_m(x) - f_{m-1}(x)\right)$$

$$\approx L(y, f_{m-1}(x)) + \frac{\partial L(y, f_m(x))}{\partial f_m(x)}\bigg|_{f_m(x)=f_{m-1}(x)} \cdot T(x;\theta_m)$$

即: $L(y, f_{m-1}(x)) - L(y, f_m(x)) \approx -\frac{\partial L(y, f_m(x))}{\partial f_m(x)}\bigg|_{f_m(x)=f_{m-1}(x)} T(x;\theta_m)$

当 $T(x;\theta_m) \approx -\frac{\partial L(y, f_m(x))}{\partial f_m(x)}\bigg|_{f_m(x)=f_{m-1}(x)}$ 时

$$L(y, f_{m-1}(x)) - L(y, f_m(x)) \geq 0$$

$$r_m(x, y) = -\left[\frac{\partial L(y, f_m(x))}{\partial f_m(x)}\right]_{f_m(x)=f_{m-1}(x)}$$

将 (x_i, y_i) 代入 $r_m(x, y)$, 即可得 r_{mi}

进而得到第m轮的训练数据集

$$T_m = \{(x_1, r_{m1}), (x_2, r_{m2}) \ldots, (x_N, r_{mN})\}$$

★梯度提升: 1. 计算当前损失函数的负梯度表达式 $r_m(x, y) = -\left[\frac{\partial L(y, f_m(x))}{\partial f_m(x)}\right]_{f_m(x)=f_{m-1}(x)}$

2. 构建新的训练样本 (第m轮的训练数据集)

算法流程:

算法 8.4（梯度提升算法）

输入：训练数据集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$，$x_i\in\mathcal{X}\subseteq\mathbf{R}^n$，$y_i\in\mathcal{Y}\subseteq\mathbf{R}$；损失函数 $L(y,f(x))$；

输出：回归树 $\hat{f}(x)$。

（1）初始化

$$f_0(x)=\arg\min_c\sum_{i=1}^N L(y_i,c)$$

常数

e.g.

$$L=\sum_{i=1}^N (y_i-c)^2$$

$$\downarrow$$

$$\sum_{i=1}^N -2c(y_i-c)$$

$$=\sum_{i=1}^N 2c^2-2c\sum_{i=1}^N y_i$$

$$=2Nc^2-2c\sum_{i=1}^N y_i$$

$$Nc^2-c\sum_{i=1}^N y_i=0$$

$$\Rightarrow c\ ..\ ?$$

（2）对 $m=1,2,\cdots,M$

（a）对 $i=1,2,\cdots,N$，计算

$$r_{mi}=-\left[\frac{\partial L(y_i,f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{m-1}(x)}$$

（b）对 r_{mi} 拟合一个回归树，得到第 m 棵树的叶结点区域 R_{mj}，$j=1,2,\cdots,J$。

（c）对 $j=1,2,\cdots,J$，计算

$$c_{mj}=\arg\min_c\sum_{x_i\in R_{mj}}L(y_i,f_{m-1}(x_i)+c)$$

（d）更新 $f_m(x)=f_{m-1}(x)+\sum_{j=1}^J c_{mj}I(x\in R_{mj})$

（3）得到回归树

$$\hat{f}(x)=f_M(x)=\sum_{m=1}^M\sum_{j=1}^J c_{mj}I(x\in R_{mj})\quad\blacksquare$$