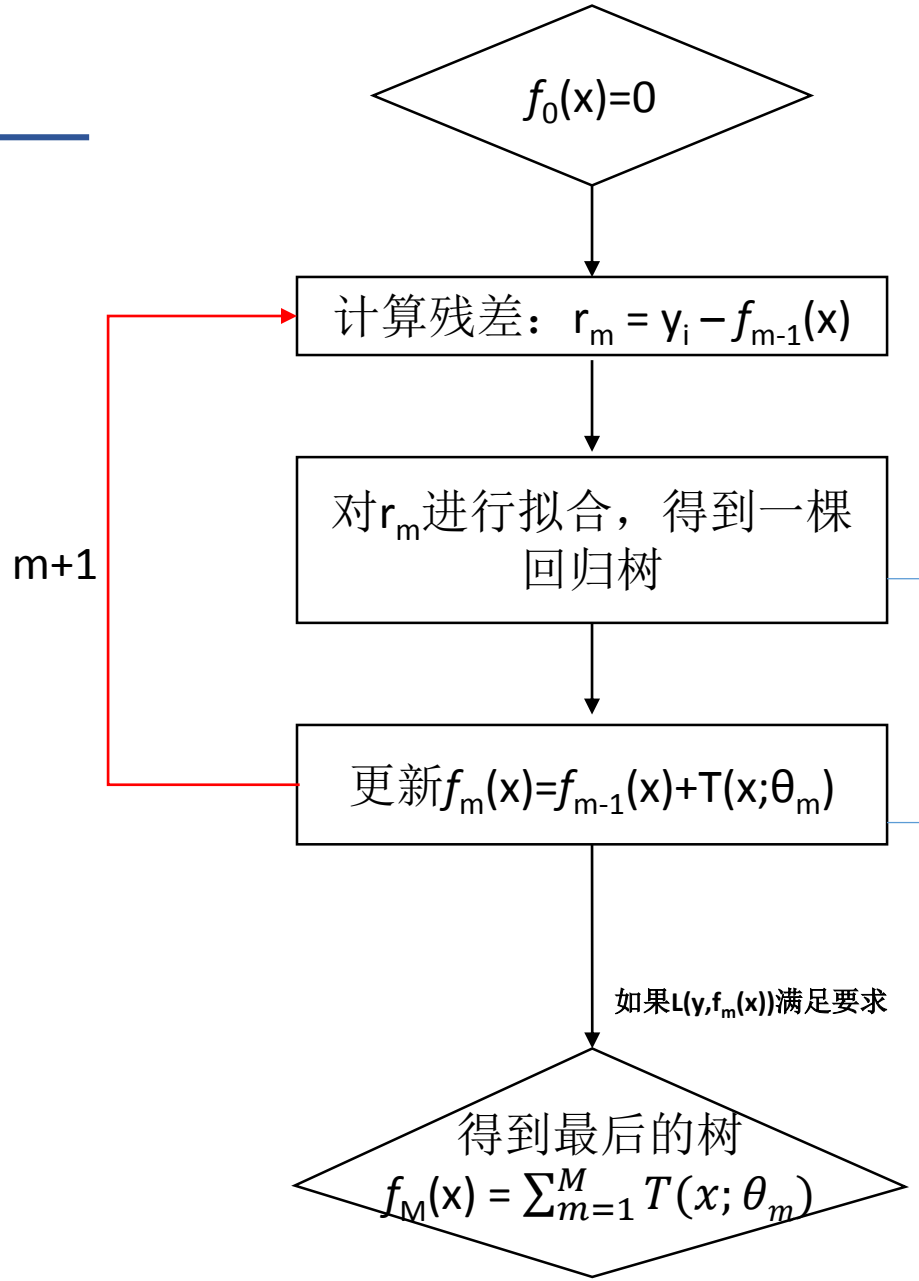
从提升到xgboost

jhs
2018/09/04

提升术



为什么是拟合残差？

新树如何相加？

为什么是拟合残差？

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

$$\hat{\theta}m = \underset{\theta m}{\operatorname{argmin}} \sum_{i=1}^{N} L(yi, fm_{\_1}(x_i) + T(x_{i\,;}\theta m))$$

$$L(y , f_m(x))=(y - f_m(x))^2$$
$$=L[y , f_{m-1}(x) + T(x;\theta)]^2$$
$$=[y - f_{m-1}(x) - T(x;\theta)]^2$$

故，损失函数写为：

$$L [r_{m-1} , T(x ; \theta)]^2$$

r_{m-1} 的值，用于第m轮的拟合

树们如何相加？

$$T_1(x) = \begin{cases} 6.24\ , & x < 6.5 \\ 8.91\ , & x \geq 6.5 \end{cases} \qquad f_1 = f_0 + T_1(x;\theta) \qquad f_1(x) = \begin{cases} 6.24\ , & x < 6.5 \\ 8.91\ , & x \geq 6.5 \end{cases}$$

$$T_2(x) = \begin{cases} -0.52\ , & x < 3.5 \\ 0.22\ , & x \geq 3.5 \end{cases} \qquad f_2 = f_1 + T_2(x;\theta) \qquad f_2(x) = \begin{cases} 5.72\ , & x < 3.5 \\ 6.46\ , & 3.5 \leq 6.46 < 6.5 \\ 9.13\ , & x \geq 6.5 \end{cases}$$

$$T_3(x) = \begin{cases} 0.15\ , & x < 6.5 \\ -0.22\ , & x \geq 6.5 \end{cases} \qquad T_4(x) = \begin{cases} -0.16\ , & x < 4.5 \\ 0.11\ , & x \geq 4.5 \end{cases} \qquad T_5(x) = \begin{cases} 0.07\ , & x < 6.5 \\ -0.11\ , & x \geq 6.5 \end{cases} \qquad T_6(x) = \begin{cases} -0.15\ , & x < 2.5 \\ 0.04\ , & x \geq 2.5 \end{cases}$$

树们如何相加？

$$f_6(x) = f_5(x) + T_6(x)$$
$$= T_1(x) + T_2(x) + T_3(x) + T_4(x) + T_5(x) + T_6(x)$$

$$T_1(x) = \begin{cases} 6.24, & x < 6.5 \\ 8.91, & x \geq 6.5 \end{cases}$$

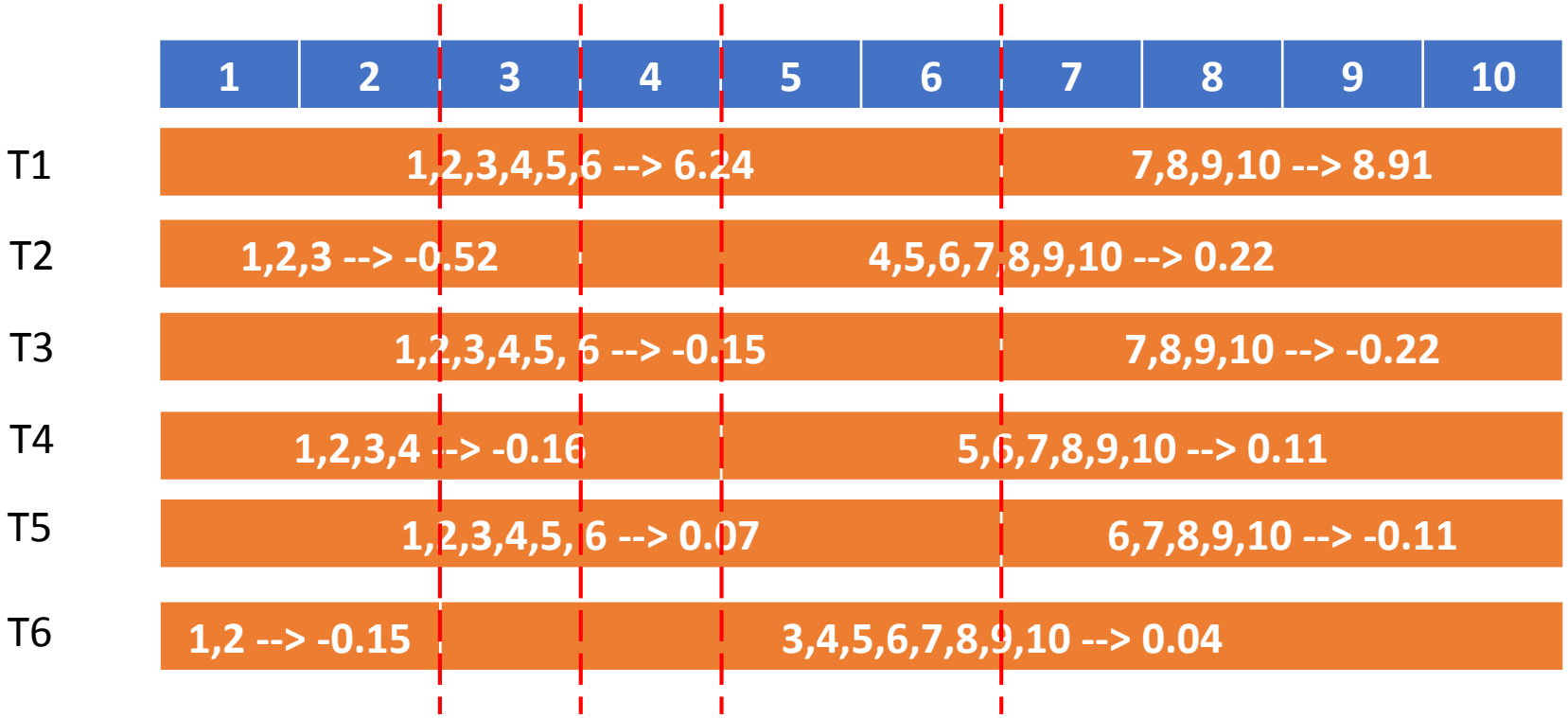
$$T_2(x) = \begin{cases} -0.52, & x < 3.5 \\ 0.22, & x \geq 3.5 \end{cases}$$

$$T_3(x) = \begin{cases} 0.15, & x < 6.5 \\ -0.22, & x \geq 6.5 \end{cases}$$

$$T_4(x) = \begin{cases} -0.16, & x < 4.5 \\ 0.11, & x \geq 4.5 \end{cases}$$

$$T_5(x) = \begin{cases} 0.07, & x < 6.5 \\ -0.11, & x \geq 6.5 \end{cases}$$

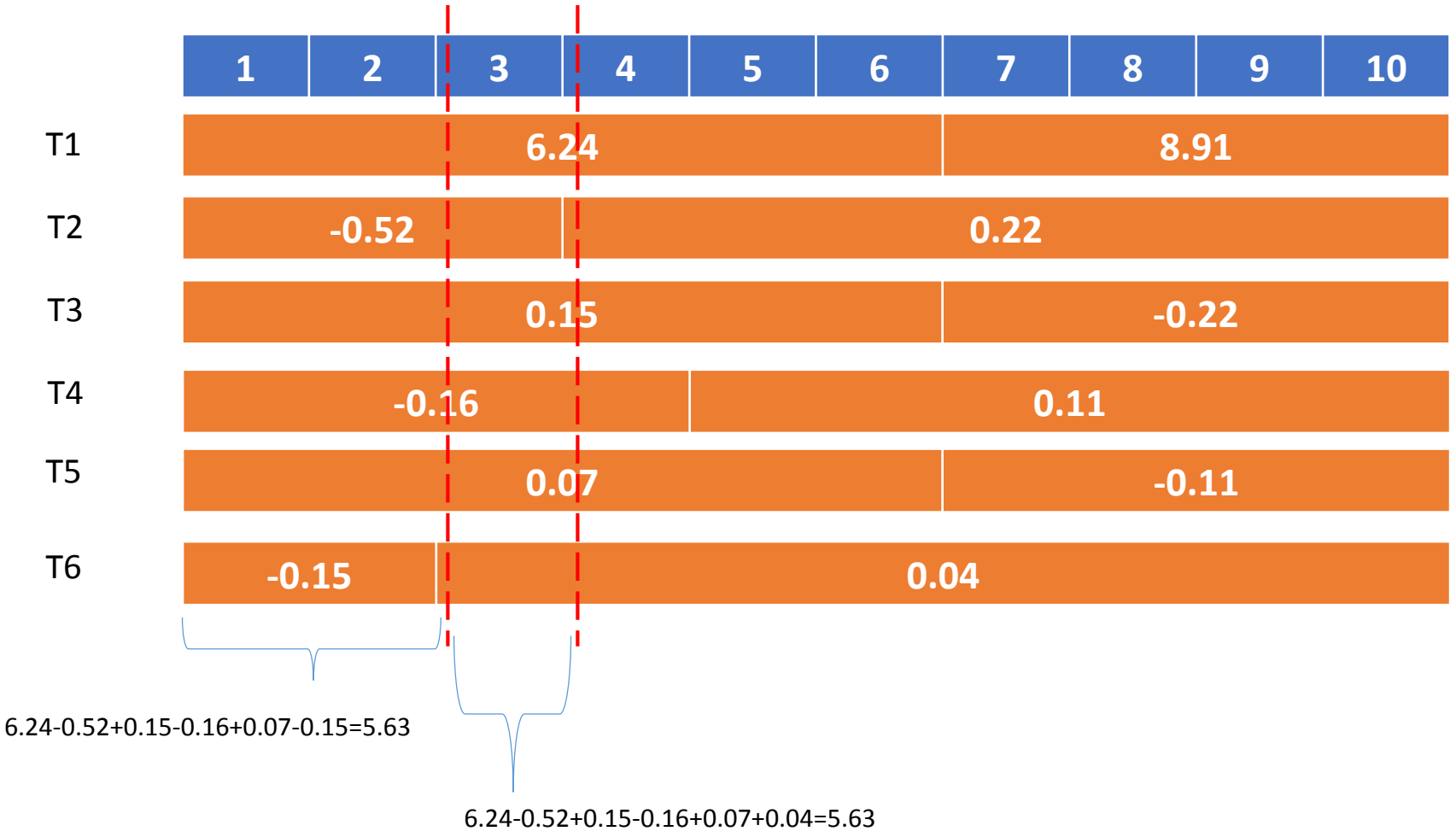
$$T_6(x) = \begin{cases} -0.15, & x < 2.5 \\ 0.04, & x \geq 2.5 \end{cases}$$



同时计算出
L1 , L2 , L3 , L4 , L5 , L6

树们如何相加？

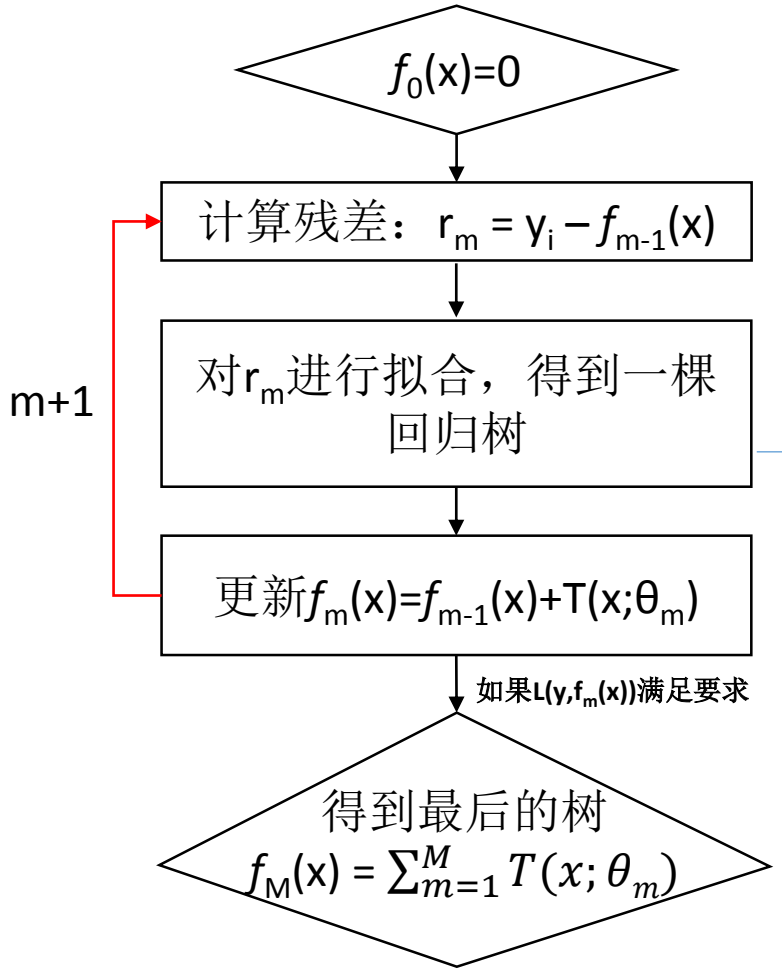
$$f_6(x) = f_5(x) + T_6(x)$$
$$= T_1(x) + T_2(x) + T_3(x) + T_4(x) + T_5(x) + T_6(x)$$



$$F(x) = f_6(x) = \begin{cases} 5.63, & x < 2.5 \\ 5.82, & 2.5 \le x < 3.5 \\ 6.56, & 3.5 \le x < 4.5 \\ 6.83, & 4.5 \le x < 6.5 \\ 8.95, & x \ge 6.5 \end{cases}$$

此时，$L(y, f_6(x)) = 0.17$
我们假设此时已经满足要求，
就不在继续计算下去了

GBDT的GB*(Gradient Boosting)*



$$L(y, f_m(x))=(y - f_m(x))^2$$
$$=L[y, f_{m-1}(x) + T(x;\theta)]^2$$
$$=[y - f_{m-1}(x) - T(x;\theta)]^2$$
$$= [r_{m-1} - T(x;\theta)]^2$$

$$r_m = -[\frac{\partial L(y,f(x))}{\partial f(x)}]_{f(x)=fm-1(x)}$$

利用损失函数的负梯度在当前模型的值，
作为回归问题提升术算法中残差的近似值

这里需要注意的是，前面我们提到一个算法步骤是Line search（具体见论文）。在GBDT里，我们通过不会直接把上一个轮的预测值$F_{m-1}(x)$直接加上$\sum_{j=1}^{J} \gamma_{jm} I(x_i \in R_{jm})$，而是会在$\sum_{j=1}^{J} \gamma_{jm} I(x_i \in R_{jm})$乘上一个学习率。可以理解，因为如果每次完全加上（学习率为1）本轮模型的预测值容易导致过拟合。所以通常在GBDT中的做法（也叫Shrinkage）

GBDT的算法流程

输入： $(x_i, y_i), T, L$

1. 初始化 f_0
2. for t = 1 to T do

2.1 计算响应：

$$\tilde{y}_i = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{t-1}(x)}, i = 1, 2, \ldots N$$

2.2 学习第t棵树：

$$w^* = \arg\min_{w} \sum_{i=1}^{N} \left(\tilde{y}_i - h_t(x_i; w)\right)^2$$

2.3 line search找步长：

$$\rho^* = \arg\min_{\rho} \sum_{i=1}^{N} L\left(y_i, F_{t-1}(x_i) + \rho h_t(x_i; w^*)\right)$$

2.4 令 $f_t = \rho^* h_t(x; w^*)$

更新模型：

$$F_t = F_{t-1} + f_t$$

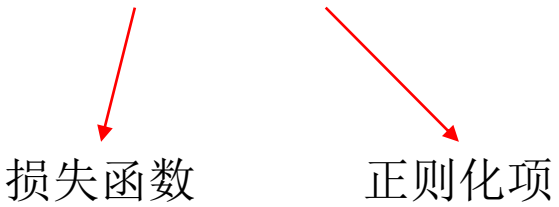
3. 输出 F_T

xgboost

Xgboost仍然是基于boosting的思想，但是在实现过程中，其所需要优化的目标函数有所变化。

Xgboost的目标函数：

$$Obj(\theta) = L(\theta) + \Omega(\theta)$$



损失函数

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

$$L^t = \sum_{i=1}^{n} l\left(y_i, y_i^{t-1} + f_t(xi)\right)$$

泰勒公式:

$$f(x + \Delta x) \approx f(x) + f'(x)\Delta x + \frac{1}{2} f''(x)\Delta x^2 + o(\Delta x^3)$$

令 $g_i = \frac{\partial L(yi, yit^{-1})}{\partial y_i^{t-1}}$ $h_i = \frac{\partial^2 L(yi, yit^{-1})}{\partial y_i^{t-1}}$

则，$L^t = \sum_{i=1}^{n} \left[L(y_i, yit^{-1}) + gift(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right]$

目标函数作变换

$$L^t = \sum_{i=1}^{n} l\left(y_i, y_i^{t-1} + f_t(x_i)\right) + \Omega(f_t)$$

$$L'^t = \sum_{i=1}^{n} \left[L(y_i, yit^{-1}) + gi(x_i) f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t)$$

均为已知，故舍去

$$L'^t = \sum_{i=1}^{n} \left[g_i(x_i) f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t)$$

$$= \sum_{i=1}^{n} \left[g_i w_{q(xi)} + \frac{1}{2} h_i w^2_{q(xi)}\right] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} w_j^2$$

对样本累加　　对叶子累加

把 $f_t, \Omega(f_t)$ 写成树结构的形式，即把下式代入目标函数中

$$f(\mathbf{x}) = w_{q(\mathbf{x})} \qquad \Omega(f) = \gamma T + \frac{1}{2}\lambda \|w\|^2$$

$$= \sum_{j=1}^{T} \left[\sum_{i \in I_j} g_{m,i}\, w_j + \frac{1}{2} \sum_{i \in I_j} h_{m,i}\, w_j^2 + \frac{1}{2} \lambda \sum_{j=1}^{T} w_t^2\right] + \gamma T$$

$$= \sum_{j=1}^{T} \left[\underbrace{\sum_{i \in I_j} g_{m,i}}_{G_j} w_j + \frac{1}{2} \left(\underbrace{\sum_{i \in I_j} h_{m,i}}_{H_j} + \lambda\right) w_j^2\right] + \gamma T$$

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

$$L't = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j + \lambda)w_j^2\right] + \gamma T$$

假如，树的结构（q(x)）确定，为了使目标函数最小，
可以令其导数为0，解得每个叶节点的最优预测分数为：

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

代入，得：

$$L't = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T$$

填坑

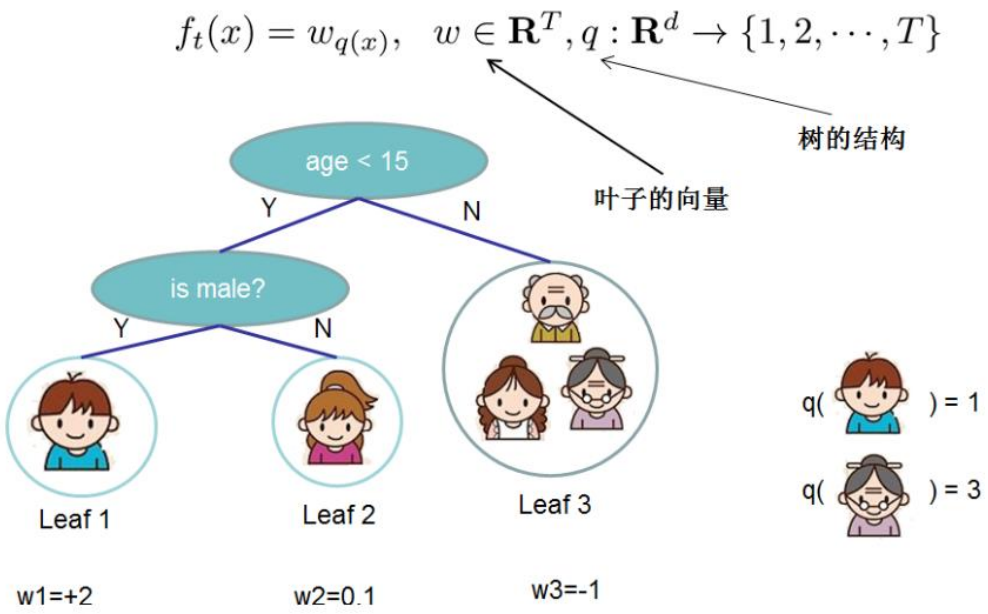
把 $f_t,\Omega(f_t)$ 写成树结构的形式，即把下式代入目标函数中

$$f(\mathbf{x}) = w_{q(\mathbf{x})} \qquad \Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^2$$

树的定义：
把树拆分成结构部分q和叶子分数部分w

$\phi(\mathbf{x}) = w_{q(\mathbf{x})}$ ，$\mathbf{w} \in R^T, q\colon R^D \to \{1, \dots, T\}$

- 结构函数q：把输入映射到叶子的索引号
- w：给出每个索引号对应的叶子的分数
- T为树中叶子结点的数目，D为特征维数



正则项

- XGBoost的目标函数（函数空间）

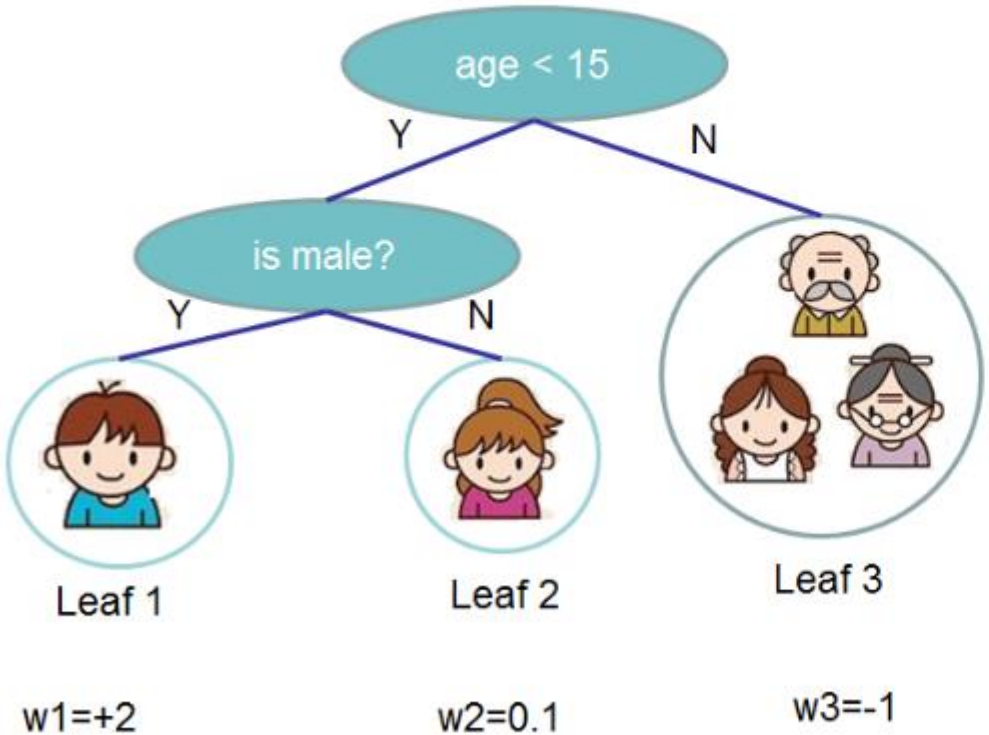
$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k)$$

正则项对每棵回归树的复杂度进行了惩罚

- 相比原始的GBDT，XGBoost的目标函数多了正则项，使得学习出来的模型更加不容易过拟合。

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

叶子的个数　　w的L2模平方



$$\Omega = \gamma 3 + \frac{1}{2}\lambda(4 + 0.01 + 1)$$

未涉及的内容

- 如何建树？
  - 树节点的分裂
  - 稀疏值的处理
- 特征的重要性

Reference

- 1.《统计学习方法》 李航. 清华大学出版社
- 2.《XGBoost 从基础到实战》冒**.CSDN
- 3.《机器学习 升级版V》邹博.chinahadoop.cn
- 4.《GBDT 算法原理与系统设计简介》wepon
- 5.《GBDT原理与Sklearn源码分析-回归篇》kingsam_ https://blog.csdn.net/qq_22238533/article/details/79185969
- 6.《GBDT原理与Sklearn源码分析-分类篇》 kingsam_ https://blog.csdn.net/qq_22238533/article/details/79192579

- 水平有限，如有错误或不当之处，恳请批评指正！
- 517949233@qq.com