XGBoost算法推导

XGBoost整体上仍属于GBDT算法系统，也是由多个基模型组成的加性模型

所以XGBoost可表示为：$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$

前向分步算法第t次迭代的基模型为$f_t(x)$，有：

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} \hat{y}_i^{(t-1)} + f_t(x_i)$$

XGBoost损失函数基本形式由经验损失项和正则化项组成：

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{i=1}^{t} \Omega(f_i) \quad \cdots \text{式(1)}$$

$\sum_{i=1}^{n} l(y_i, \hat{y}_i)$为经验损失项，表示训练数据预测值与真实值之间的损失；$\sum_{i=1}^{t} \Omega(f_i)$为正则化项

表示全部t棵树的复杂度之和，这也是XGBoost控制模型过拟合的方法。

根据前向分步算法，以第t步模型为例，假设模型对第i个样本x_i的预测值为：

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

式(1)的目标函数可以改写为：

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i)$$

$$= \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i)$$

因为前t-1棵树的结构已经确定，所以前t-1棵树的复杂度之和也可以表示为常数

$$\sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + \text{Constant} \quad \cdots \text{式(2)}$$

针对式(2)的前半部分使用二阶泰勒展开式，这里需要用到函数的二阶导数，相应的

损失函数经验损失项可以改写为：

$$l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) = l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)$$

g_i为损失函数一阶导数，h_i为损失函数二阶导数。注意这里是对$\hat{y}_i^{(t-1)}$求导。

将上式代入式(2)中得到损失函数的近似表达式：

$$L^{(t)} \approx \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + \text{Constant}$$

去掉相关常数项，得到简化后的损失函数表达式为：

$$L^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t)$$

由上式知，只需要求出损失函数每一步的一阶导数和二阶导数值，并对目标函数进行优化求解，就可以得到前向分步中每一步的模型$f(x)$，最后根据加法得到XGBoost。

为了计算XGBoost决策树对结点分裂条件，进一步进行推导：

假设一棵决策树是由叶子结点的权重w和样本实例到叶子结点的映射关系q构成，映射关系可以理解为决策树的分支结构。

$$f_t(x) = w_{q(x)}$$

定义决策树复杂度的正则化项。模型复杂度Ω可由单棵决策树的叶子结点数T和叶子权重w决定，所以模型的复杂度可以表现为：

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

XGBoost的损失函数改写为

$$L^{(t)} \approx \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t)$$

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

将属于第j个叶子结点的所有样本x_i划入一个叶子结点的样本集合中，因而：

$$L^{(t)} = \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)w_j^2\right] + \gamma T$$

定义$G_j = \sum_{i\in I_j} g_i$，$H_j = \sum_{i\in I_j} h_i$，其中$G_j$可以理解为叶子结点$j$所包含样本的一阶偏导数累加之和，$H_j$可以理解为叶子结点$j$所包含样本的二阶偏导数累加之和，都为常量。

将G_j和H_j代入上式，损失函数变换为 $L^{(t)} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}(H_j+\lambda)w_j^2\right] + \gamma T$

又对每个叶子结点j，将其从目标函数中单独取出： $G_j w_j + \frac{1}{2}(H_j+\lambda)w_j^2$

上式是一个只包含一个变量叶子结点权重w_j的一元二次函数，可根据最值公式求其最值点。当相互独立的每棵树的叶子结点都达到最优值时，整个损失函数也相应地达到最优。在树结构固定的情况下，对上式求导并令其为0，可得到最优点、和最优值为：

$$w_j^* = -\frac{G_j}{H_j+\lambda} \qquad L = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$$

假设决策树模型在某个结点进行j分裂，分裂前的损失函数为：

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

分裂后的信息增益为：

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

如果增益 Gain > 0，即分裂为两个叶子结点后，目标函数下降了，则考虑此次结果。

XGBoost 的核心就是通过损失函数展到二阶导数来进一步逼近真实损失。

当定义或者选择 XGBoost 损失函数时，需要其二阶可导，以平方损失为例：

$$l(y_i, \hat{y}_i^{(t-1)}) = (y_i - \hat{y}_i^{(t-1)})^2$$

对其的一阶导数和二阶导数分别为：

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} = -2(y_i - \hat{y}_i^{(t-1)})$$

$$h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2} = 2$$