集成学习终篇：从CART回归树开始，经历BDT、GBDT彻底理解XGBoost

大电鳗
机器学习

要了解此系列相关的算法，基础为CART回归树，将这些算法整理起来方便理解和比较其中的不同。

一.CART回归树

回归树的模型可以表示如下：

$$T(x)=\sum_{m=1}^{M}c_mI(x_i\in R_m)$$

上式中， c_m 为对应叶子节点的输出值， $I(x_i\in R_m)$ 为指示函数，当x属于 R_m 时，值为1，否则为0。

回归树的建立过程，优化策略或损失函数为最小化平方误差，即最小化下式：

$$\sum_{x_i\in D}(T(x_i)-y_i)^2$$

CART回归树的建树过程是二分裂节点，并且保证分裂的结果符合最小化平方误差，这里采用了比较暴力的遍历法，即遍历所有特征j和每个特征的多个阈值s，以平方误差最小的组合作为分裂依据，数学描述如下：

$$\arg\min_{j,s}[\min_{c_1}\sum_{x_i\in R_1}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2}(y_i-c_2)^2]$$

上式中，R为以s为分割点分割的左右子树样本合集，c为该集合的均值。

确定了j，s后，就可以就行分裂了，将树分裂为左右两个区域：

$$R_1(j,s)=\{x_i[j]\leqslant s\},R_2(j,s)=\{x_i[j]>s\}$$

$$c_m = \frac{1}{N_m} \sum_{x_i \in y_m} y_i$$

二.提升树BDT

概论：提升树是一个集成算法，训练出多颗CART树进行预测。其核心思想为残差拟合。

什么是残差拟合？

用训练样本训练出了一颗决策树 T_i ,将样本X的特征x喂入 T_i 后得预测值 $\bar{y}$ *,残差及为 $\bar{y}$ 和X的真实标签y的距离。例如，当损失函数为均方损失时，残差为 $y-\bar{y}$ 。*

提升树为加法模型，数学表示：

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

也可以迭代表示为：

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

损失函数为：

$$\arg\min_{\theta} \sum_{i=1}^{n} L(y_i, f_m(x)) = \arg\min_{\theta} \sum_{i=1}^{n} L(y_i, f_{m-1}(x) + T(x;\theta_m))$$

当L为均方误差时，每一轮的损失函数如下：

$$\begin{aligned} L(y_i, f_{m-1}(x) + T(x;\theta_m)) &= (y_i - f_{m-1}(x) - T(x;\theta_m))^2 \\ &= (r_{m,i} - T(x;\theta_m))^2 \end{aligned}$$

$r_{m,i} = y_i - f_{m-1}(x)$ 即为残差，同时，这个式子还表示在进行第m颗决策树生成时，要 $r_{m,i}$ 作为标签值去拟合，即 （X, r_m） 作为样本输入去拟合第m棵决策树 $T(x;\theta_m)$ 。

1.GBDT原理

梯度提升树是基于提升树改进而来的，运用了一些BDT的思想，但又有不同，这里先说一下两个不同点：

(1).在拟合值方面,GBDT用负梯度代替了BDT中的残差，其本质是用泰勒一阶展开式近似值。

(2).BDT中，叶子节点的输出取平均值（由CART回归树的建立过程决定）；而GBDT叶子节点的输出需要拟合损失函数最好的输出。

2.GBDT回归算法流程

(1).初始化强学习器:

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

(2).在第t轮中，计算每个样本i的负梯度

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

以均方损失为例:

$$L(y_i,f(x_i)=(y_i-f(x_i))^2$$

$$r_{t,i}=\frac{\partial(y_i-f(x_i))^2}{\partial f(x_i)}=y_i-f(x_i)$$

(3).用 （$x_i,r_{t,i}$），$i=1,2,\ldots,m$ 拟合得到第t棵CART回归树 T_t ，叶子节点区域划分为： $R_{t,j}$ ，j=1,2,...,J.

(4).遍历节点区域，计算回归树 T_t 的每个叶子节点 $R_{t,j}$ 的输出值即最佳拟合值 $c_{t,j}$ ：

(5).更新强学习器:

$$f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} c_{t,j} I(x \in R)$$

(6).重复(2)-(5)直到达到终止条件，得最终强学习器的表达式:

$$f(x) = f_0(x) + \sum_{t=1}^{T} \sum_{j=1}^{J} c_{t,j} I(x \in R)$$

3.GBDT二分类算法

对于分类问题，一般将损失函数改为指数损失或者对数似然损失。

使用对数似然损失函数:

$$L(y, f(x)) = \log(1 + exp(-yf(x)))$$

此时的负梯度为:

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

叶节点最佳拟合值:

$$c_{t,j} = \arg\min_{c} \sum_{x_i \in R_{t,j}} \log(1 + exp(-y_i + f_{t-1}(x_i) + c))$$

由于上式比较难求，常用近似值:

$$c_{t,j} = \frac{\sum_{x_i \in R_{t,j}} r_{t,j}}{\sum_{x_i \in R_{t,j}} |r_{t,j}|(1 - |r_{t,j}|)}$$

4.GBDT多分类算法

$$L(y, f(x)) = -\sum_{k=1}^{K} y_k log p_k(x)$$

y为one-hot编码，及样本输出的类别为k， $y_k = 1$ ，此时，概率为：

$$p_k(x) = \frac{exp(f_k(x))}{\sum_{l=1}^{K} exp(f_l(x))}$$

多分类时，会为每一个类别训练一个决策树。第t轮，第i个样本 ，对应的类别l的负梯度为：

$$r_{t,i} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{t-1,l}(x)} = y_{i,l} - p_{t-1,l}(x_i)$$

叶子节点输出：

$$c_{t,l,j} = \arg\min_{c} \sum_{x_i \in R_{t,l,j}} L(y_i, f_{t-1,l}(x_i) + c)$$

一般用近似值：

$$c_{t,j} = \frac{K-1}{K} \frac{\sum_{x_i \in R_{t,l,j}} r_{t,l,j}}{\sum_{x_i \in R_{t,l,j}} |r_{t,l,j}|(1 - |r_{t,l,j}|)}$$

5.GBDT正则化

$$f_k(x) = f_{k-1}(x) + v h_k(x), 0 < v \leqslant 1$$

四.XGBoost

1.XGBoost与GBDT的不同点

(1)GBDT的弱学习器只支持决策树，XGBoost的弱学习器支持其他机器学习器。

(3)GBDT的损失函数只对误差部分做负梯度（一阶泰勒）展开，而XGBoost损失函数对误差部分做二阶泰勒展开，拟合更准确。

(4)优化了算法的运行效率。

(5)缺失值的处理。

2.XGBoost损失函数推导

GBDT损失函数:

$$L_t=\sum_{i=1}^{m} L(y_i, f_{t-1}(x_i)+h_t(x))$$

XGBoost损失函数相当于GBDT损失函数加上正则化项 $\Omega(h_t)$ ：

$$\begin{aligned} L_t &=\sum_{i=1}^{m} L(y_i, f_{t-1}(x_i)+h_t(x))+\Omega(h_t) \\ &=\sum_{i=1}^{m} L(y_i, f_{t-1}(x_i)+h_t(x))+\gamma J+\frac{\lambda}{2}\sum_{j=1}^{J} w_{t,j}^2 \end{aligned}$$

其中，γ, λ 为 正则化系数，J为叶子节点个数， $w_{t,j}$ 和GBDT中的 $c_{t,j}$ 相同，为对应叶子节点 $R_{t,j}$ 的输出(最佳拟合值)。

泰勒二阶展开式为：

$$f(x+\Delta x) \simeq f(x)+f'(x)\Delta x+f''(x)\Delta x^2$$

对于 L_t ,我们将 L(x)当做f(x), $f_{t-1}(x)$ 当做 x ， $h_t(x)$ 当做 Δx ，则损失函数近似为：

$$\begin{aligned} L_t &=\sum_{i=1}^{m} L(y_i, f_{t-1}(x_i)+h_t(x_i))+\gamma J+\frac{\lambda}{2}\sum_{j=1}^{J} w_{t,j}^2 \\ &\simeq \sum_{i=1}^{m}[L(y_i, f_{t-1}(x_i))+\frac{\partial L(y_i, f_{t-1}(x_i))}{\partial f_{t-1}(x)}h_t(x_i)+\frac{\partial^2 L(y_i, f_{t-1}(x_i))}{\partial f_{t-1}^2(x)}h_t^2(x_i)]+\gamma J+\frac{\lambda}{2}\sum_{j=}^{J} \end{aligned}$$

令：

损失函数更新为：

$$L_t \simeq \sum_{i=1}^{m}[L(y_i,f_{t-1}(x_i))+g_{t,i}h_t(x_i)+\frac{1}{2}h_{t,i}h_t^2(x_i)]+\gamma J+\frac{\lambda}{2}\sum_{j=1}^{J}w_{t,j}^2$$

最小化 L_t 的 过程中，由于 $L(y_i,f_{t-1}(x_i))$ 为常数，对最小化过程不产生影响，将其省略。

同时， x_i 经过第t个决策树 $h_t(x)$ 被分配到子节点区域 $R_{t,j}$ ， $R_{t,j}$ 的输出值为 $w_{t,j}$ ，也就是说 $h_t(x_i)$ 的结果就是 $w_{t,j}$ 。

根据以上两点，更新损失函数：

$$\begin{aligned}L_t &\simeq \sum_{i=1}^{m}[g_{t,i}h_t(x_i)+\frac{1}{2}h_{t,i}h_t^2(x_i)]+\gamma J+\frac{\lambda}{2}\sum_{j=1}^{J}w_{t,j}^2\\&=\sum_{j=1}^{J}[\sum_{x_i\in R_{t,j}}g_{t,i}w_{t,j}+\frac{1}{2}\sum_{x_i\in R_{t,j}}h_{t,i}w_{t,j}^2]+\gamma J+\frac{\lambda}{2}\sum_{j=1}^{J}w_{t,j}^2\\&=\sum_{j=1}^{J}[\sum_{x_i\in R_{t,j}}g_{t,i}w_{t,j}+\frac{1}{2}\sum_{x_i\in R_{t,j}}(h_{t,i}+\lambda)w_{t,j}^2]+\gamma J\end{aligned}$$

令：

$$G_{t,j}=\sum_{x_i\in R_{t,j}}g_{t,i},H_{t,j}=\sum_{x_i\in R_{t,j}}h_{t,i}$$

损失函数更新为：

$$L_t=\sum_{j=1}^{J}[G_{t,j}w_{t,j}+\frac{1}{2}(H_{t,j}+\lambda)w_{t,j}^2]+\gamma J$$

3.XGBoost损失函数求解

(1)叶子节点最佳输出值

点区域和每个区域的最佳拟合值 $w_{t,j}^2$ 。具体做法如下：

在损失函数 L_t 中，只有 $w_{t,j}$ 为未知量，此时，对其求导并令导数为0：

$$\frac{\partial L_t}{\partial w_{t,j}} = G_{t,j} + (H_{t,j} + \lambda) w_{t,j} = 0$$
$$\Rightarrow w_{t,j} = -\frac{G_{t,j}}{H_{t,j} + \lambda}$$

(2)基学习器决策树的分裂方式

GBDT中，CART回归树的分裂依据是选择平方误差最小的特征及分割点进行分裂。

XGBoost的分裂依据为最小化损失函数的误差。具体过程如下：

首先根据已经求得的 $w_{t,j}$ ，继续优化损失函数：

$$\begin{aligned} w_{t,j} = -\frac{G_{t,j}}{H_{t,j} + \lambda} \Rightarrow L_t &= \sum_{j=1}^{J} [G_{t,j} w_{t,j} + \frac{1}{2} (H_{t,j} + \lambda) w_{t,j}^2] + \gamma J \\ &= -\sum_{j=1}^{J} [\frac{G_{t,j}^2}{H_{t,j} + \lambda} - \frac{1}{2} \frac{G_{t,j}^2}{H_{t,j} + \lambda}] + \gamma J \\ &= -\frac{1}{2} \sum_{j=1}^{J} \frac{G_{t,j}^2}{H_{t,j} + \lambda} + \gamma J \end{aligned}$$

假设我们现在根据样本的某个特征A的某个值a将节点分裂出左、右两个节点，则左子树对应的G和H为 G_L, H_L ，对应分值为：

$$-\frac{1}{2} \frac{G_L^2}{H_L + \lambda} + \gamma$$

右子树对应的G和H为 G_R, H_R ，对应分值为：

$$-\frac{1}{2} \frac{G_R^2}{H_R + \lambda} + \gamma$$

$$-\frac{1}{2}\frac{G_L^2+G_R^2}{H_L+H_R+\lambda}+\gamma$$

则分裂过后的总分值：

$$-\frac{1}{2}\frac{G_R^2}{H_R+\lambda}+\gamma-\frac{1}{2}\frac{G_R^2}{H_R+\lambda}+\gamma+\frac{1}{2}\frac{G_L^2+G_R^2}{H_L+H_R+\lambda}-\gamma$$

整理上式并取反，最小化损失函数变为最大化下式：

$$\frac{1}{2}[\frac{G_R^2}{H_R+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{G_L^2+G_R^2}{H_L+H_R+\lambda}]-\gamma$$

及遍历样本的所有特征和对应取值，选择使上式最大化的特征和切分点组合进行分裂。

编辑于 2019-08-04

机器学习

推荐阅读

集成学习(Ensemble Learning)——提升树...

周明 发表于水木学者

还没有评论

写下你的评论...