Xgboost

2022-10-07

基模型

考虑一个二叉树，如何生长？最大化父节点的样本集合的损失值与其之下的两个子节点的样本集合损失值的差值。

回归树

变量竞争准则：

$$\Delta_{SST} = \sum_{i \in father\ node} (y_i - \overline{y})^2 - \left\{ \sum_{i \in son\ node\ 1} (y_i - \overline{y})^2 + \sum_{i \in son\ node\ 2} (y_i - \overline{y})^2 \right\}$$

拆分变量以及生长：

1. 变量内部寻找最佳分割。数值变量一般遍历样本之间的均值找到该变量使得 Δ_{SST} 最大的分割点 τ，而分类变量遍历所有的组合找使得 Δ 最大的 $\mathcal{A}$ 以及其补集 $\mathcal{A}^c$。
2. 变量之间寻找最佳拆分变量。比较每个变量最佳分割的 Δ_{SST} 的大小，拥有最大 Δ 的变量，作为拆分分量，其 τ 或 $\mathcal{A}$ 作为拆分准则。
3. 重复 1，2。根据不同的准则，停止生长。比如总 R^2 增长不会大于复杂性参数 (complexity parameter, cp) 的某个值时，或者到了分叉的限制点，或者某个节点的观测值太少。

Note: 对于回归树而言，样本在某个节点的预测值就是该节点样本集合的均值；每个节点下面如果停止生长，那么序号会保留。

决策树

决策树拆分变量以及生长的过程与回归树类似，只不过竞争的准则与回归有所不同。

假定观测值一共有 K 类，在一个节点的观测值中属于第 i 类的比例为 $p_k(k = 1, 2, ..., K)$. 显然 $\sum_{k=1}^{K} p_k = 1$. 常有的准则有下面几种：

- 误分率按照少数服从多数的原则，在树的某叉中，某一类 i 的数目是最多的，则类 i 被认为是分类正确的，那么误分率为 $1 - p_i$
- 熵定义为 —— $\sum_{k=1}^{K} p_k log_2 p_k$. 在所有的观测值都为一类的时候，熵为 0. 因此所选择的拆分变量是使得父节点的熵和子节点的熵差别（称为信息增益 (information gain)）最大的变量。子节点的熵应为各个子节点的数目比例对各个节点熵的加权平均。
- Gini 不纯度（或 Gini 指数）定义为

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

  在所有的观测值都为一类的时候，Gini 不纯度的度量都为 0. 因此，所选择的拆分变量是使得父节点 Gini 不纯度和子节点的 Gini 不纯度差别最大的变量。子节点的 Gini 不纯度应该用各个子节点观测值数目比例对各个节点的 Gini 不纯度的加权平均来计算。

何时停止树的生长按照某些确定的误判函数计算。例如控制参数 cp，如果纯度改变达不到它的值，就不会再分叉了。另外还有一个复杂性（complexity parameter）的量 $\alpha \in [0, \infty)$, 计算每增加一个变量到模型中的损失。

再如，从长成的树的终节点开始剪枝，每剪一次，看看误差是不是增加，如果超过要求则停止剪枝。

另一种剪枝原则是使得下式最小：

$$\frac{T \quad t \quad - \ T}{T \quad - \ T \quad t}$$

其实本质就是最大化剪去一个节点能够减少的误差。

Bagging

自助法抽样：对样本的抽样；一般我们只有一个样本，通过对这个样本进行抽样（部分或者全部），得到许多自助法样本。

抽样：对总体的抽样。

自助法的数学推断以及与总体近似的证明：

经典的统计推断基于抽样分布 (sampling distribution), 抽样分布是从总体重复抽样所得样本的分布，但是从原始总体中重复抽样是很困难的，因此我们只有对分布做出各种无法验证的假定，然后根据这些假定对统计量或者参数做出推断。

自主法也有其限制所在：太小的样本不是总体的很好近似；不平衡样本；不规范数据；以及非独立观测值组成的数据。

决策树可以看作一个简单的学习器，多个决策树的组合可以组合成非常精确的学习器。但是同一个数据产生的学习器是一样的，但是如果对原始数据做自助法抽样（放回抽样）则会产生不同的数据，因而会产生不同的决策树，如果再在抽样概率以及决策树拆分变量等方面做些改变，则可以产生基于同一个原始数据的各个不同的决策树，形成一个可以 “投票”（平均或加权平均）的决策树群。

Note：bagging 解决的是基模型 overfitting 的问题。

Boosting

核心：基于残差的拟合；

假定第一个基模型的拟合以后每个样本的残差为 $[y_1, y_2, ..., y_m]$, 那么第二个基模型拟合的观测值就为 $[y_1, y_2, ..., y_m]$.

Note: boosting 解决的是 underfitting 的问题。

xgboost

构造目标函数

假设已经训练了 K 棵树，则对于第 i 个样本（**最终**) 预测值为:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

目标函数:

$$obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

对于 K 个模型的优化是叠加式训练；给定 x_i: $\hat{y_i}^{(0)} = 0 \leftarrow$ (default case)
$\hat{y_i}^{(1)} = \hat{y_i}^{(0)} + f_1(x_i)$ $\hat{y_i}^{(2)} = \hat{y_i}^{(0)} + f_1(x_i) + f_2(x_i)$
…… $\hat{y_i}^{(k)} = \hat{y_i}^{(0)} + f_1(x_i) + f_2(x_i) + ... + f_{k-1}(x_i) + f_k(x_i) \rightarrow \hat{y_i}^{(k)} = \hat{y}^{(k-1)} + f_k(x_k)$ 当训练第 k 树时，目标函数写作:

$$obj_k = \sum_{i=1}^{n} l(y_i, \hat{y}^{(k-1)} + f_k(x_k)) + \sum_{j=1}^{K-1} \Omega(f_j) + \Omega(f_k) \rightarrow \sum_{i=1}^{n} l(y_i, \hat{y}^{(k-1)} + f_k(x_k)) + \Omega(f_k)$$

目标函数直接优化难，如何近似?（talor expansion)

$$obj_k = \sum_{i}^{n} \left[l(y_i, \hat{y_i}^{(k-1)}) + \partial_{\hat{y_i}^{(k-1)})} l(y_i, \hat{y_i}^{(k-1)}) * f_k(x_i) + \frac{1}{2} \partial^2_{\hat{y_i}^{(k-1)})} l(y_i, \hat{y_i}^{(k-1)}) * f_k^2(x_i) \right] + \Omega(f_k) = \sum_{i}^{n} \left[g_i * f_k(x_i) + \right.$$

Note: g_i 以及 h_i 就是残差的一种形式，传递给第 k 个模型训练的方向；记住 g_i 以及 h_i 是样本 i 具有两个常量可以帮助后面的理解。

如何把树的结构引入目标函数

参数化一棵树：定义：$q(x_i) \rightarrow$ 样本 x_i 在树的哪个叶节点。$\omega \rightarrow$ 叶节点的权重值（值）那么 $\omega_{q(x_i)} \rightarrow$ 样本 x_i 的预测值 $f_k(x_i) = \omega_{q(x_i)}$ $I_j \rightarrow$ 在叶子节点 j 的样本集合。

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

新的目标函数:

$$obj_k = \sum_{i}^{n} \left[g_i * f_k(x_i) + \frac{1}{2} h_i * f_k^2(x_i) \right] + \Omega(f_k) = \sum_{i}^{n} \left[g_i * \omega_{q(x_i)} + \omega^2_{q(x_i)}(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j = \sum_{j}^{T} \left[(\sum_{i \in I_j} g_i) \omega_j + \frac{1}{2} (\right.$$

Note：这里的关键在于上面提到的 g_i 以及 h_i 是样本 i 所具有的一个常量，观察目标函数可以知道，其值就是每个样本的 g_i 和 h_i 与其叶子节点 ω_j 值的乘积之和。目标函数形式的改变就是从样本遍历其和变为从叶子节点遍历其和。定义 $G_j = \sum_{i \in I_j} g_i$, $H_j = \sum_{i \in I_j} h_i$:

$$obj_k = \sum_{j=1}^{T} \left[G_i \omega_j + \frac{1}{2}(H_i + \lambda)\omega_j^2 \right] + \gamma T$$

Note: 可以看到最后的目标函数就是，某个叶子节点的样本集合一阶导数之和该节点权重值 + 二阶导数之和该节点权重值的平方。注意 G_i 以及 H_i 的都是常数。

利用贪心算法求解树的生长过程。

注意 G_i 以及 H_i 的都是常数，最后的目标函数实际熵就是一个形如 $b + ax^2$ 的二次函数，二次函数为了取得最大值，其 x 取值为：$\omega_j* = -\frac{G_i}{H_i+\lambda} + \gamma T$，那么目标函数取值为 $-\frac{1}{2}\sum_{j=1}^{T} \frac{G_i^2}{H_i+\lambda}$。如之前提到的决策树的生长，这里的生长，和其类似，不过分割的准则变成了 obj；每一次分割就是使得 Δ_{obj} 最大的方向。