Adaboost

3. 提升树

3.1 提升树算法

该模型是以决策树为基函数的加法模型，利用前向分步算法求解该模型的最优解。可以表示为

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

其中$T(\mathbf{x};\Theta_m)$表示决策树，Θ_m是决策树的参数，M是树的个数。

求解该模型最优解的过程如下 确定初始的提升树$f_0(\mathbf{x}) = 0$，第m步的模型是

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

定义损失函数为

$$\sum_{i=1}^{N} L(y^{(i)}, f_{m-1}(\mathbf{x}^{(i)}) + T(x^{(i)};\Theta_m))$$

这是在第m步的损失函数。对Θ_m进行最小化损失函数，从而求得$\Theta_m^\star$。

3.2 提升回归树模型

定义回归树

$$T(\mathbf{x};\Theta) = \sum_{j=1}^{J} c_j I(\mathbf{x} \in R_j)$$

其中$\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$，$\mathcal{X}$是输入空间。现在将输入空间$\mathcal{X}$划分成$J$个互补相交的区域$R_1, R_2, \ldots, R_J$。将$\mathbf{x}$映射到其中一个区域，输出这个区域对应的常量$c_j$。$J$表示回归树的叶节点个数，也表示复杂度。

这个回归树的参数$\Theta = \{(R_1, c_1), (R_2, c_2), \ldots, (R_J, c_J)\}$

3.3 提升回归树算法

求解提升回归树模型的参数，依旧采用前向分步算法。遵循3.1节的定义，同时定义损失函数为平方误差

$$\begin{aligned}\sum_{i=1}^{N} L(y^{(i)}, f_{m-1}(\mathbf{x}^{(i)}) + T(x^{(i)};\Theta_m)) &= \sum_{i=1}^{N} [y^{(i)} - f_{m-1}(\mathbf{x}^{(i)}) - T(\mathbf{x}^{(i)};\Theta_m)]^2 \\ &= \sum_{i=1}^{N} [r^{(i)} - T(\mathbf{x}^{(i)};\Theta_m)]^2 \quad \text{定义残差} r^{(i)} = y^{(i)} - f_{m-1}(\mathbf{x}^{(i)})\end{aligned}$$