机器学习概论 实验报告

Lab3: XGBoost

2020 年 12 月 21 日

目录

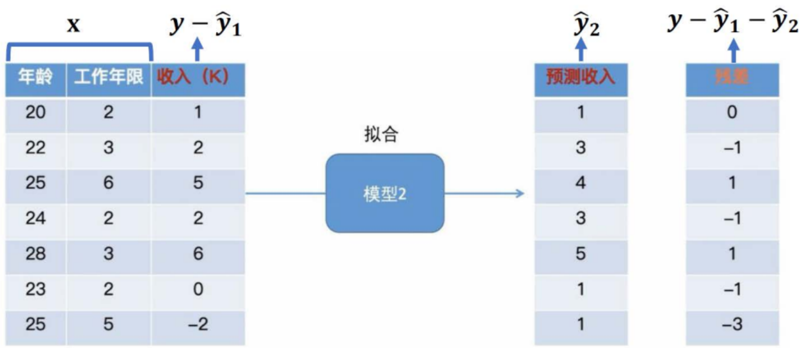
1 算法数学基础

XGBoost: e**X**treme **G**radient **Boost**ing, 最初是由 陈天奇 负责的研究项目, 后来其在各种数据挖掘比赛表现出了极大的性能优势.

1.1 提升树

1.1.1 基本思想

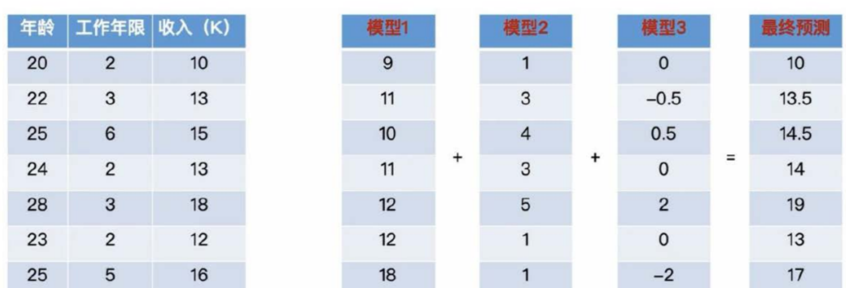
如果一个预测问题中, 训练的模型(称为 Model1)效果不好, 误差比较大, 我们可以通过基于残差的训练来拟合 Model2, 甚至 Model3, Model4,...



年龄	工作年限	收入（K）	预测收入	
20	2	1	1	0
22	3	2	3	−1
25	6	5	4	1
24	2	2	3	−1
28	3	6	5	1
23	2	0	1	−1
25	5	−2	1	−3

图 1: 基于 Model1 的残差训练 Model2

于是, 最后将各个模型的结果相加即可得到一个更优的模型. 这就是其基本思想.



年龄	工作年限	收入（K）	模型1		模型2		模型3		最终预测
20	2	10	9		1		0		10
22	3	13	11		3		−0.5		13.5
25	6	15	10		4		0.5		14.5
24	2	13	11	+	3	+	0	=	14
28	3	18	12		5		2		19
23	2	12	12		1		0		13
25	5	16	18		1		−2		17

图 2: 最后叠加每个模型的结果

1.1.2 相关公式

现在不妨考虑已经训练了 K 棵树, 则对第 i 个样本的预测为:

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

这里目标函数为

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

其中第一项为 训练误差, 第二项为正则化项(可以是叶节点个数, 叶节点评分, 树的深度等)

1.2 Gradient Tree Boost

1.2.1 目标函数与最优权重

考虑第 i 轮的预测为:

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

那么我们的目标函数就是

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

我们可以对目标函数中的训练误差项目, 进行泰勒展开, 并忽略常数项:

$$\begin{aligned}
\tilde{L}^{(t)} &= \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t(\mathbf{x}_i)] + \Omega(f_t) \\
&= \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t(\mathbf{x}_i)] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T}\omega_j^2 \\
&= \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\omega_j^2\right] + \gamma T
\end{aligned}$$

这里有$g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ 及$h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$
因此, 进一步我们能得到, 对于一个固定的树结构, 最优的叶子权重为:

$$\omega_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda},$$

而相应的最优值则为

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T.$$

1.2.2 split 的准则

根据上节给出的 $\tilde{\mathcal{L}}$, 我们很容易得到, 将叶子 split 的收益:

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right]$$

2 算法过程简介

根据上面的 split 准则, 我们可以设计这样的算法来计算最优 split 方法. 注意其中采用了按每个属性, 从小到大逐个分开地枚举, 从而贪心地得到当前最优 split 方法.

Algorithm 1 PISANO-DELETE(H,x)

Require: I, instance set of current node

Require: m, feature dimension

$gain \leftarrow 0$

$G \leftarrow \sum_{i \in I} g_i, H \leftarrow \sum_{i \in I} h_i$

for $k = 1$ to m **do**

  $G_L \leftarrow 0, H_L \leftarrow 0$

  for j in $sorted(I, by\ x_{jk})$ **do**

    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$

    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$

    $score \leftarrow \max\left(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda}\right)$

return split with max score