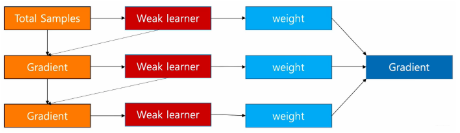
GBDT

GBDT全称梯度提升决策树，在传统机器学习算法里面是对真实分布拟合的最好的几种算法之一，在前几年深度学习还没有大行其道之前，gbdt在各种竞赛是大放异彩。原因大概有几个，一是效果确实挺不错；二是即可以用于分类也可以用于回归；三是可以筛选特征， 所以这个模型依然是一个非常重要的模型。

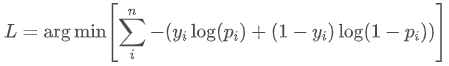
GBDT是通过采用加法模型（即基函数的线性组合），以及不断减小训练过程产生的误差来达到将数据分类或者回归的算法，其训练过程如下：



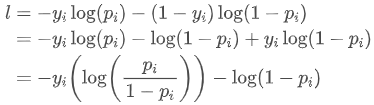
GBDT通过多轮迭代，每轮迭代会产生一个弱分类器，每个分类器在上一轮分类器的残差基础上进行训练。GBDT对弱分类器的要求一般是足够简单，并且低方差高偏差。因为训练的过程是通过降低偏差来不断提高最终分类器的精度。由于上述高偏差和简单的要求，每个分类回归树的深度不会很深。最终的总分类器是将每轮训练得到的弱分类器加权求和得到的（也就是加法模型）。

这里分析一下GBDT如何来进行二分类的，因为我们要明确一点就是GBDT每轮的训练是在上一轮的训练的残差基础之上进行训练的，而这里的残差指的就是当前模型的负梯度值， 这个就要求每轮迭代的时候，弱分类器的输出的结果相减是有意义的，而GBDT无论用于分类还是回归一直都是使用的CART 回归树，那么既然是回归树，是如何进行二分类问题的呢？

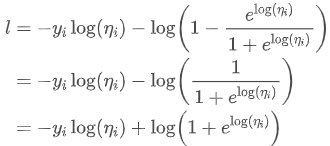
GBDT 来解决二分类问题和解决回归问题的本质是一样的，都是通过不断构建决策树的方式，使预测结果一步步的接近目标值， 但是二分类问题和回归问题的损失函数是不同的，回归问题中一般使用的是平方损失，而二分类问题中，GBDT和逻辑回归一样，使用的下面这个：



其中，yi是第i个样本的观测值，取值要么是0要么是1，而pi是第i个样本的预测值，取值是0-1之间的概率，由于我们知道GBDT拟合的残差是当前模型的负梯度，那么我们就需要求出这个模型的导数，即dL/dpi，对于某个特定的样本，求导的话就可以只考虑它本身， 去掉加和号，那么就变成了dl/dpi，其中l如下：



如果对逻辑回归非常熟悉的话，一定不会陌生吧，这就是对几率比取了个对数， 并且在逻辑回归里面这个式子会等于，所以才推出了的那个形式。这里令，即，则上面这个式子变成了：



这时候，我们对求导，得：



这样，我们就得到了某个训练样本在当前模型的梯度值了，那么残差就是yi-pi。GBDT二分类的这个思想，其实和逻辑回归的思想一样，逻辑回归是用一个线性模型去拟合P(y=1|x)这个事件的对数几率，GBDT二分类也是如此， 用一系列的梯度提升树去拟合这个对数几率， 其分类模型可以表达为：



下面我们具体来看GBDT的生成过程，构建分类GBDT的步骤有两个：

1. 初始化GBDT

和回归问题一样，分类 GBDT 的初始状态也只有一个叶子节点，该节点为所有样本的初始预测值，如下：



上式里面，F代表GBDT模型，F0是模型的初始状态， 该式子的意思是找到一个，使所有样本的 Loss 最小，在这里及下文中，都表示节点的输出，即叶子节点， 且它是一个形式的值(回归值)，在初始状态，。

下面看例子，假设我们有下面3条样本：



我们希望构建 GBDT 分类树，它能通过「喜欢爆米花」、「年龄」和「颜色偏好」这3个特征来预测某一个样本是否喜欢看电影。我们把数据代入上面的公式中求Loss:



为了令其最小，我们求导，且让导数为0，则：



于是，就得到了初始值p=2/3，模型的初识状态F0(x)=0.69

<https://mp.weixin.qq.com/s/XP5z_BEeFr6oJp9VmVJRqQ>