GBDT + LR理论与实践

# 1、背景

CTR预估（Click-Through Rate Prediction）是互联网计算广告中的关键环节，预估准确性直接影响公司广告收入。CTR预估中用的最多的模型是LR（Logistic Regression），LR是广义线性模型，与传统线性模型相比，LR使用了Logit变换将函数值映射到0~1区间，映射后的函数值就是CTR的预估值。LR很容易并行化，处理上亿条训练样本不是问题，但线性模型学习能力有限，需要大量特征工程预先分析出有效的特征、特征组合，从而去间接增强LR 的非线性学习能力。

特征组合，是通过特征的一些线性叠加或者非线性叠加得到一个新的特征，可以有效的提高分类效果。常见的特征组合方式有笛卡尔积方式。

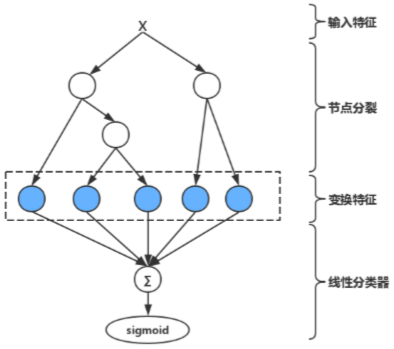
LR模型中的特征组合很关键，但又无法直接通过特征笛卡尔积解决，因为这种方式会带来很多冗余的特征，只能依靠人工经验，耗时耗力同时并不一定会带来效果提升。如何自动发现有效的特征、特征组合，弥补人工经验不足，缩短LR特征实验周期，是亟需解决的问题。

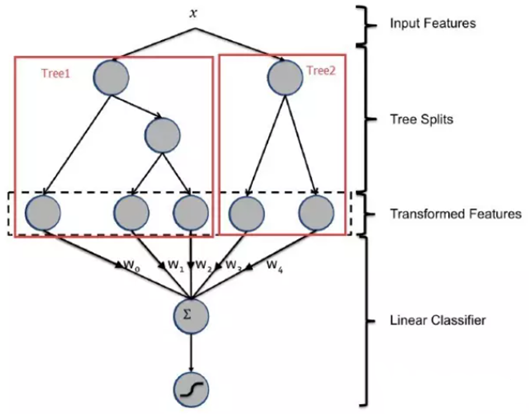
FM模型（对FM算法不熟悉的，**可以点击链接**）通过隐变量的方式，发现两两特征之间的组合关系，但这种特征组合仅限于两两特征之间，后来发展出来了使用深度神经网络去挖掘更高层次的特征组合关系。但其实在使用神经网络之前，GBDT也是一种经常用来发现特征组合的有效思路。

GBDT（Gradient Boost Decision Tree）是一种常用的非线性模型，它基于集成学习中的boosting思想，每次迭代都在减少残差的梯度方向新建立一颗决策树，迭代多少次就会生成多少颗决策树。GBDT的思想使其具有天然优势，可以发现多种有区分性的特征以及特征组合，决策树的路径可以直接作为LR输入特征使用，省去了人工寻找特征、特征组合的步骤。这种通过GBDT生成LR特征的方式（GBDT+LR），正是Facebook 2014年提出的CTR预估模型。随后Kaggle竞赛也有实践此思路，GBDT与LR融合开始引起了业界关注。

# 2、模型结构

GBDT+LR模型的结构图如下：





其中，GBDT部分相当于进行了特征选择、特征组合和特征变换。基于GBDT的特征转换：将单棵决策树的每个叶子节点对应LR一维特征，那么通过遍历树，就得到了该样本对应的所有LR特征，取值为样本落入在决策树的叶子节点的编号。例如，上图中提升树包含两棵子树，第一棵子树包含3个叶子节点，第二棵树包含2个叶子节点。对于一条输入样本x（包含多个特征），采用提升决策树（GBDT）进行训练，最终对于第一棵子树上，样本分裂之后落到第二个叶子节点，对于第二棵子树，样本落到了第1个叶子节点，那么通过特征进行转化之后就是[0,1,0,1,0]。由于树的每条路径，是通过最小化均方差等方法最终分割出来的有区分性路径，根据该路径得到的特征、特征组合都相对有区分性，效果理论上不会亚于人工经验的处理方式。

# 3、算法流程

GBDT+LR 由两部分组成，其中GBDT用来对训练集提取特征作为新的训练输入数据，LR作为新训练输入数据的分类器。

## 3.1训练

训练时，GBDT建树的过程相当于自动进行的特征组合和离散化，然后从根结点到叶子节点的这条路径就可以看成是不同特征进行的特征组合，用叶子节点可以唯一的表示这条路径，并作为一个离散特征传入LR进行二次训练。

与通常做法不同的是，当GBDT训练好做预测的时候，输出的并不是最终的二分类概率值，而是要把模型中的每棵树计算得到的预测概率值所属的叶子结点位置记为1，这样，就构造出了新的训练数据。

这里的思想与One-hot类似，事实上，在用GBDT构造新的训练数据时，采用的也正是One-hot方法。并且由于每一弱分类器有且只有一个叶子节点输出预测结果，所以在一个具有n个弱分类器、共计m个叶子结点的GBDT中，每一条训练数据都会被转换为1\*m维稀疏向量，且有n个元素为1，其余m-n 个元素全为0。

在把数据输入到Logistic Regression分类器中进行最终分类器的训练时，原始数据进行GBDT提取为新的数据这一操作之后，数据不仅变得稀疏，而且由于弱分类器个数，叶子结点个数的影响，可能会导致新的训练数据特征维度过大的问题，因此，在Logistic Regression这一层中，可使用正则化来减少过拟合的风险，在Facebook的论文中采用的是L1正则化。

## 3.2预测

预测时，会先走 GBDT 的每棵树，得到某个叶子节点对应的一个离散特征(即一组特征组合)，然后把该特征以one-hot形式传入LR进行线性加权预测。

注：通过GBDT进行特征组合之后得到的离散向量是和训练数据的原特征一块作为逻辑回归的输入，而不仅仅全是这种离散特征。

# 4、关键问题

笔者调研了Facebook、Kaggle竞赛关于GBDT建树的细节，发现两个关键点：采用ensemble决策树而非单棵树；建树采用GBDT而非RF（Random Forests）。解读如下：

**问题1：为什么建树采用ensemble决策树？**

一棵树的表达能力很弱，不足以表达多个有区分性的特征组合，多棵树的表达能力更强一些。GBDT每棵树都在学习前面棵树尚存的不足，迭代多少次就会生成多少颗树。按paper以及Kaggle竞赛中的GBDT+LR融合方式，多棵树正好满足LR每条训练样本可以通过GBDT映射成多个特征的需求。

**问题2：为什么建树采用GBDT而非RF？**

RF也是多棵树，但从效果上有实践证明不如GBDT。且GBDT前面的树，特征分裂主要体现对多数样本有区分度的特征；后面的树，主要体现的是经过前N颗树，残差仍然较大的少数样本。优先选用在整体上有区分度的特征，再选用针对少数样本有区分度的特征，思路更加合理，这应该也是用GBDT的原因。

**问题3：Facebook和Kaggle竞赛的思路是否能直接满足现在CTR预估场景呢？**

按照Facebook、Kaggle竞赛的思路，不加入广告侧的AD ID特征？但是现CTR预估中，AD ID类特征是很重要的特征，故建树时需要考虑AD ID。直接将AD ID加入到建树的feature中？但是AD ID过多，直接将AD ID作为feature进行建树不可行。下面将介绍针对现有CTR预估场景GBDT+LR的融合方案。

**问题4：点击率预估为什么选择LR？**

点击率预估模型涉及的训练样本一般是上亿级别，样本量大，LR模型训练速度较快，可以快速迭代。LR虽然是线性模型，但是在业界广泛使用。为什么呢？虽然模型本身表达能力差，但是可以通过特征工程不断减少问题的非线性结构。又由于模型计算复杂度低，可以吞吐超大规模的特征空间和样本集合，这样就为效果优化打开了空间。同时，LR可以学习id化特征，从而减少了特征工程的环节，可以提高特征的实时性。但正因为LR学习能力有限，此时特征工程尤其重要。

**问题5：选用GBDT模型有哪些优点？**

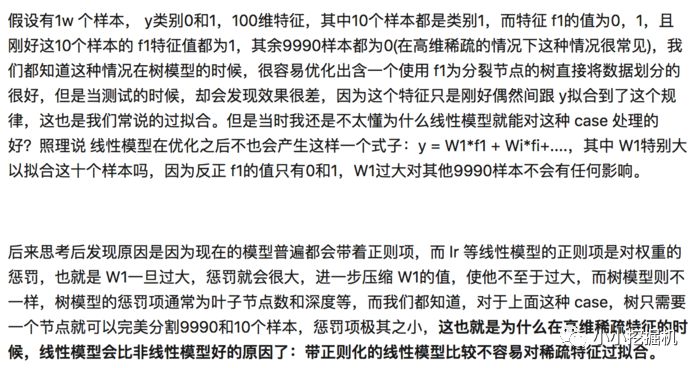
* 弱分类器要求不高，树的层数一般较小，小数据可用，扩展到大数据也能方便处理
* 需要更少的特征工程，比如不用做特征标准化
* 可以处理字段缺失的数据
* 可以自动组合多个特征并且不用关心特征间是否依赖，可以自动处理特征间的交互，不用担心数据是否线性可分
* 可以灵活处理多种类型的异构数据，这是决策树的天然特性
* 损失函数选择灵活，可以选择具有鲁棒性的损失函数，对异常值有一定的鲁棒性

**问题6：GBDT有哪些局限性？**

GBDT的缺点如下：

* GBDT在高维稀疏的数据集上，表现不如支持向量机或者神经网络。
* 训练过程基学习器需要串行训练，只能通过局部并行提高速度。

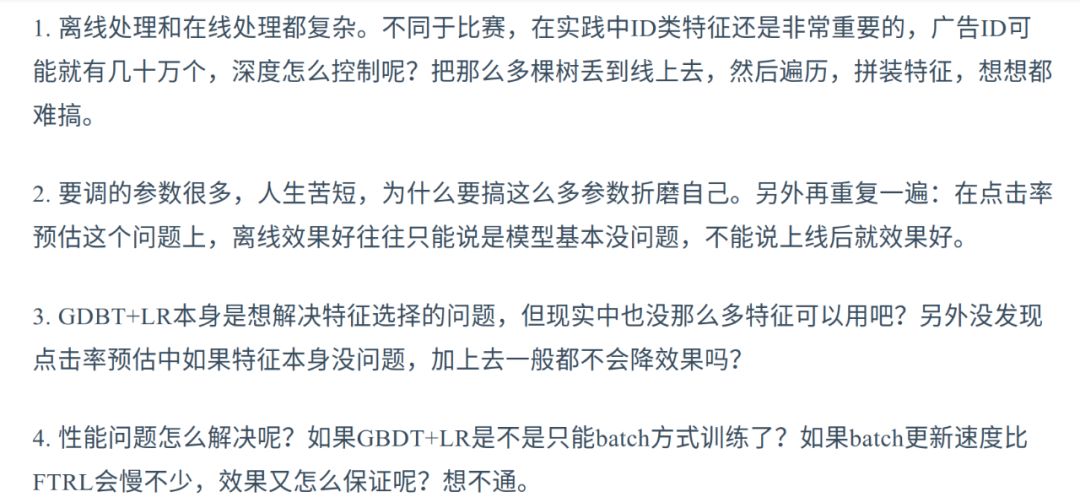
有人说，对于高维稀疏的特征，GBDT容易过拟合，表现不理想，甚至LR都比GBDT好，并给了一个例子：



GBDT只是对历史的一个记忆罢了，没有推广性，或者说泛化能力。但这并不是说对于大规模的离散特征，GBDT和LR的方案不再适用。

**问题7：GBDT+LR的缺点是什么？**

这里给出大佬的想法，不做过多讨论。



**问题8：GBDT和LR模型分别具有什么特点？**

* 模型刻画能力：GBDT能更好刻画头部样本，LR能更好刻画长尾
* GBDT偏variance，LR偏bias，所以需要注意控制GBDT的过拟合，而对于LR需要提升拟合能力

**问题9：GBDT作为一个编码器，起了那些作用呢？**

* 对连续特征进行了离散化
* 一棵树里面一条游走的路径就是一个特征组合
* GBDT树生长过程也是一个特征选择的过程
* 多棵树表示多种特征组合的方法，早期树产生的特征组合侧重区分度，后期则侧重于关注分类效果不好的样本

# 5、GBDT与LR融合方案

AD ID类特征在CTR预估中是非常重要的特征，直接将AD ID作为feature进行建树不可行，故考虑为每个AD ID建GBDT树。但互联网时代长尾数据现象非常显著，广告也存在长尾现象，为了提升广告整体投放效果，不得不考虑长尾广告。在GBDT建树方案中，对于曝光充分训练样本充足的广告，可以单独建树，发掘对单个广告有区分度的特征，但对于曝光不充分样本不充足的长尾广告，无法单独建树，需要一种方案来解决长尾广告的问题。

综合考虑方案如下，使用GBDT建两类树，非ID建一类树，ID建一类树。

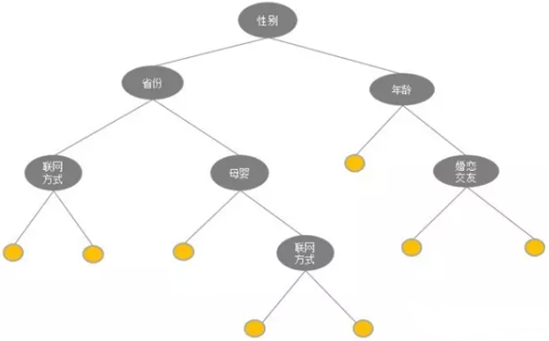
* 非ID类树：不以细粒度的ID建树，此类树作为base，即便曝光少的广告、广告主，仍可以通过此类树得到有区分性的特征、特征组合。
* ID类树：以细粒度的ID建一类树，用于发现曝光充分的ID对应有区分性的特征、特征组合。

**如何根据GBDT建的两类树，对原始特征进行映射？**

当一条样本x进来之后，遍历两类树到叶子节点，得到的特征作为LR的输入。当AD曝光不充分不足以训练树时，其它树恰好作为补充。

**通过GBDT 映射得到的特征空间维度如何？**

GBDT树有多少个叶子节点，通过GBDT得到的特征空间就有多大。如下图，一个叶子节点对应一种有区分性的特征、特征组合，对应LR的一维特征。这颗树有8个叶子节点，即对应LR 的8维特征。估算一下，通过GBDT转换得到的特征空间较低，Base树、ID树各N颗，特征空间维度最高为N+N\*广告数+N\*广告主数+ N\*广告类目数。其中广告数、广告主数、广告类目数都是有限的，同时参考Kaggle竞赛中树的数目N最多为30，则估算通过GBDT 映射得到的特征空间维度并不高，且并不是每个ID训练样本都足以训练多颗树，实际上通过GBDT 映射得到的特征空间维度更低。

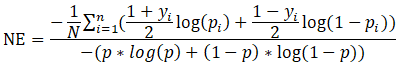


**如何使用GBDT 映射得到的特征？**

通过GBDT生成的特征，可直接作为LR的特征使用，省去人工处理分析特征的环节，LR的输入特征完全依赖于通过GBDT得到的特征。此思路已尝试，通过实验发现GBDT+LR在曝光充分的广告上确实有效果，但整体效果需要权衡优化各类树的使用。同时，也可考虑将GBDT生成特征与LR原有特征结合起来使用，待尝试。

# 6、评估

在Facebook的paper中，模型使用NE(Normalized Cross-Entropy)，进行评价，计算公式如下:



# 7、实践

见Github：<https://github.com/jpegbert/code_study/tree/master/GBDT_LR>

既有GBDT+LR又有GBDT+LR与XGB+LR和RF+LR的对比。欢迎Star

# 8、GBDT + LR 模型提升

现在，我们思考这样一个问题，Logistic Regression是一个线性分类器，也就是说会忽略掉特征与特征之间的关联信息，那么是否可以采用构建新的交叉特征这一特征组合方式从而提高模型的效果？其次，我们知道GBDT很有可能构造出的新训练数据是高维的稀疏矩阵，而Logistic Regression使用高维稀疏矩阵进行训练，会直接导致计算量过大，特征权值更新缓慢的问题。

针对上面可能出现的问题，使用FM算法代替LR，这样就解决了Logistic Regression的模型表达效果及高维稀疏矩阵的训练开销较大的问题。然而，这样就意味着可以高枕无忧了吗？当然不是，因为采用FM对本来已经是高维稀疏矩阵做完特征交叉后，新的特征维度会更加多，并且由于元素非0即1，新的特征数据可能也会更加稀疏，那么怎么办？

所以，我们需要再次回到GBDT构造新训练数据这里。当GBDT构造完新的训练样本后，我们要做的是对每一个特征做与输出之间的特征重要度评估并筛选出重要程度较高的部分特征，这样，GBDT构造的高维的稀疏矩阵就会减少一部分特征，也就是说得到的稀疏矩阵不再那么高维了。之后，对这些筛选后得到的重要度较高的特征再做FM算法构造交叉项，进而引入非线性特征，继而完成最终分类器的训练数据的构造及模型的训练。

# 9、总结

GBDT模型的特点，非常适合用来挖掘有效的特征、特征组合。业界不仅GBDT+LR融合有实践，GBDT+FM也有实践，2014 Kaggle CTR竞赛冠军就是使用GBDT+FM，可见，使用GBDT融合其它模型是非常值得尝试的思路。

参考：

<https://mp.weixin.qq.com/s/p3KT1AEyWf0IwF-bsfiITQ>

<https://mp.weixin.qq.com/s/04fNQRrFz3TrE7ObrNrdlw>

<https://mp.weixin.qq.com/s/9US7frWORbUg-MFJby773A>

<https://mp.weixin.qq.com/s/fXICNSXb7sN9O4kpW8di7Q>

<https://mp.weixin.qq.com/s/4JuU-kPVLRR74nW8YppGqA>

<https://mp.weixin.qq.com/s/RLH-VXJ4FK3PpJeqmj1n4w>

<https://mp.weixin.qq.com/s/LJzbcqV7I6xbYZ4ELztMCw>

<https://mp.weixin.qq.com/s/KzcQkaq6rnt0m64oq54ieg>

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

<https://zhuanlan.zhihu.com/p/152525097>

<https://mp.weixin.qq.com/s/_WUM9GaudslwPwLtqKJBfg>

<https://mp.weixin.qq.com/s/py9FJ7gDgm5H0sXu-k8_LA>