GBDT + LR理论与实践

在上一篇文章，提到了Facebook 2014年发表的一篇采用GBDT构建特征的论文：Practical lessons from predicting clicks on ads at facebook。为了深入学习GBDT，本文将重点分析这篇文章的思路，即CTR预估经典模型：GBDT+LR，一个曾风靡Kaggle、至今在工业界仍存有余温的传奇模型。同时采用scikit-learn里面的GBDT和LR来完成GBDT+LR的实验。

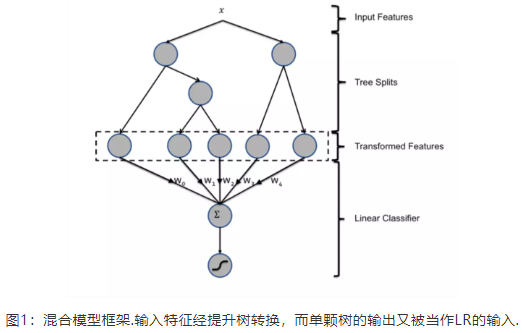
# 背景介绍

论文开篇介绍在计算广告领域，Facebook日活用户超过7.5亿，活跃广告超过1百万，这种数据规模对Facebook来说也是一大挑战。在这种情形下，Facebook是怎么做的呢？引入了一个组合决策树和LR的模型，该模型比单一的LR或GBDT的效果都要好，不仅将点击率提升了3%，还大大提升了整个系统的性能。除此之外，Facebook还在online learning、data freshness, 学习率等参数上进行了探索。

# 模型结构

Facebook论文的Section 1给出了一个重要结论：只要有正确的特征和正确的模型，其他因素对模型结果的影响就非常小。那么，正确的特征是什么呢？论文对比了两类特征，一类是用户或广告的历史信息特征(historical features)，另一类是contextual features(上下文特征)，相比之下historical features要优于contextual features。正确的模型指的boosted decision tree + LR，其中boosted decision tree又相当于对重要的特征做了feature selection。

在Section 3描述了论文的核心模型，整个hybird模型框架示意图如下：



对于线性分类器，有两种特征转换方式可以提升分类器的精度。

* 对于连续特征，可以对特征分bin，然后将bin的index作为类别特征，如此线性分类器就可以学习特征的非线性映射，这种方式里，学习有效的bin边界非常重要。
* 对于类别特征，可以采用笛卡尔积（Cartesian product）枚举出所有的二元特征组合。缺点是得到的特征会包含冗余特征。

为此，基于GBDT的特征转换方法诞生了。

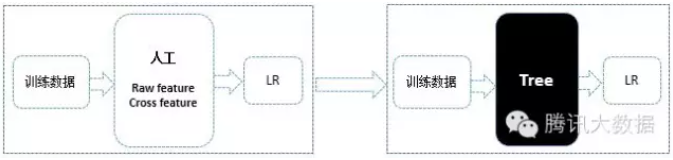
|  |
| --- |
| 基于GBDT的特征转换：将单棵决策树的结果看作是一个类别特征，取值为样本落入在决策树的叶子节点的编号。例如，图1中提升树包含两棵子树，第一棵子树包含3个叶子节点，第二棵树包含2个叶子节点。对于输入样本x（包含多个特征），采用提升决策树（GBDT）进行训练，最终对于第一棵子树上，样本分裂之后落到第二个叶子节点，对于第二棵子树，样本落到了第1个叶子节点，那么通过特征进行转化之后就是[0,1,0,1,0]。 |

==================================================================

CTR预估，广告点击率（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）[1]，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 [2]，映射后的函数值就是CTR的预估值。LR，逻辑回归模型，这种线性模型很容易并行化，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。

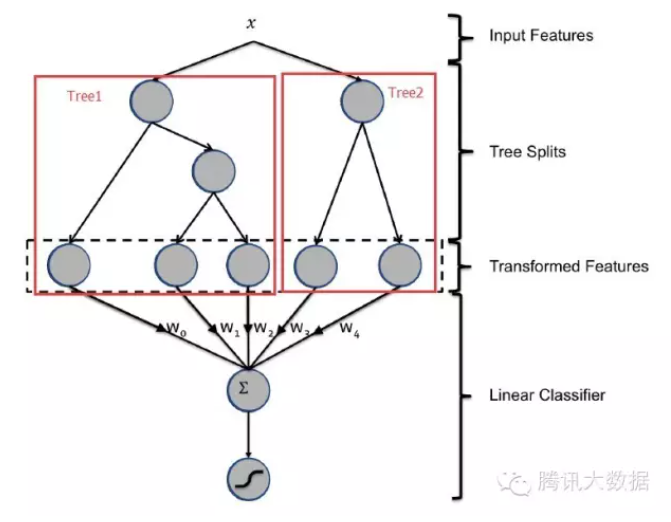
LR模型中的特征组合很关键，但又无法直接通过特征笛卡尔积 解决，只能依靠人工经验，耗时耗力同时并不一定会带来效果提升。如何自动发现有效的特征、特征组合，弥补人工经验不足，缩短LR特征实验周期，是亟需解决的问题。Facebook 2014年的文章介绍了通过GBDT （Gradient Boost Decision Tree）解决LR的特征组合问题[3]，随后Kaggle竞赛也有实践此思路[4][5]，GBDT与LR融合开始引起了业界关注。

GBDT（Gradient Boost Decision Tree）是一种常用的非线性模型[6][7][8][9]，它基于集成学习中的boosting思想[10]，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），业界已有实践（Facebook，Kaggle-2014），且效果不错，是非常值得尝试的思路。下图1为使用GBDT+LR前后的特征实验示意图，融合前人工寻找有区分性特征（raw feature）、特征组合（cross feature），融合后直接通过黑盒子（Tree模型GBDT）进行特征、特种组合的自动发现。



# GBDT与LR融合现状

GBDT与LR的融合方式，Facebook的paper有个例子如下图2所示，图中Tree1、Tree2为通过GBDT模型学出来的两颗树，x为一条输入样本，遍历两棵树后，x样本分别落到两颗树的叶子节点上，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。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。



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[11]。

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决策树而非单颗树；建树采用GBDT而非RF（Random Forests）。解读如下：

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。

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。

然而，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。下面第三部分将介绍针对现有CTR预估场景GBDT+LR的融合方案。

# GBDT与LR融合方案

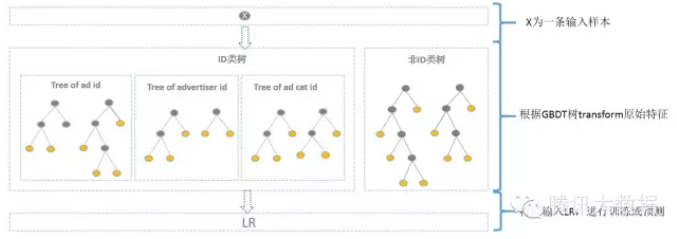
AD ID类特征在CTR预估中是非常重要的特征，直接将AD ID作为feature进行建树不可行，故考虑为每个AD ID建GBDT树。但互联网时代长尾数据现象非常显著，广告也存在长尾现象，为了提升广告整体投放效果，不得不考虑长尾广告[12]。在GBDT建树方案中，对于曝光充分训练样本充足的广告，可以单独建树，发掘对单个广告有区分度的特征，但对于曝光不充分样本不充足的长尾广告，无法单独建树，需要一种方案来解决长尾广告的问题。

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。

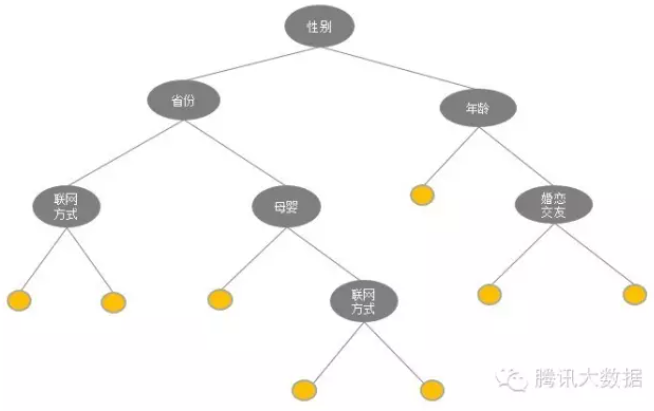
1）非ID类树：不以细粒度的ID建树，此类树作为base，即便曝光少的广告、广告主，仍可以通过此类树得到有区分性的特征、特征组合。

2）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？以如下图3为例，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4一颗树，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。

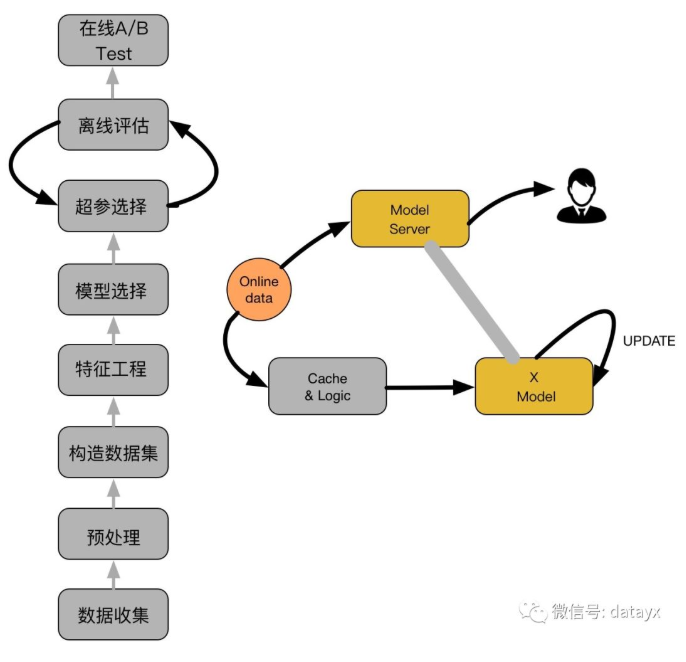


**如何使用GBDT 映射得到的特征？**

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。

# 4总结与展望

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，模型常采用速度较快的LR。但LR是线性模型，学习能力有限，此时特征工程尤其重要。现有的特征工程实验，主要集中在寻找到有区分度的特征、特征组合，折腾一圈未必会带来效果提升。GBDT算法的特点正好可以用来发掘有区分度的特征、特征组合，减少特征工程中人力成本，且业界现在已有实践，GBDT+LR、GBDT+FM等都是值得尝试的思路。不同场景，GBDT融合LR/FM的思路可能会略有不同，可以多种角度尝试。从知乎https://zhuanlan.zhihu.com/p/29053940上看到了一个关于CTR的流程，如下图所示：



如上图，主要包括两大部分：离线部分、在线部分，其中离线部分目标主要是训练出可用模型，而在线部分则考虑模型上线后，性能可能随时间而出现下降，弱出现这种情况，可选择使用Online-Learning来在线更新模型：

2.1 离线部分

数据收集：主要收集和业务相关的数据，通常会有专门的同事在app位置进行埋点，拿到业务数据

预处理：对埋点拿到的业务数据进行去脏去重；

构造数据集：经过预处理的业务数据，构造数据集，在切分训练、测试、验证集时应该合理根据业务逻辑来进行切分；

特征工程：对原始数据进行基本的特征处理，包括去除相关性大的特征，离散变量one-hot，连续特征离散化等等;

模型选择：选择合理的机器学习模型来完成相应工作，原则是先从简入深，先找到baseline，然后逐步优化；

超参选择：利用gridsearch、randomsearch或者hyperopt来进行超参选择，选择在离线数据集中性能最好的超参组合；

在线A/B Test：选择优化过后的模型和原先模型（如baseline）进行A/B Test，若性能有提升则替换原先模型；

2.2 在线部分

Cache &amp; Logic：设定简单过滤规则，过滤异常数据；

模型更新：当Cache &amp; Logic 收集到合适大小数据时，对模型进行pretrain+finetuning，若在测试集上比原始模型性能高，则更新model server的模型参数；

Model Server：接受数据请求，返回预测结果；

3. GBDT + LR 的结构

正如它的名字一样，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。

具体来讲，有以下几个步骤：

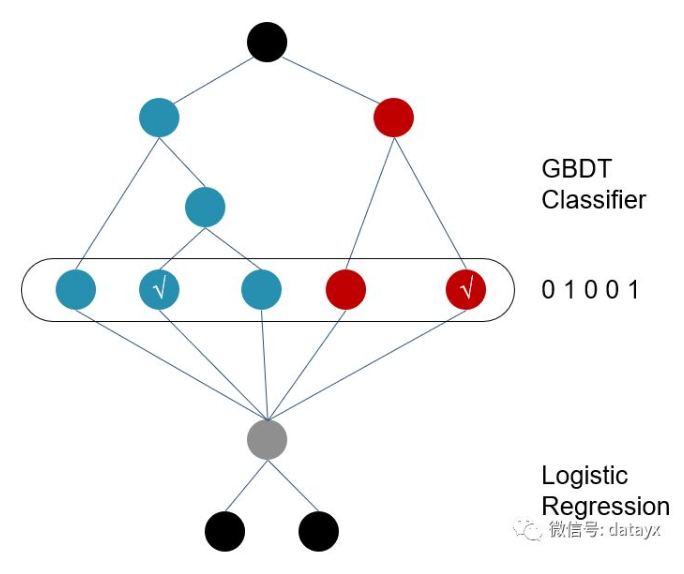
3.1 GBDT首先对原始训练数据做训练，得到一个二分类器，当然这里也需要利用网格搜索寻找最佳参数组合。

3.2 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。

举个例子，下图是一个GBDT+LR 模型结构，设GBDT有两个弱分类器，分别以蓝色和红色部分表示，其中蓝色弱分类器的叶子结点个数为3，红色弱分类器的叶子结点个数为2，并且蓝色弱分类器中对0-1 的预测结果落到了第二个叶子结点上，红色弱分类器中对0-1 的预测结果也落到了第二个叶子结点上。那么我们就记蓝色弱分类器的预测结果为[0 1 0]，红色弱分类器的预测结果为[0 1]，综合起来看，GBDT的输出为这些弱分类器的组合[0 1 0 0 1] ，或者一个稀疏向量（数组）。

这里的思想与One-hot独热编码类似，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。

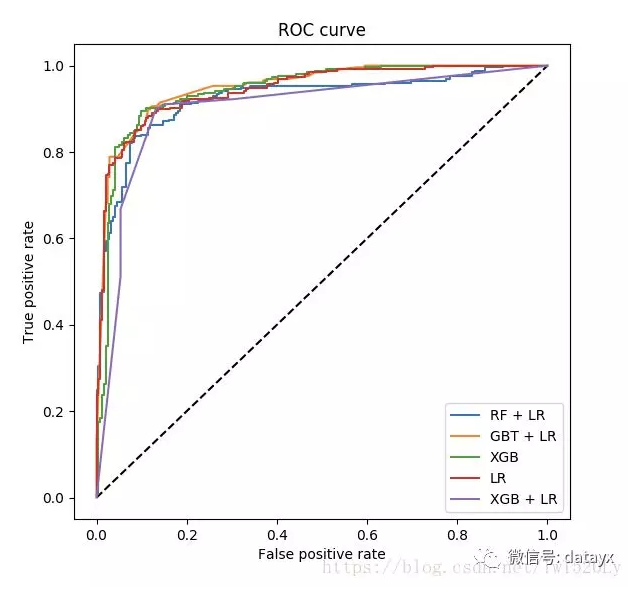
3.3 新的训练数据构造完成后，下一步就要与原始的训练数据中的label(输出)数据一并输入到Logistic Regression分类器中进行最终分类器的训练。思考一下，在对原始数据进行GBDT提取为新的数据这一操作之后，数据不仅变得稀疏，而且由于弱分类器个数，叶子结点个数的影响，可能会导致新的训练数据特征维度过大的问题，因此，在Logistic Regression这一层中，可使用正则化来减少过拟合的风险，在Facebook的论文中采用的是L1正则化。

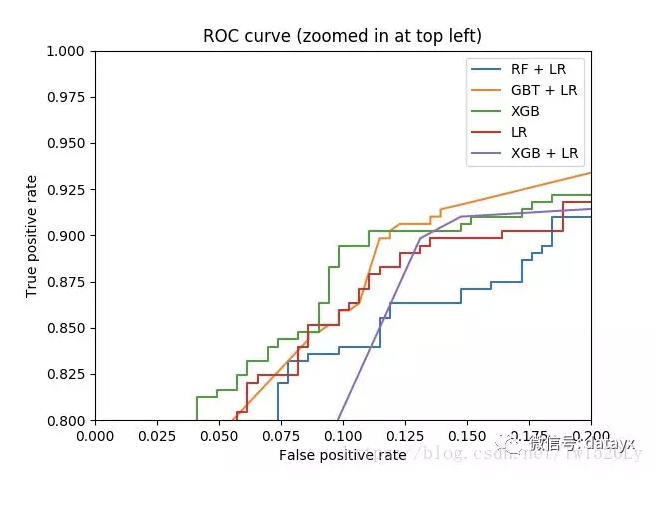


4. RF + LR ? Xgb + LR?

有心的同学应该会思考一个问题，既然GBDT可以做新训练样本的构造，那么其它基于树的模型，例如Random Forest以及Xgboost等是并不是也可以按类似的方式来构造新的训练样本呢？没错，所有这些基于树的模型都可以和Logistic Regression分类器组合。至于效果孰优孰劣，我个人觉得效果都还可以，但是之间没有可比性，因为超参数的不同会对模型评估产生较大的影响。下图是RF+LR、GBT+LR、Xgb、LR、Xgb+LR 模型效果对比图，然而这只能做个参考，因为模型超参数的值的选择这一前提条件都各不相同。

顺便来讲，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。





6. 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，我们已经在2.3小节中了解到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？

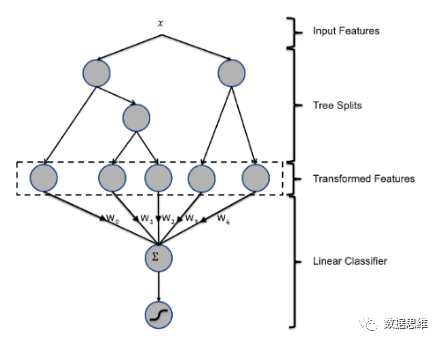
所以，我们需要再次回到GBDT构造新训练数据这里。当GBDT构造完新的训练样本后，我们要做的是对每一个特征做与输出之间的特征重要度评估并筛选出重要程度较高的部分特征，这样，GBDT构造的高维的稀疏矩阵就会减少一部分特征，也就是说得到的稀疏矩阵不再那么高维了。之后，对这些筛选后得到的重要度较高的特征再做FM算法构造交叉项，进而引入非线性特征，继而完成最终分类器的训练数据的构造及模型的训练。

=====================================

# 0.简介

CTR估计也就是广告点击率预估，计算广告训练与平滑思想说明了是用LR算法对于预测的有效性。LR（Logistic Regression）是广义线性模型，与传统线性模型相比，LR通过Logit变换将函数值映射到0~1区间，映射后的函数就是CTR的预估值。LR模型十分适合并行化，因此对于大数据的训练十分有效。但是对于线性模型而言，学习能力是有限的，因此需要大量的特征工程预先分析出有效的特征或者是特征组合，从而去间接的增强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。为了降低人工组合特征的工作量，FaceBook提出了一个自动特征提取的方式GBDT+LR。



GBDT是梯度提升决策树，首先会构造一个决策树，首先在已有的模型和实际样本输出的残差上再构造一颗决策树，不断地进行迭代。每一次迭代都会产生一个增益较大的分类特征，因此GBDT树有多少个叶子节点，得到的特征空间就有多大，并将该特征作为LR模型的输入。

1.核心问题

（1）建树采用ensemble决策树？

一棵树的区分性是具有一定的限制的，但是多棵树可以获取多个具有区分度的特征组合，而且GBDT的每一棵树都会学习前面的树的不足。

（2）建树算法为什么采用GBDT而不是RF?

对于GBDT而言，前面的树，特征分裂主要体现在对多数样本的具有区分度的特征；后面的树，主要体现的是经过前面n棵树，残差依然比较大的少数样本。优先选用在整体上具有区分度的特征，再选用针对少数样本有区分度的特征。

===============================================================

# 前言

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，模型常采用速度较快的LR(logistic regression)。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，他可以学习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[1]。但正因为LR学习能力有限，此时特征工程尤其重要。

在深度学习大行其道之前，一般采用人工或一些传统的方法，人工成本高就不说了，传统的方法像FM,FFM，只能挖掘两个特征间的特征交互关系，作用有限。GBDT是解决这个问题的一种不错方案。回顾我们上篇文章所讲的，GBDT有以下优点：

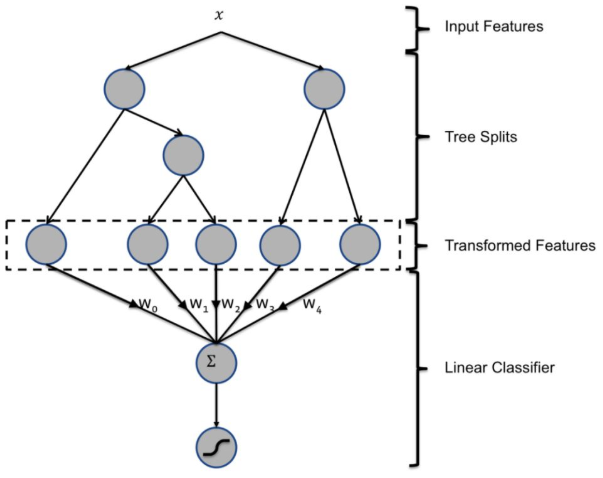
* 弱分类器要求不高，树的层数一般较小，小数据可用，扩展到大数据也能方便处理。
* 需要更少的特征工程，比如不用做特征标准化
* 可以处理字段缺失的数据
* 可以自动组合多个特征并且不用关心特征间是否依赖，可以自动处理特征间的交互，不用担心数据是否线性可分
* 可以灵活处理多种类型的异构数据，这是决策树的天然特性
* 损失函数选择灵活，可以选择具有鲁棒性的损失函数，对异常值有一定的鲁棒性

显而易见，GBDT对于处理特征有很多优点。而LR虽然是线性模型，但是Facebook探索出一种将GBDT和LR结合的方案，来预测广告的点击通过率(Click Trough Rate,CTR)的预测问题。结果显示融合方案比单个的GBDT或LR的性能高3%左右。

# GBDT+LR模型

首先要说明的是，对点击通过问题，要么点击要么不点击，因此y∈{1,\_1}，是个二分类的问题。因此LR+GBDT中的GBDT是上篇文章中所说的L2-Treeboost方案，上篇文章介绍过了这里不多说。

那么，GBDT与LR是如何结合的呢？看懂论文上的一张图就够了：



如上图所示，这个强学习器由两个弱学习器前向加和组成。假设x输入GBDT后，落到左边回归树的第一个节点，落到右边回归树的第一个节点。则GBDT对样本x的特征进行工程处理得到的转换特征，就可以表示为：(1,0,0)串联（1,0）==》(1,0,0,1,0)。然后将特征输入LR即可进行分类。

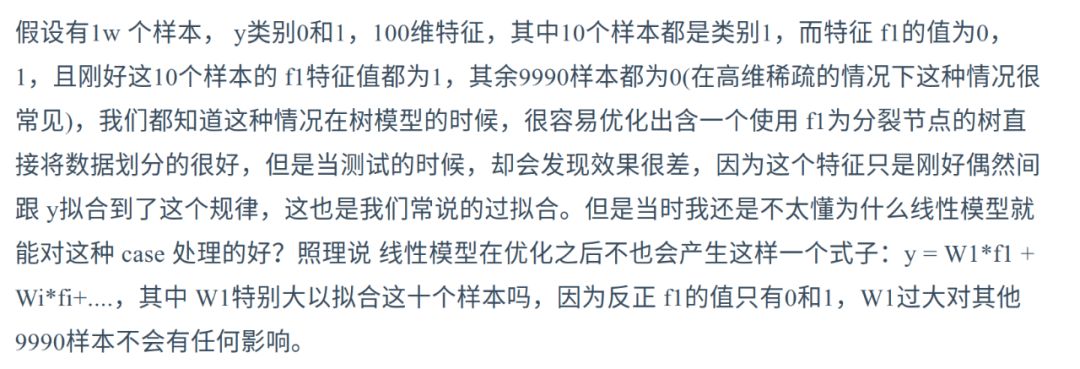
# 反思与总结

## GBDT局限性

GBDT+LR这个模型还是有一些局限性的。首先看GBDT的缺点，上篇文章中我们提到过：GBDT有一些缺点：

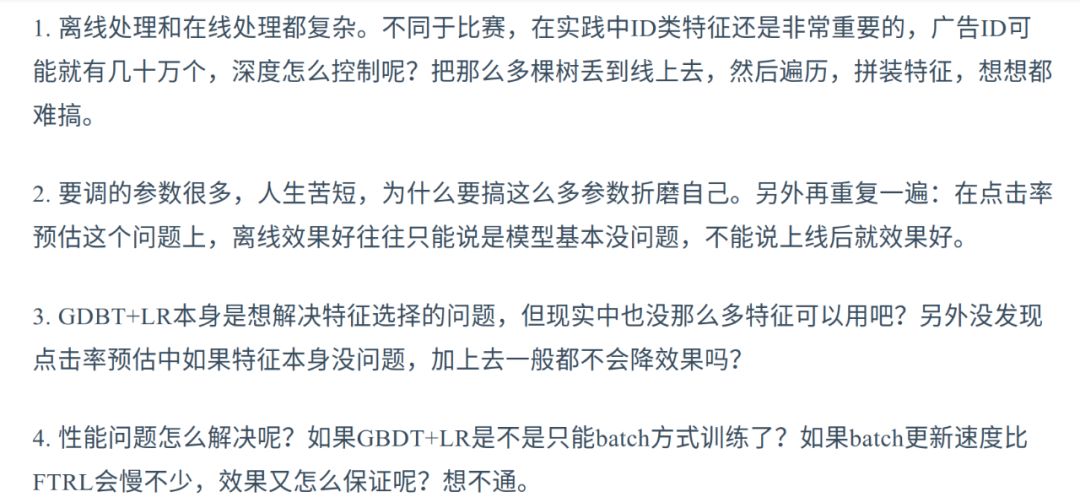
* GBDT在高维稀疏的数据集上，表现不如支持向量机或者神经网络[2]。
* 训练过程基学习器需要串行训练，只能通过局部并行提高速度。

[3]中凯菜大佬指出，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+LR缺点

关于GBDT+LR的缺点，[4]中屈伟大佬给出了一些看法：



我们分析模型的优缺点，是为了从中借鉴，也是为了更好地使用模型。没有哪个模型可以解决所有问题，不同模型都有自己的优点和缺点。GBDT+LR这种组合，有局限性，但也提供了一种不错的思路。实际业务中还是根据不同情况选择最合适的模型，扬长补短。

==========================

先比较下GBDT和LR本身的一些特点：

* 模型刻画能力：GBDT能更好刻画头部样本，LR能更好刻画长尾
* Variance vs. Bias：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

那么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劲就是一个特征组合
* 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

===========

# 一、GBDT+LR简介

协同过滤和矩阵分解存在的劣势就是仅利用了用户与物品相互行为信息进行推荐，忽视了用户自身特征，物品自身特征以及上下文信息等，导致生成的结果往往会比较片面。而这次介绍的这个模型是2014年由Facebook提出的GBDT+LR模型，该模型利用GBDT自动进行特征筛选和组合，进而生成新的离散特征向量，再把该特征向量当做LR模型的输入，来产生最后的预测结果，该模型能够综合利用用户、物品和上下文等多种不同的特征，生成较为全面的推荐结果，在CTR点击率预估场景下使用较为广泛。

下面首先会介绍逻辑回归和GBDT模型各自的原理及优缺点，然后介绍GBDT+LR模型的工作原理和细节。

逻辑回归模型非常重要，在推荐领域里面，相比于传统的协同过滤，逻辑回归模型能够综合利用用户、物品、上下文等多种不同的特征生成较为“全面”的推荐结果，关于逻辑回归的更多细节， 可以参考下面给出的链接，这里只介绍比较重要的一些细节和在推荐中的应用。

相比于协同过滤和矩阵分解利用用户的物品“相似度”进行推荐，逻辑回归模型将问题看成了一个分类问题，通过预测正样本的概率对物品进行排序。这里的正样本可以是用户“点击”了某个商品或者“观看”了某个视频，均是推荐系统希望用户产生的“正反馈”行为， 因此逻辑回归模型将推荐问题转化成了一个点击率预估问题。而点击率预测就是一个典型的二分类，正好适合逻辑回归进行处理，那么逻辑回归是如何做推荐的呢？过程如下：

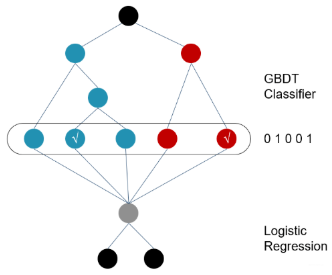
1. 将用户年龄、性别、物品属性、物品描述、当前时间、当前地点等特征转成数值型向量
2. 确定逻辑回归的优化目标，比如把点击率预测转换成二分类问题，这样就可以得到分类问题常用的损失作为目标，训练模型
3. 在预测的时候，将特征向量输入模型产生预测，得到用户“点击”物品的概率
4. 利用点击概率对候选物品排序，得到推荐列表

LR 需要进行**人工特征组合**，这就需要开发者有非常丰富的领域经验，才能不走弯路。这样的模型迁移起来比较困难，换一个领域又需要重新进行大量的特征工程。所以如何**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，而GBDT模型，正好可以自动发现特征并进行有效组合。

# GBDT+LR模型

2014年，Facebook提出了一种利用GBDT自动进行特征筛选和组合，进而生成新的离散特征向量，再把该特征向量当做LR模型的输入，来产生最后的预测结果，这就是著名的GBDT+LR模型了。GBDT+LR 使用最广泛的场景是CTR点击率预估，即预测当给用户推送的广告会不会被用户点击。

有了上面的铺垫，这个模型解释起来就比较容易了，模型的总体结构长下面这样：



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。

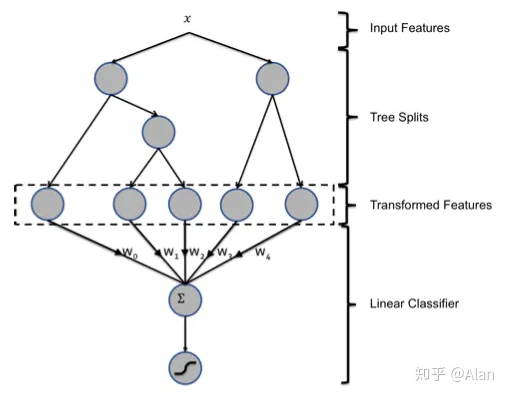
比如上图中，有两棵树，x为一条输入样本，遍历两棵树后，x样本分别落到两颗树的叶子节点上，每个叶子节点对应LR一维特征，那么通过遍历树，就得到了该样本对应的所有LR特征。构造的新特征向量是取值0/1的。比如左树有三个叶子节点，右树有两个叶子节点，最终的特征即为五维的向量。对于输入x，假设他落在左树第二个节点，编码[0,1,0]，落在右树第二个节点则编码[0,1]，所以整体的编码为[0,1,0,0,1]，这类编码作为特征，输入到线性分类模型（LR or FM）中进行分类。

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。

这个方案应该比较简单了，下面有几个关键的点我们需要了解：

1. 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
2. 建树的时候用ensemble建树的原因就是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棵树。
3. 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。
4. 在CRT预估中，GBDT一般会建立两类树(非ID特征建一类，ID类特征建一类)，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。

* 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。



# 二、LR+GBDT的特点与方案

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。

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。

然而，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。下面第三部分将介绍针对现有CTR预估场景GBDT+LR的融合方案。

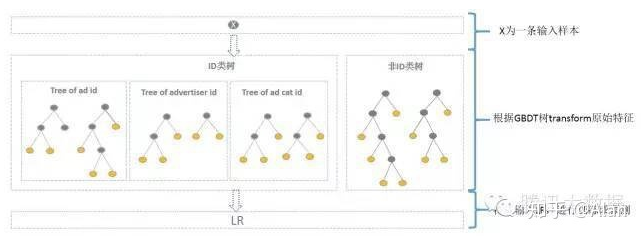
3、GBDT与LR融合方案

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。

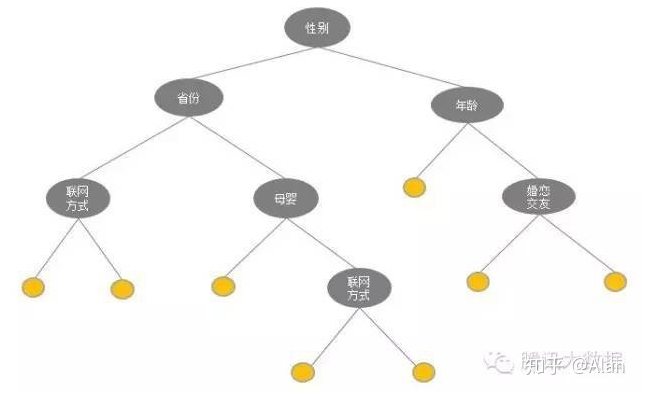
非ID类树(GBDT的输入特征不含id类)：不以细粒度的ID建树，此类树作为base，即便曝光少的广告、广告主，仍可以通过此类树得到有区分性的特征、特征组合。

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？以如下图3为例，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4一颗树，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（黄色点），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。此证明，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。

================================================================

GBDT和LR的融合在广告点击率预估中算是发展比较早的算法，为什么会在这里写这么一篇呢？本来想尝试写一下阿里的深度兴趣网络(Deep Interest Network)，发现阿里之前还有一个算法MLR，然后去查找相关的资料，里面提及了树模型也就是GBDT+LR方案的缺点，恰好之前也不太清楚GBDT+LR到底是怎么做的，所以今天我们先来了解一下GBDT和LR的融合方案。

# 1、背景

在CTR预估问题的发展初期，使用最多的方法就是逻辑回归(LR)，LR使用了Sigmoid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属于线性模型，容易并行化，可以轻松处理上亿条数据，但是学习能力十分有限，需要大量的特征工程来增加模型的学习能力。但大量的特征工程耗时耗力同时并不一定会带来效果提升。因此，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。

Facebook 2014年的文章介绍了通过GBDT解决LR的特征组合问题，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。

在介绍这个模型之前，我们先来介绍两个问题：

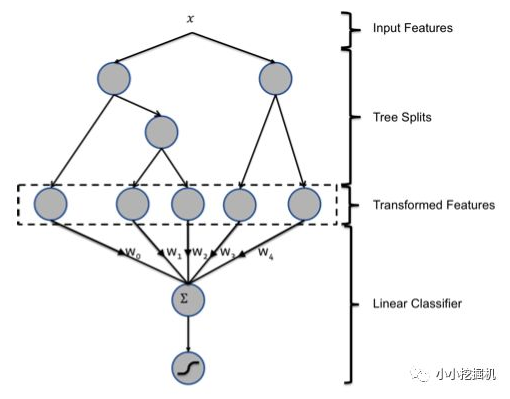
1）为什么要使用集成的决策树模型，而不是单棵的决策树模型：一棵树的表达能力很弱，不足以表达多个有区分性的特征组合，多棵树的表达能力更强一些。可以更好的发现有效的特征和特征组合

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。

了解了为什么要用GBDT，我们就来看看到底二者是怎么融合的吧！

# 2、GBDT和LR的融合方案

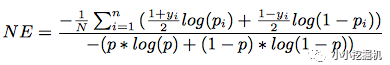
GBDT和LR的融合方案，FaceBook的paper中有个例子：



图中共有两棵树，x为一条输入样本，遍历两棵树后，x样本分别落到两颗树的叶子节点上，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。构造的新特征向量是取值0/1的。举例来说：上图有两棵树，左树有三个叶子节点，右树有两个叶子节点，最终的特征即为五维的向量。对于输入x，假设他落在左树第一个节点，编码[1,0,0]，落在右树第二个节点则编码[0,1]，所以整体的编码为[1,0,0,0,1]，这类编码作为特征，输入到LR中进行分类。

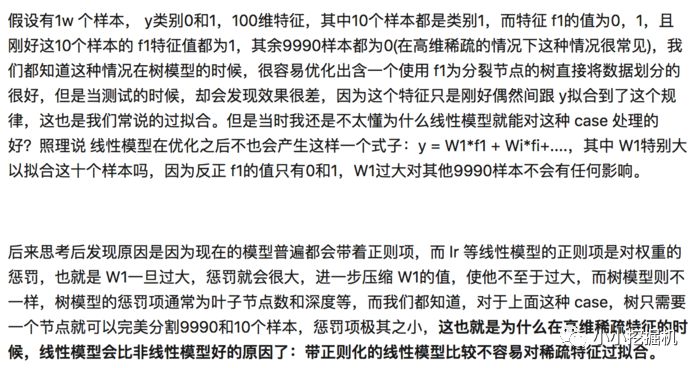
# 效果评价

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



# 4、反思

现在的GBDT和LR的融合方案真的适合现在的大多数业务数据么？现在的业务数据是什么？是大量离散特征导致的高维度离散数据。而树模型对这样的离散特征，是不能很好处理的，要说为什么，因为这容易导致过拟合。下面的一段话来自知乎：



用盖坤的话说，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。

==================================================================

# 1、算法背景

2014年Facebook发表了一篇介绍将GBDT+LR模型用于其广告推荐系统的论文，之后，无论是Kaggle竞赛还是淘宝商品推荐，都有借鉴该论文中的GBDT+LR模型组合思想，即通过GBDT来发掘有区分度的特征和组合特征，来代替人工组合特征。

对于支撑互联网半壁江山的广告收入，推荐系统和CTR预估于其技术框架中占据重要地位，而LR模型则是其中最为常用的模型。

LR模型有以下特点：

* 计算复杂度低；
* 易于并行化处理；
* 易于得到离散化目标值0或1，利用sigmoid函数将传统线性模型的输出值映射到(0,1)区间；
* 学习能力限于线性特征，需要提前进行大量的特征工程得到有效的特征及特征组合。

输入LR模型的特征很重要，但是特征组合不能直接通过特征笛卡尔积获取，只能依靠人工经验。故而如何自动化进行特征工程，规范化LR模型使用流程是一个值得研究的问题。

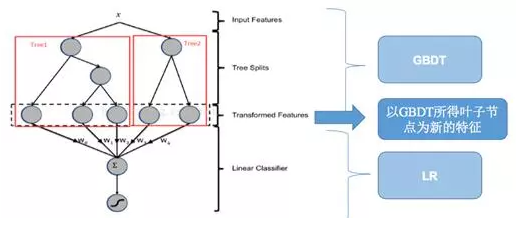
GBDT作为一种常用的树模型，可天然地对原始特征进行特征划分、特征组合和特征选择，并得到高阶特征属性和非线性映射。从而可将GBDT模型抽象为一个特征处理器，通过GBDT分析原始特征获取到更利于LR分析的新特征。这也正是GBDT+LR模型的核心思想——利用GBDT构造的新特征来训练LR模型。

# 2、算法原理及实现

前面简单介绍了GBDT+LR模型的产生背景和核心思想，接下来将会更为详细地描述GBDT+LR模型的算法组合思想和简单实现流程。

## 2.1、算法组合——stacking

stacking方法有些类似于农业中的嫁接，通过stacking方法组合的模型亦类似于嫁接植物，例如，解决了人类吃饭问题的杂交水稻。



如上图所示，GBDT算法的图示部分形如一棵倒过来的树，其根部即代表训练GBDT算法的原始数据集，经过树算法对原始数据的切分，可得到代表不同新特征的叶子节点。

再将GBDT所得的叶子节点输入LR算法，经过线性分析和sigmoid映射，即可得到模型分类结果。

以上的模型组合方式就是stacking方法，即将学习层模型对原始数据所得的预测结果作为新的特征集，并输入给输出层模型得到分类结果。Facebook论文中的GBDT+LR模型就采用了GBDT算法作为学习层，以LR算法为输出层。

## 2.2、算法流程& 代码简单实现



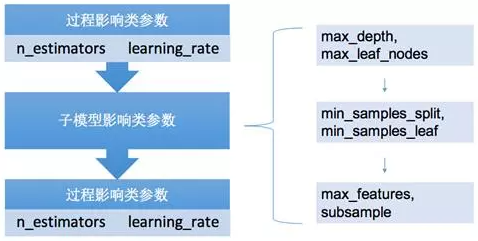
2.3、调参方法简述

构建了模型框架后，模型中的函数参数调整也是必不可少的。对模型参数的适当调整，往往可以有效提升模型的效果。

由于GBDT+LR模型无法整体使用GridSearchCV函数，所以调参时

使用sklearn.cross\_validation中的StratifiedKFold方法，将数据集进行k折交叉切分，然后以auc值为模型评估指标，对混合模型进行调参。

调参时的重点为GradientBoostingClassifier函数，可用如下图所示的调参顺序进行调参。



其中，n\_estimators和learning\_rate应该联合调参。