CTR 预估

梁鹏

1. 项目简介

广告点击率（Click-Through Rate Prediction, CTR）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。本项目我们对 Avazu 提供的 Kaggle 竞赛数据进行移动CTR 预估。竞赛数据包括 11 天的数据，10 天为训练数据 train，1 天为测试数据 test。项目周期为 4 周时间，整个任务由团队共同完成。Kaggle 竞赛网页为：https://www.kaggle.com/c/avazu-ctr-prediction。

2. 项目过程

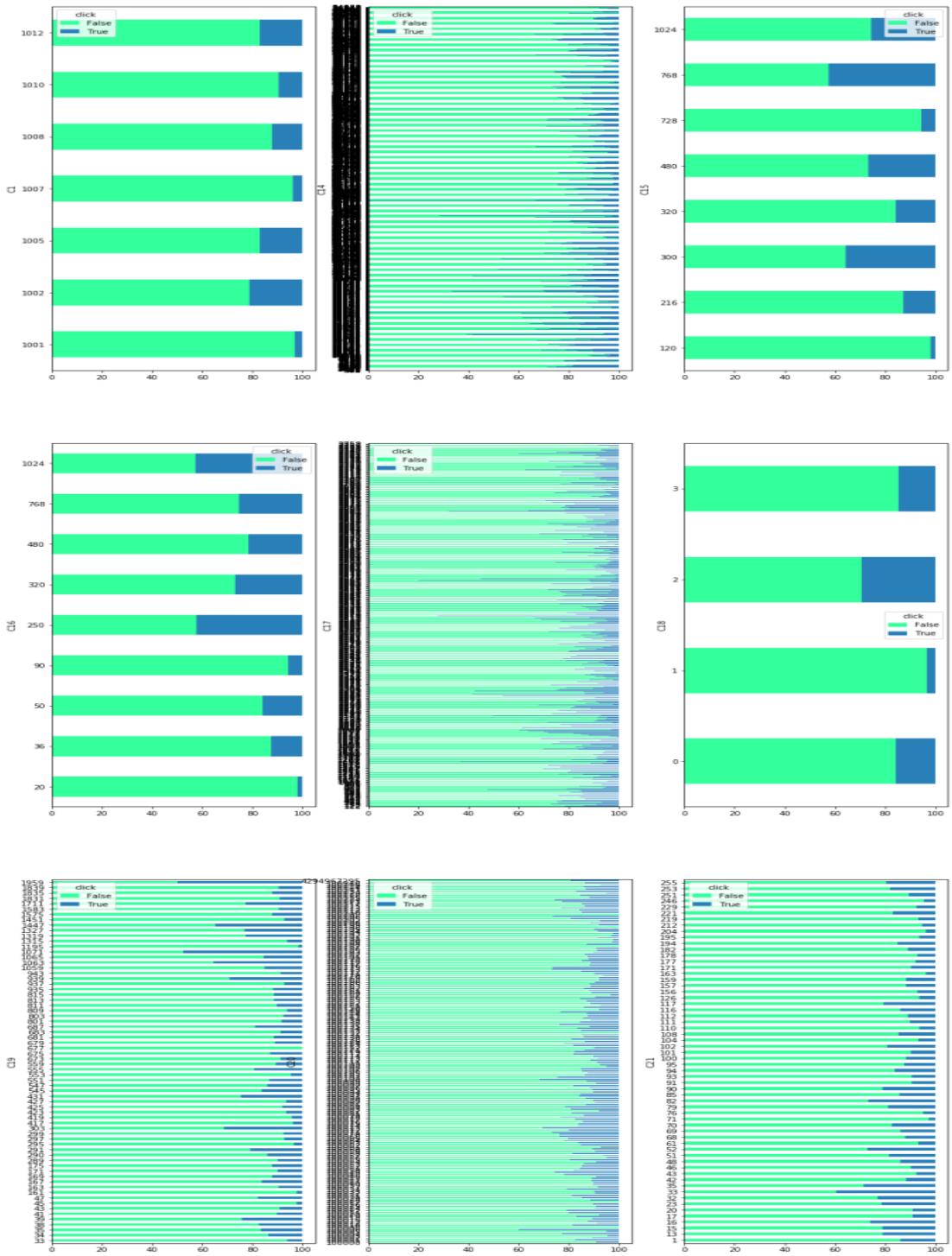
2.1 数据探索

数据特征包括以下字段：id，click，hour，C1，banner_pos，site_id，site_domain，site_category，app_id，app_domain，app_category，device_id，device_ip，device_model，device_type，device_conn_type，C14-C21。通过字段的文件说明、取值类型、取值范围，对特征意义作出如下推测：

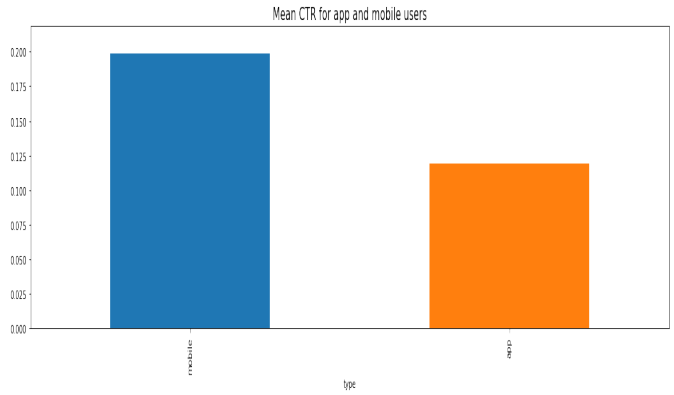
特征	取值	含义（推测）
id	不重复的 float64	样本 id
click	0、1	是否点击
hour	YYMMDDHH 时间特征	样本的采集时间
C1、C14-C21	离散数值	类别特征
banner_pos	离散数值	广告位置
site_id	字符串	网站 ID
site_domain	字符串	网站领域
site_category	字符串	网站类型
app_id	字符串	App ID
app_domain	字符串	App 领域
app_category	字符串	App 类型
device_id	字符串	设备 ID
device_ip	字符串	设备 IP
device_model	字符串	设备型号
device_type	离散数值	设备类型
device_conn_type	离散数值	设备接入类型

从数据来看，site 和 app 的 category、domai、id 特征种类依次增多，推测 category 是更上层的分类，意味着 category 是最容易相同的。

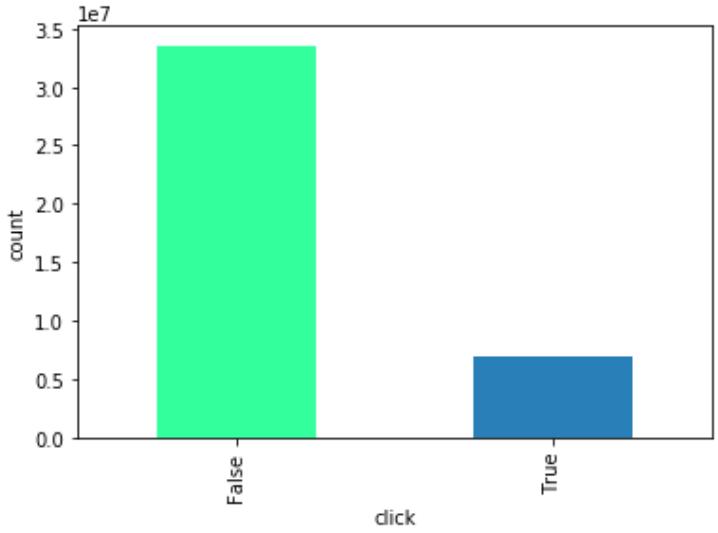
作出每个特征的分布情况，由于特征较多，下面不一一展示每个特征的分布，主要展示跟结论有关的分布。



匿名类别特征 C14、C17、C20 和 C21 具有大量的属性，并且不同的属性下点击率差别较大，因此可能对正标签具有更大的影响，需要重点关注。



site_id 为 85f751fd 占总数据量的 33%,app_id 为 ecad2386 占总数据量的 67%。意味着两者不会同时出现，并且猜测 site_id 为 85f751fd 代表 app 用户, app_id 为 ecad2386 代表 site 用户。并且可以看到 app 用户和 site 用户的平均 CTR 几乎相差 2 倍，意味着两种用户 ctr 的模型参数可能有所不同，考虑使用不同的处理方式。



点击的数量和未点击的数量的比例为 2:8，可以考虑尝试使用下采样。

数据探索阶段主要对数据进行一个直观的了解，对各特征的含义、分布、重要程度有一个认识，并考虑对数据进行以下几个方面的特征工程：根据 site_id 和 app_id 的媒介差别，对两种媒介的样本进行分开处理；利用 device_id 构建用户信息；对时间信息进行处理。

2.2 特征工程

由于 CTR 预估数据集较大，制约了模型的复杂程度，因此模型的选择往往以线性模型为主，而线性模型对特征工程提出了较高的要求：所构建的特征需要

满足和结果较强的线性关系。

在数据探索的基础上，我们从以下几个方面进行特征工程：

（a）定位用户

定位用户的目的在于能够根据模型中每个用户的其它特征找到该用户的点击率喜好，从信息论的角度出发该处理应该能够降低 logloss。

对于 device_id、device_ip 和 device_model，可以通过(device_id)，(device_ip+device_model)来区别用户。其中，当 device_id 为“a99f214a”时，这一属性可能为其它某种汇总数据的接入，无法代表用户，通过结合 device_ip 和 device_model 作为 user_id；否则，取 device_id 作为 user_id。

（b）定位媒介

定位媒介的作用同定位用户。

app_id 为“ecad2386”和 site_id 为“85f751fd”两者不会同时出现，并且 app_id 和 site_id 中两者必出现一个，并且两者的 CTR 平均值相差一倍以上，因此可以通过两者区分不同的广告浏览媒介(site or app)。可以在特征工程时将 app 和 site 进行融合以减小内存开销，或者在训练预测时直接进行不同的处理。

（c）统计特征

将部分特征属性的统计值作为新的特征往往可以取得较好的效果，每个统计特征一般可以认为是该属性的一个权值，意义比较直观。考虑新增以下统计特征：

每个广告 ID 在每天被不同用户浏览的次数：number_ad_day。

每个广告 ID 在每小时被不同用户浏览的次数：number_ad_hour。

每类广告 ID 在每天被不同用户浏览的次数：number_ad_category_day。

每类广告 ID 在每小时被不同用户浏览的次数：number_ad_category_hour。

每天不同用户浏览广告的次数：statistics_id。

每天不同用户通过不同方式(App or Site)浏览的次数：number_portal_day。

用户出现的天数：user_date

（d）时间特征

时间特征可以认为是用户特征中较为高级的一种特征。其代表的是用户在时间上进行活动的规律。

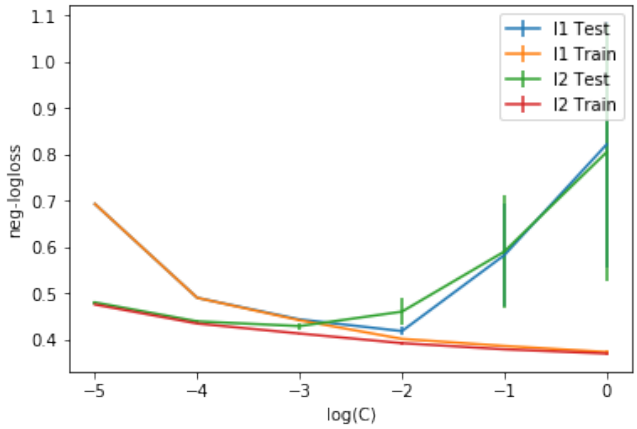
考虑对时间特征进行以下处理：

将原特征 hour 的格式 YYMMDDHH 改为 HH 格式，训练的目的是为了预测，要使训练集和测试机的时间特征具有共同点。

新增 interval 特征，表示距离最后一次访问的时间间隔。

2.3 基本模型

因为是二分类模型，输出的是概率，因此采用 logistics 回归。模型训练后，最低的 logloss 得分为 0.4。最佳超参数为{C: 0.1, penalty: l1}。



备注：在运行基本模型的阶段，有些特征工程的处理还没有做，是后续添加进去的，如果在这个阶段做完全部的特征工程，效果应该能再有所提升。

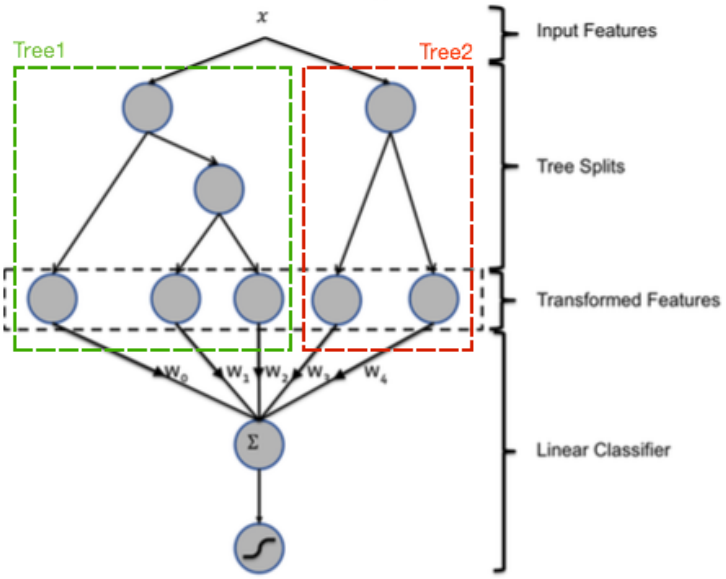
可以看出，跟 kaggle 上的模型比起来，基本模型的效果很一般，分析原因为点击率更多的是和用户特征、广告特征的交叉特征线性相关，因此可以直接尝试高级模型。

2.4 高级模型

（a）GBDT+FFM

这种模型的选择从理解的角度来说是首选。FFM 是二阶多项式，对于点击率预估是非常合适的单模型，GBDT 可以用来构造二阶多项式以外的线性特征。

如下图所示，利用 GBDT 构造新特征：模型的输入为新增的 8 个数值特征，输出为树的索引。构造的新特征向量是取值 0/1 的，向量的每个元素对应于 GBDT 模型中树的叶子结点。当一个样本点通过某棵树最终落在这棵树的一个叶子结点上，那么在新特征向量中这个叶子结点对应的元素值为 1，而这棵树的其他叶子结点对应的元素值为 0。新特征向量的长度等于 GBDT 模型里所有树包含的叶子结点数之和。经过训练，最优的树的个数为 20，每棵树的深度为 5。



GBDT 生成的 20 个特征、8 个新增的特征、21 个原始特征共同组成 FFM 模型的输入，FFM 模型调用 libffm。在数据探索的过程中，'site_id'=='85f751fd'以及'app_id'=='ecad2386'的平均 CTR 几乎相差 2 倍，在此使用 FFM 对二者分开进行处理。

对于'app_id'=='ecad2386'样本：FFM 的参数为，隐变量个数 K=8，学习率 l=0.00002，迭代次数 t=4；对于'site_id'=='85f751fd'样本：FFM 的参数为，隐变量个数 K=8，学习率 l=0.00003，迭代次数 t=10。

GBDT+FFM 模型结果

从提交的分数来看，这应该是一个不错的成绩，GBDT+FFM 达到了令人满意的效果。

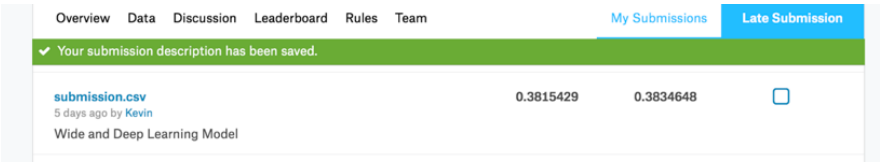
（b）Wide & Deep 模型

核心思想是结合线性模型的记忆能力（memorization，Wide 端）和 DNN 模型的泛化能力（generalization，Deep 端），在训练过程中同时优化 2 个模型的参数，从而达到整体模型的预测能力最优。Wide 端对应的是线性模型，输入特征可以是连续特征，也可以是稀疏的离散特征，离散特征之间进行交叉后可以构成更高维的离散特征。Deep 端对应的是 DNN 模型，每个特征对应一个低维的实

数向量，我们称之为特征的 embedding。DNN 模型通过反向传播调整隐藏层的权重，并且更新特征的 embedding。

线性模型参数（采用默认值）:linear_optimizer，线性模型的优化函数，定义权重的梯度更新算法，默认采用 FTRL; join_linear_weights，经过自己线下的对比试验，对模型的预测能力似乎没有太大影响，对训练速度有所提升，最终训练模型时我们保持了默认值。

DNN 模型参数：dnn_feature_columns，DNN 模型的输入特征；dnn_optimizer，DNN 模型的优化函数，定义权重的梯度更新算法，采用 Adagrad；dnn_hidden_units，每个隐藏层的神经元数目，每个隐层的数量设置为 100；dnn_activation_fn，隐藏层的激活函数，默认采用 RELU；dnn_dropout，模型训练中隐藏层单元的 drop_out 比例，设置为 0.8。



Wide & Deep 模型结果

3. 心得体会

3.1 困难与收获

在项目的推进过程中，遇到了很多困难，其中最主要的困难是数据集的规模过大和本地编码配置不高，直接导致在课程学习的部分很多 demo 拿过来没法用，并且训练的时间相对较长，导致后期优化工作进展很慢。

不过，有困难才会有收获，克服困难的过程便是披荆斩棘的过程。在参阅其他高手的代码过程中，能够学习到别人特征工程的思路是很有意义的，其次在利用别人的框架的过程中，越发体会到数学和编码的重要性，这也是我们以后业余时间用来学习的方向。

3.2 改进设想

总的来说，GBDT+FFM 的模型已经能够表征数据之间的关系了，模型改进可以从两方面入手，第一是可以构建更为复杂的时间特征，毕竟每个用户的点击时间在规律中存在着随机，应该在训练中消除随机点击带来的影响，第二是从 FFM+GBDT 所不能表征的因素出发，可以考虑构建高阶特征或者非线性特征，

或者考虑融合一些非线性模型，比如核化的 SVM，应该能带来效果的提升，但应该注意过拟合。

3.3 致谢

最后，我们小组在此感谢卿老师、智老师以及各位助教，感谢你们半年来悉心的指导和付出，感谢 CSDN 提供的这个学习平台，在这里我们遇到了良师益友，大家有着丰富的学识、求实的态度、勤奋的精神、对知识的渴望，这些将成为我们乘风破浪、披荆斩棘的力量源泉！