代码运行说明

文件名	说明
cache/	缓存中间文件夹
datasets/	存放原始数据文件夹
feat/	特征保存文件夹
model/	模型保存文件夹
ConcatFile.py	拼接2/3/4月数据并产生窗口字段
cv_params.py	调参
extract_feat_to_skuModel.py	商品模型特征提取
extract_feat_to_userModel.py	用户模型特征提取
features_generator.py	特征生成函数文件
rule.py	规则提取
sku_model.py	商品模型特征文件
TimeDecay.py	时间衰减权重文件
tools.py	评测函数文件
user_model_final.py	用户模型特征文件
xgb_skuModel.py	商品模型训练文件
xgb_userModel.py	用户模型训练文件
GenerateResult.py	从训练好的模型快速产生结果文件
requirements.txt	python依赖库
generateFeature_and_trainModel.bat	从原始数据提取特征并训练模型产生结果
trainModel.bat	从已经提取的特征训练模型(特征文件已上传保存至feat文件夹)
	从已经训练好的模型产生预测结果,模型文

generateResult_with_trainedModel.bat	件已经上传保存至model文件夹

Python版本3.5 64位 ，Winows10 64位

1. 安装依赖包 pip install -r requirements.txt (主要是numpy,pandas,scikit-learn,已经安装可跳过)
2. 安装xgboost (版本0.6, 选择对应python版本与系统版本安装)
   http://www.lfd.uci.edu/~gohlke/pythonlibs/#xgboost
   下载后pip install xxx.whl安装,若已经安装可跳过。
3. 若想快速生成线上提交结果文件，请双击运行generateResult_with_trainedModel.bat；若想通过特征训练模型产生结果文件，请双击运行trainModel.bat，耗时约40分钟。若想从原始数据提取特征至产生线上提交结果，请双击运行generateFeature_and_trainModel.bat，请提前将官方下载的JData_Action_201602.csv，JData_Action_201603.csv，JData_Action_201604.csv，JData_Comment.csv，JData_Product.csv，JData_User.csv存放至datasets文件夹下，由于会缓存中间结果，请把该文件夹存放至至少150G的硬盘上，并且请耐心等待半天。

解题思路

用户模型

训练集，验证集，线上集划分

我们选取目标预测期间前7天与第8类商品有过交互的用户(过滤曾经购买过第8类商品的用户)来构造样本，前60天用来提取特征。划分如下:

- 验证集
  - 特征提取区间 [2016-02-01 00:00:00, 2016-04-06 00:00:00)
  - 样本构造区间 [2016-03-30 00:00:00, 2016-04-06 00:00:00)
  - 标签提取区间 [2016-04-06 00:00:00, 2016-04-11 00:00:00)
- 训练集
  - 特征提取区间 [2016-02-06 00:00:00, 2016-04-11 00:00:00)
  - 样本构造区间 [2016-04-04 00:00:00, 2016-04-11 00:00:00)

    - 标签提取区间 [2016-04-11 00:00:00, 2016-04-16 00:00:00)
- 线上预测集
    - 特征提取区间 [2016-02-11 00:00:00, 2016-04-16 00:00:00)
    - 样本构造区间 [2016-04-09 00:00:00, 2016-04-16 00:00:00)

数据预处理

删除非第8类商品的记录并丢弃model_id字段(减少内存与计算开销)，通过预测期间前7天的用户单秒点击次数与用户点击总量/用户浏览总量(爬虫直接访问某个页面，点击数目远远小于浏览数目，正常用户点击数目远大于浏览数目)过滤可能的爬虫用户。

模型

我们这次只是使用了单模型xgboost.

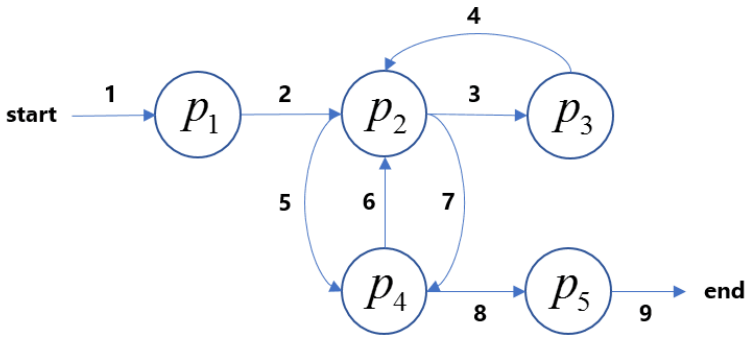
精简版特征

- 用户属性特征
    - 用户等级
    - 用户注册时间距离(注册日距离预测区间第一天距离) (单位: 天)
- 用户时间行为特征 (范围:特征提取区间)
    - 用户总登陆天数 (单位: 天)
    - 用户第一次登陆时间至预测区间第一天时间距离 (单位: 天)
    - 用户最后一次行为时间至预测区间第一天时间距离 (单位: 秒)
    - 用户最后一次与第8类商品交互至预测区间第一天时间距离 (单位: 秒)
    - 用户前(1/2/3/5/7)天的有效行为时间(交互时间) (单位: 秒)
    - 用户前(7/15)天与第8类有过交互天数/用户前(7/15)登陆天数
- 用户行为特征 (范围: 特征提取区间)
    - 用户前(7/15/60)天与第8类商品的操作数/用户前(7/15/60)总操作数
    - 用户前7天的加购物车/删除购物车/关注行为统计
    - 用户前7天行为数/用户前7天有交互天数
    - 用户前15天的点击/加购物车/删除购物车/关注/浏览行为统计(乘上时间衰减权重)
    - 用户前7天单秒点击频率
    - 用户前7天单秒最大点击频率
    - 用户前15天浏览总量/点击总量
    - 用户前7天在窗口(0-6,6-12,12-18,19-24)的行为数 / 前7天交互天数
    - 用户前(1/2/3/5/7)天的总入度
    - 用户前(1/2/3/5/7)天对第8类商品的入度
    - 用户前(1/2/3/5/7)天对第8类商品的入度 / 用户前(1/2/3/5/7)天的总入度
    - 人工规则特征 (该规则在rule.py中有实现，具体为高潜用户ID)

重要特征

1. 用户等级
2. 用户注册时间距离(注册日距离预测区间第一天距离) (单位: 天)
3. 用户总登陆天数 (单位: 天)
4. 用户第一次登陆时间至预测区间第一天时间距离 (单位: 天)
5. 用户最后一次行为时间至预测区间第一天时间距离 (单位: 秒)
6. 用户最后一次与第8类商品交互至预测区间第一天时间距离 (单位: 秒)
7. 用户前(1/2/3/5/7)天的有效行为时间(交互时间) (单位: 秒)
8. 用户前(7/15/60)天与第8类商品的操作数/用户前(7/15/60)总操作数
9. 用户前15天的点击/加购物车/删除购物车/关注/浏览行为统计(乘上时间衰减权重)
10. 用户前7天单秒点击频率
11. 用户前(1/2/3/5/7)天对第8类商品的入度
12. 用户前15天浏览总量/点击总量
13. 人工规则特征 (该规则在rule.py中有实现，具体为高潜用户ID)

以上特征为用户模型关键特征,大部分特征是通过统计分析发现，有两个特征(10,12)用于过滤爬虫用户,并且其本身能反映用户的活跃程度，因为有过爬虫项目经历，对于爬虫特征了解，不难想到这两特征。人工规则特征是基于统计分析发现，具体关注于短期登陆用户的行为分析(近期登陆在前7天，且用户行为只集中于第8类商品，用户等级>2,交互天数<=2,加购过第8类商品)，该规则结果本身在A榜使得f11达到0.254+的成绩，在模型中也有着高分表现。入度特征为本队原创特征，下图为入度特征的解释

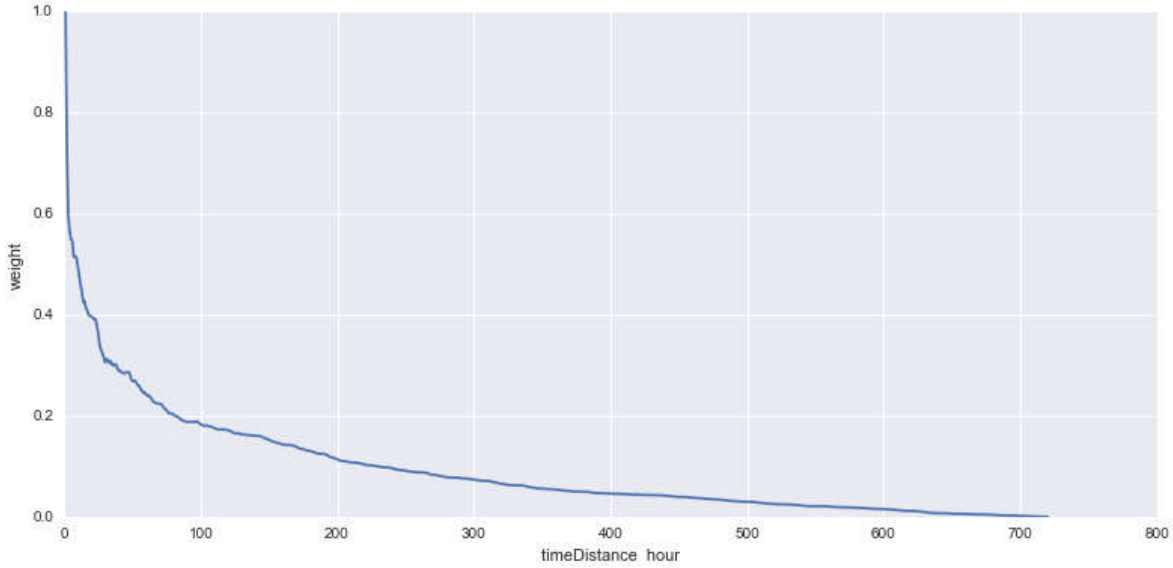


用户访问商品图

节点入度/商品被对比次数

p_1	1
p_2	3
p_3	1
p_4	2
p_5	1

将用户与第8类商品的操作转化为图模型，展示了一位用户的图模型，其中节点表示商品，路径上的数字表示用户访问该商品的顺序，节点的入度表示了商品被对比的次数，整个图的入度表示了用户对比商品的次数，用户更易购买其来回对比多次的商品，例如图中商品 2 被对比了 3 次，其他商品的入度都比该商品低，所以商品 2 被购买量可能性更大。待研究：图中每个节点都包含自身的能量（用户对该节点商品的操作,如点击，浏览，添加），该能量表示用户对该商品的关注度，整个图的能量表示了用户的对第八类商品的关注度。
时间衰减特征是通过统计各个时间窗口的未来购买用户/当前窗口用户数归一化得来，该特征在模型中也有不错的效果。



模型选择与训练

本次只使用了单模型xgboost，该模型支持并行训练，接口丰富且精度高，因此作为我们的首选模型。模型训练过程中样本正负比例极度不平衡，原本打算通过xgboost欠拟合训练筛选样本，但是由于最优成绩代码丢失导致只有最后两天时间，所以并未实现该方法，只是简单设置了xgboost的scale_pos_weight参数来降低该影响。参数调优使用的是sklearn中的网格搜索。

商品模型

训练集，验证集，线上集划分

商品模型的训练集，验证集，线上集划分窗口与用户模型一致，我们选取的是预测区间前7天与第8类商品有过交互的用户-商品pair做样本(前7天可预测的样本占35%且样本比例在1:400~500之间，曾考虑扩大样本范围，前30天可预测样本占45%,但是正负比例在1:1100+,考虑到模型的泛化与机器的配置，选择前7天较为划算)，并且过滤掉购买过第8类商品的用户的所有交互行为。(基于统计发现购买过第8类商品的用户短期并不会再次购买)

数据预处理

初期预想保留每个用户总行为TopN个样本，但是用户基数大，筛选耗时较长，时间有限，并未实现，因此商品模型未做处理，正负比约在1：400~500。

模型

单模型xgboost。

精简版特征

商品特征整合了用户特征(用户属性特征与用户-品类特征)，已经在用户模型表述，之后将不再赘述。

- 用户-商品特征
  - 用户与商品最早/晚交互时间至预测窗口时间距离 (单位 秒)
  - 用户前(7/15/28/60)天与该商品有交互的天数
  - 用户前(7/15/28/60)天与该商品有交互的天数 / 用户前(7/15/28/60)登陆天数
  - 用户前28天对该商品的总交互行为统计(乘上时间衰减权重)
  - 用户前28天对该商品的浏览/加购物车/删除购物车/关注/点击行为统计(乘上时间衰减权重)
  - 用户前(4h,8h,16h,24h,2,3,5,7,15,28)对该商品加购物车/关注行为统计
  - 用户前(1/2/3/5/7/15/28)天的对该商品总行为数/用户前(1/2/3/5/7/15/28)总行为数
  - 用户前(1/2/3/5/7)天对该商品的有效交互时间
  - 用户前(1/2/3/5/7)天对该商品的有效交互时间 / 用户前(1/2/3/5/7)天对该类商品的有效交互时间
  - 用户对该商品前7天的浏览数/点击数
  - 用户前(4h,8h,16h,24h,2,3,5,7)对该商品的入度
  - 用户加购该商品数目/总购物车数目
- 用户-品牌特征
  - 用户前(1/3/5/7/10/14/28)对该品牌的操作数/用户前(1/3/5/7/10/14/28)的总操作数
- 商品特征
  - 商品近期差评率
  - 商品属性特征(one-hot)
  - 商品前(1/2/3/5/7/10)的净流量/该特征提取窗的总净流量(净流量: 指访问用户数目)
- 其他特征
  - 该商品前(1/2/3/5/7/10/14/28)销量/该商品品牌同类商品(1/2/3/5/7/10/14/28)的销量

重要特征

1. 用户与商品最早/晚交互时间至预测窗口时间距离 (单位 秒)
2. 用户前(7/15/28/60)天与该商品有交互的天数 / 用户前(7/15/28/60)登陆天数

3. 用户前28天对该商品的总交互行为统计(乘上时间衰减权重)
4. 用户前28天对该商品的浏览/加购物车/删除购物车/关注/点击行为统计(乘上时间衰减权重)
5. 用户前(1/2/3/5/7/15/28)天的对该商品总行为数/用户前(1/2/3/5/7/15/28)总行为数
6. 用户前(1/2/3/5/7)天对该商品的有效交互时间
7. 用户前(1/2/3/5/7)天对该商品的有效交互时间 / 用户前(1/2/3/5/7)天对该类商品的有效交互时间
8. 用户对该商品前7天的浏览数/点击数
9. 用户前(4h,8h,16h,24h,2,3,5,7)对该商品的入度
10. 用户加购该商品数目/总购物车数目
11. 商品近期差评率
12. 商品前(1/2/3/5/7/10)的净流量/该特征提取窗的总净流量(净流量: 指访问用户数目)
13. 该商品前(1/2/3/5/7/10/14/28)销量/该商品品牌同类商品(1/2/3/5/7/10/14/28)的销量

以上特征与用户模型的重要特征均为商品模型的重要特征，思考特征方向为用户特征，用户-品类，用户-商品，用户-品牌，商品特征与其他交叉特征。整个思考流程是先确定用户是否购买商品，然后通过用户对商品的关注程度与活动契机来预测用户是否购买。特征12反映了商品近期的热销程度，特征13反映了商品在同类商品的热销程度，其他特征均为衡量用户对商品的关注程度。

模型选择与训练

商品模型依旧采用xgboost算法，由于样本数量较大，cv调参速度慢，只调整了部分参数。由于用户模型与商品模型的正负比例极度不均衡且正例数目极少，在训练模型的时候为了保证更多的正例被学习到，我们并没有划分部分数据用于观察模型的loss来控制模型的拟合程度，而是直接使用了全部数据训练，通过5折交叉验证获取最佳迭代次数，这为B榜成绩带来一定的提升。预测是结合了用户模型的预测结果，选取用户模型Top500,然后在商品模型中选择用户购买概率最大的商品，商品模型选择Top500,两者取并集为提交结果，该融合方式为B榜总分带来了很大的提升。

有趣的发现

后期提升的原因主要是特征质量的提升，在之前无脑堆砌特征的模型上观察了各种特征的重要排序，对许多类特征做了筛选与扩充。模型调参非常重要，用户模型调参之后为线上f11带来了近0.03的提升。我们队伍最突出的优势应当是构建了稳定线下验证集与线上一致的评测方式，虽然线下验证集的分数与线上分数相差较大，但是线下与线上能够做到同步变化，这是我们在B榜能够做到稳定提升的重要原因。

generated by haroopad