2017“达观杯”
个性化推荐算法挑战赛

团队：THLUO

成员：罗江伟（队长） 花志祥 李凯东 洪鹏飞 张丽品

目录

contens

1 团队介绍

团队介绍

花志祥：**kaggle全球数据科学家top50，Rental Listing Inquiries第一名**

李凯东：京东数据科学家

洪鹏飞：浙江财经大学硕士在读，主要研究方向：生存分析，降维

张丽品：北京航空航天大学硕士在读，主要研究方向：机器学习

罗江伟：机器学习、数据挖掘爱好者

IJCAI-17 口碑商家客流量预测第二名
阿里云安全算法挑战赛线上赛第一名
【广东大赛】机场客流量的时空分布预测第二名
生活大实惠：O2O优惠券使用预测第三名
阿里聚安全算法挑战赛第二名

阿里聚安全算法挑战赛唯一人气奖
融360天机“天机”金融风控大数据竞赛三等奖
2017中国网络安全技术对抗赛线上赛第三名
2017中国网络安全技术对抗赛特殊贡献奖
携程用户预订售卖房型概率预测第一名

团队成员在许多数据挖掘比赛中都取得过非常好的名次，这也是本次达观杯能够获得A/B榜最高分的重要基石

2 问题描述

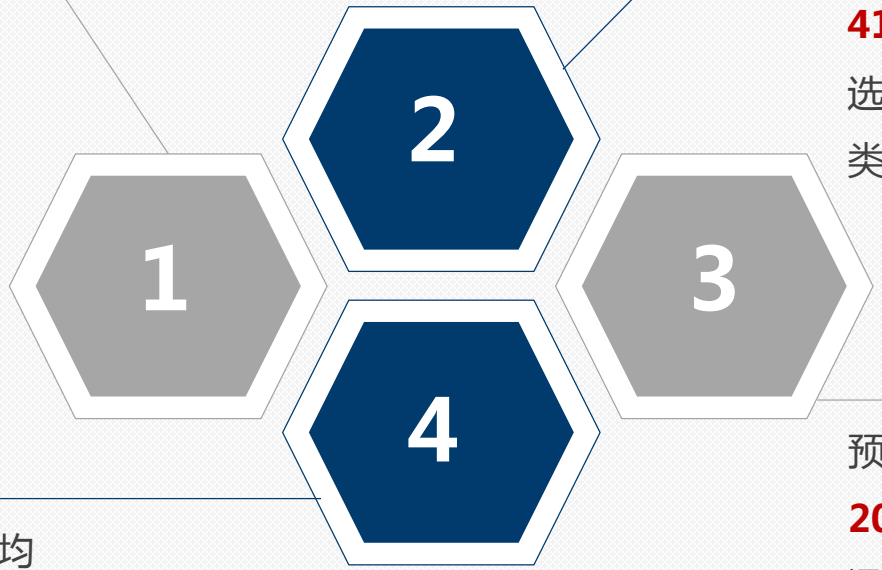
问题描述

历史记录

28501个用户在2017.02.16到2017.02.18期间与资讯的**3272043**次交互行为

资讯候选集

41252条候选资讯，候选资讯的属性包括资讯类别和资讯发布时间



目标

预测出**28501**个用户在**2017.02.19**会浏览的资讯候选集中的**5条**资讯

评价标准

使用**第5位**平均精度均值进行评估

$$ap@5 = \sum_{k=1}^{5} P(k)/\min(m, 5)$$

核心：发掘用户的兴趣爱好

3 算法框架

算法框架

使用test.txt与news_info.csv中item_id的交集构成资讯候选集

test.txt中item_id的数量（有重复）：31082个

new_info中item_id的数量（无重复）：41252个

两者的交集中item_id的数量（有重复）：30313个

通过这一条规则加上去掉用户之前访问过资讯，线上分数达到 **0.072287**

算法框架

加入时间衰减因子，改善用户对资讯类别的偏好R_{uc}的计算方式

通过下面的公式更新R_{uc}：

$$Ruc = \sum_{d} \frac{Ruc\_d}{1+0.15 \cdot GAP\_d}$$

R_{uc_d}表示用户在第d天对资讯类别的交互次数

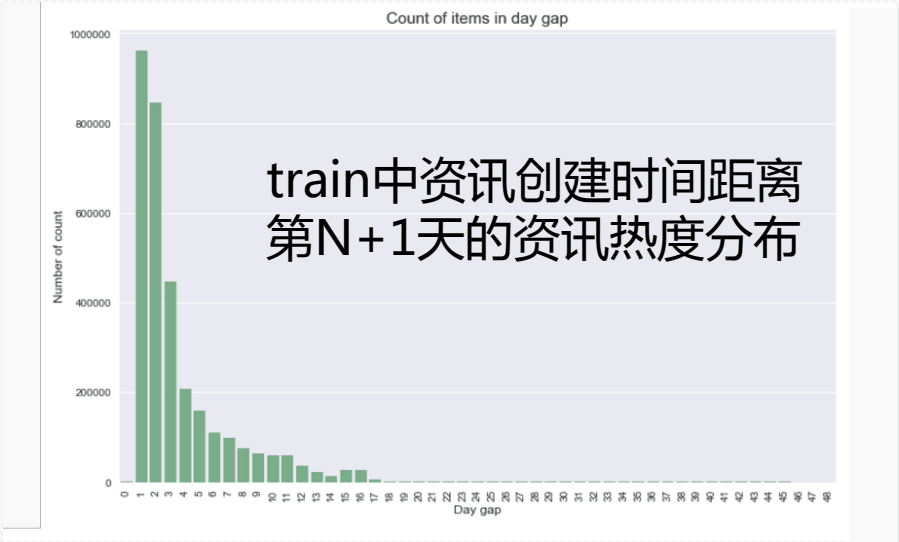
GAP_d表示第d天距离第N+1天的天数

即用户点击资讯类别的时间距离第N+1天越近，越能体现用户的即时兴趣

通过这次改善，线上分数达到 **0.07236**

算法框架

在计算test中资讯的热度时，我们发现如下图所示的现象



- ✓ test只是第N+1天的百分之5的随机抽样，两者分布并不一致
- ✓ 可通过对资讯的创建时间做衰减，来修正test中的热度分布

算法框架

通过如下公式来更新test中资讯的热度H_i

$$Hi = \frac{Hi}{1+0.4 \cdot GAP\_i}$$

GAP_i表示资讯创建日期距离第N+1的天数

另外，由右图中可知，创建日期距离6天的资讯明显偏多，因此对这一天的资讯加大惩罚，将6改为7.2。

这次规则的改进后，线上分数达到 **0.07368**

算法框架

用户点击越不热门的资讯类别，越能体现用户的个性化需求，也就是用户可能会越喜欢

因此使用如下公式来改进用户对资讯类别的偏好R_{uc}

$$R_{uc} = \frac{R_{uc}}{\log(1+N\_Cate\_u)}$$

N_Cate_u表示交互该资讯类别的人数

这次规则的改进后，线上分数达到 **0.073944**

算法框架

之后发现了谷歌的一篇论文:

Personalized News Recommendation Based on Click Behavior.

论文的核心骨架是：

通过这篇论文，进一步改进了用户对资讯类别的偏好

算法框架

用户过往对cate的偏好可以通过如下公式得出:

$$interest^t(category = c_i)$$

$$= p^t(click \mid category = c_i)$$

$$= \frac{p^t(category = c_i \mid click)\, p^t(click)}{p^t(category = c_i)}$$

$p^t(click \mid category = c_i)$: 用户在第$t$天对资讯类别的偏好

$p^t(category = c_i \mid click)$: 用户在第$t$天交互类别$ci$的概率，可以从*train*中计算得出

$p^t(category = c_i)$: 资讯类别ci在第t天的点击率，同样可以从*train*中计算得出

算法框架

假设用户点击新闻的概率是固定的，因此$p^t(click)$忽略不计

然后用户过去三天对资讯类别的偏好可以用如下的公式加权得到：

$$interest(category = c_i)$$

$$= \frac{\sum_t \left(N^t \times interest^t(category = c_i) \right)}{\sum_t N^t}$$

$$= \frac{\sum_t \left(N^t \times \frac{p^t(category = c_i \mid click)\, p^t(click)}{p^t(category = c_i)} \right)}{\sum_t N^t}$$

其中N^t表示用户在第t天交互新闻的次数

算法框架

再考虑现阶段品类的流行度，可以通过如下公式得出

$$p^0(category = c_i \mid click)$$

$$= \frac{p^0(click \mid category = c_i)\, p^0(category = c_i)}{p^0(click)}$$

将其展开得到：

$$p^0(category = c_i \mid click)$$

$$\propto \frac{interest(category = c_i)\, p^0(category = c_i)}{p(click)}$$

$$\propto \frac{p^0(category = c_i) \times \sum_t \left(N^t \times \frac{p^t(category = c_i \mid click)}{p^t(category = c_i)} \right)}{\sum_t N^t}$$

算法框架

$p^0(click \mid category = c_i)$：使用用户过往对*cate*的偏好来代替

$p^0(category = c_i)$：test中资讯类别的点击率

综上，通过下式改进用户对cate的偏好的计算

$$p^0(category = c_i \mid click)$$

$$\propto \frac{p^0(category = c_i) \times \left(\sum_t \left(N^t \times \frac{p^t(category = c_i \mid click)}{p^t(category = c_i)} \right) + G \right)}{\sum_t N^t + G}$$

G表示用户的虚拟点击次数。

通过这次改进，线上分数达到 **0.074705**

算法框架

用户交互的资讯的创建时间越晚，则该类资讯越能够代表用户当前对资讯类别的即时兴趣

因此根据资讯的创建时间距离第N+1天的天数，对用户交互资讯的次数进行时间衰减，公式如下

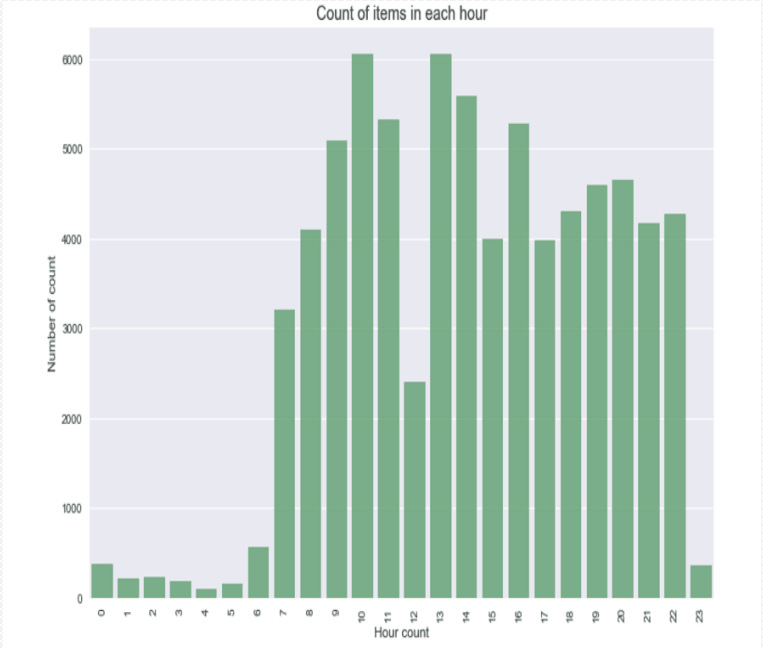
$$R_{ui} = \frac{R_{ui}}{1 + 0.15 \cdot GAP\_i}$$

GAP_i表示资讯创建日期距离第N+1的天数

这次规则的改进后，线上分数上升了 **0.0004**

算法框架

经过分析，我们发现每天各个时刻的资讯的创建数量是不同的



越是在非工作时间创建的资讯，资讯的重要度越大，也越有可能在创建后成为热点资讯

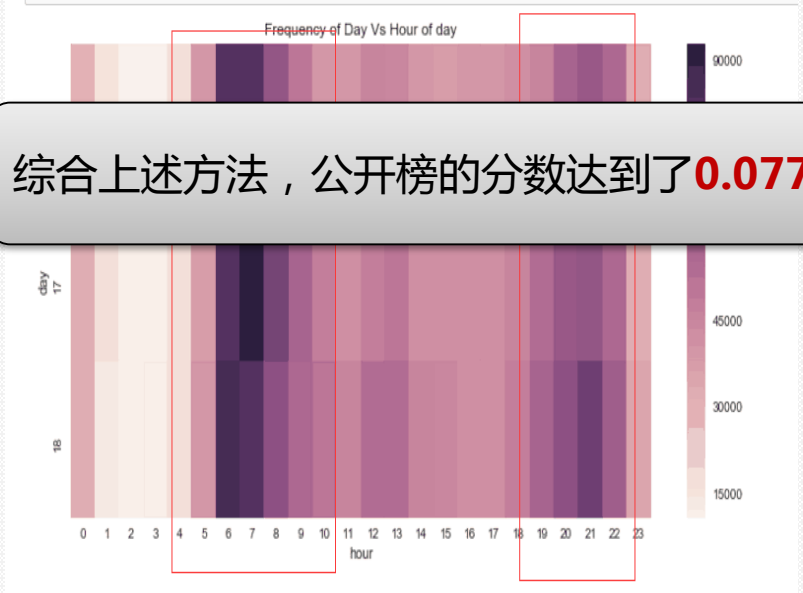
计算用户对于cate的偏爱程度和test中资讯的热度时，根据资讯的创建小时对资讯进行加权

资讯创建时间hour	权重weight
0-6	1
23	1
12	0.8
7-11, 13-22	0.6

经过本次加权，线上分数上升了 **0.0013**

算法框架

然后画出16,17,18号三天中在各个时刻被创建的资讯的数量热度图



在新闻量较大的时间段，用户有很多种选择。而用户在有很多选择的情况下依然会点击的cate，正是用户偏爱的cate。

在计算用户对cate的偏爱程度时，根据用户的action_time对train中资讯进行加

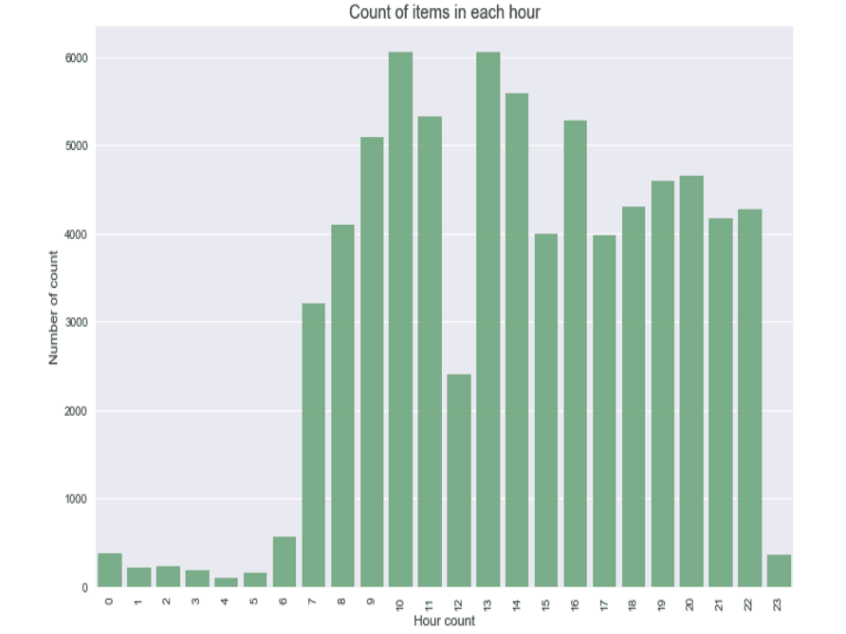
综合上述方法，公开榜的分数达到了**0.077290**，该分数的结果文件是最终提交的两个文件之一

6-9	1
19-22	0.8
其他时间点	0.6

经过本次加权，我们的分数有所上升。

算法框架

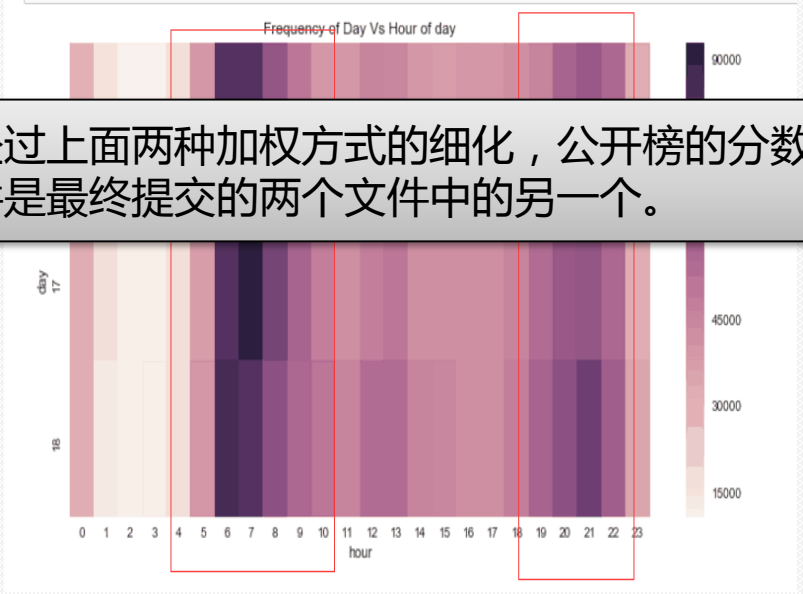
为了扩大分数优势，在计算用户对于cate的偏爱程度和test中资讯的热度时，对加权方式进行细化



资讯创建时间hour	权重weight
6-8	1.1
9	1
7	1.2
19,20,22	0.9
21	1
12,13	0.7
其他时间点	0.6

算法框架

为了扩大分数优势，在计算用户对cate的偏爱程度时，将根据用户的action_time的时间点对train中资讯的加权进行细化



资讯的action_time	权重weight
7	1.2
19,20,22	0.9
21	1
12,13	0.7
其他时间点	0.6

经过上面两种加权方式的细化，公开榜的分数从**0.077290**涨到了**0.077490**，该分数的结果文件是最终提交的两个文件中的另一个。

算法框架（思路总结）

- 对train中资讯action_time做衰减
- 用户点击越不热门的资讯类别，越能体现用户的个性化需求
- 谷歌论文：品类偏好=过往偏好*品类流行度
- 用户交互的资讯的创建时间越晚，越能体现用户当前对资讯类别的即时兴趣
- 越是在非工作时间创建的资讯，资讯的重要度越大
- 在新闻量较大的时间段，用户在有很多选择的情况下依然会点击的cate，正是用户偏爱的cate

- 对test中资讯的创建时间做衰减
- 越是在非工作时间创建的资讯，资讯的重要度越大

4 难点亮点

难点亮点

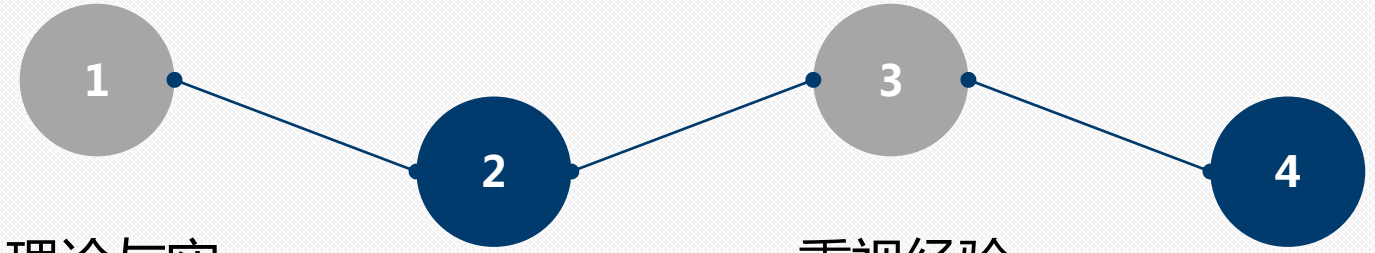
1. 采用谷歌论文中的贝叶斯框架改进用户对cate偏好程度的计算

2. 考虑到：越是在非工作时间创建的资讯，资讯的重要度越大

3. 考虑到：在新闻量较大的时间段，用户在有很多选择的情况下依然会点击的cate，正是用户偏爱的cate

5 总结

总结



理论与实践相结合

重视数据分析，关注数据分布

重视经验的积累

结识志同道合的人

对数据的分析，对业务场景的理解永远是最重要的。

THANKS!

请各位专家批评指正