

# 用户长期兴趣域 推荐

赵海臣

# 遇到的问题

❧ 千人千面自从上线以来，用户对ALS长期模型的推荐结果兴趣度在不断地下降，究其根源在于用户的长期个性比较稳定，并且通过ALS推荐出的商品变化率较低。

❧ 经过近一年的推荐系统研发积累，发现用户的长期兴趣应该是一个较为模糊的兴趣域，与短期兴趣所需要追求的快速精确度相比，长期模型的基础算法过于精细并不是一种合理的解决方案。

❧ 因为长期个性化考察的是用户的长期行为，例如一年，因此单天的数据变化( $1/365$ )相当缓慢。因此，商品精确度过高且低变化率容易造成用户疲乏。

# 长期个性化模型的兴趣域泛化途径

☞ 用户的长期个性对应的是个性兴趣域：

## ★ 兴趣域

- 每个商品都属于一个兴趣域
- 某个或者某些共有特性的商品聚集成一个兴趣域
- 通过用户的行为进行聚类，获得用户的心理兴趣域

## ★ 用户长期兴趣域模型

- 将基于用户行为聚类出来的用户的兴趣域，使用ALS MF算法进行整理排序，获得用户兴趣域分数
  - ALS MF协同过滤算法可以参考整体用户兴趣域，对指定用户进行全局性地兴趣分析
- 用户获得的个性化模型是一个兴趣域分数，兴趣域分数越高，说明用户对该兴趣域的兴趣越高

# 用户兴趣域行为rating—用户商品rating

✎ 首先生成用户商品rating

★ 使用的用户行为数据：

- 浏览事件

- userId - productId - data\_date

- 订单事件

- userId - productId - data\_date

- 购物车事件

- userId - productId - data\_date

- 收藏事件

- userId - productId - data\_date

★ 考察时长范围：

- 浏览事件：180天

- 订单事件：90天

- 购物车事件：90天

- 收藏事件：90天

# 用户兴趣域行为rating—用户商品rating

## ★ 时间衰减降权：

- 考虑到用户的近期行为更反映用户的现在的兴趣偏好，因此，根据用户的行为发生时间，对事件的有效值进行降权：

$$actionValue = baseValue + \exp(-\omega * |\Delta day|)$$

- $baseValue = 0.3$

- $\omega = 0.1$

## ★ 用户商品rating

- 对用户不同事件进行加权，获得用户商品综合分：

- 不同事件发生一次的权重值：

- 浏览事件：1
- 订单事件：2
- 购物车事件：2
- 收藏事件：2

- 用户商品rating

- 将用户-商品的交互事件的值加权累加，获得用户对商品的rating

# 用户兴趣域行为rating

✎ 将用户-商品rating转换为用户-兴趣域rating：

- ★ 将用户-兴趣域下对应的商品rating进行累加，获得用户-兴趣域rating

# 使用ALS进行用户兴趣域推荐排序

将用户-兴趣域-rating使用ALS进行推荐排序，参数为：

- ★ rank = 100
- ★ iteration = 10
- ★ recommendOutputNum = 30

通过ALS算法推荐排序，获得用户-兴趣域-recommendationScore

- ★ 一个用户取top30最高兴趣分的兴趣域
- ★ 兴趣域对应着商品聚类的簇
- ★ 每个簇对应着不同数量的商品



# 兴趣域推荐商品

✎ 由于一个用户的推荐列表有限(大约100个推荐名单)，而有的簇很大，一个簇就上百个商品，因此需要对每个簇的商品进行限制

★ 如何在兴趣域中挑选商品能获得最高概率的点击：

- 由于用户的兴趣粒度为兴趣域，一个兴趣域下有众多商品，因此如何从一个兴趣域中挑选用户最可能喜欢的商品
- 取每个兴趣域中topN销售量/销售额最高的商品——销售量/销售额最高的商品拥有最高的用户购买概率

★ 簇内商品销售额归一化，便于后续簇与簇之间的推荐值融合：

- 簇内商品优先级“distance”：将簇内商品的销售额除以簇内最大商品销售额，获得簇内商品归一化的销售额，作为簇内商品推荐概率优先级
- 单一簇推荐上限：对单一簇的推荐数量设置上限，取Top20商品。



# 商品推荐池

✎ 获得了用户-兴趣域-rating (30/user)，以及兴趣域-商品-distance (20/cluster)，接下来将用户-兴趣域-商品-推荐分联系起来，并组合成一个推荐商品池。

## ★ 最后推荐分的确定：

- 需要将用户-兴趣域-rating的分数与兴趣域-商品-distance的分数融合，可以加上各种系数： $a \cdot \text{rating} + b \cdot \text{distance}$
- 但经过最后调试，发现 $1 \cdot \text{rating} + 1 \cdot \text{distance}$ 的召回率最高，并且各个域的混合程度也最高，不会出现簇成块出现的现象，因此融合公式为：

- $\text{recommendationScore} = 1 \cdot \text{rating} + 1 \cdot \text{distance}$

## ★ 用户商品推荐池的确定：

- 一个用户推荐有30个兴趣域，兴趣域的商品最大数量是20个，因此一个用户最多能获得 $30 \cdot 20 = 600$ 个推荐商品
- 取融合分数最高的300个推荐商品作为推荐商品池

# 推荐商品池中的每日推荐列表

每个用户有300个推荐商品池，每天从推荐商品池中随机推荐130个商品，按照融合分数排序给分。

- ★ 变化率：隔天推荐列表变化率大概是30%
- ★ 能够大大改善之前高精确度商品推荐的变化率低的问题

# 更新策略

## 兴趣域更新策略：

- ★ 周六/周三定期对兴趣域商品重新聚类，因为随机种子采用日期，因此能保证每次聚类出的商品兴趣域不会相同。
- ★ 兴趣域改变的情况下，ALS的推荐排序也会有变化，ALS推荐排序周六/周三随着兴趣域更新而重新计算
- ★ 商品推荐池周六/周三随着兴趣域更新而重新计算
- ★ 除了周六/周三更新商品推荐池，其余时候都是使用现有的推荐池，从300个商品中随机抽取130个推荐商品。

## 变化率数据：

- ★ 相同推荐池的情况下，隔天变化率 70%
- ★ 隔推荐池的情况下，变化率85%

# 推荐效果

🌀AB测试：

**THE END**

**THANK YOU!**