

# 推荐系统之Exploitation & Exploration

原创 hellobill 比尔的新世界 2020-09-05

收录于话题

#推荐系统 20 #探索与利用 1 #深度学习 2



扎什伦布寺

## 背景

推荐系统通过对用户在APP里面的历史行为信息进行挖掘，向用户推荐与其历史行为相匹配的内容（商品、图文或者视频等），这其实就是一种对信息的Exploitation（利用）行为。但是推荐系统经常被诟病的问题是，总是推荐相似的信息，缺乏新意，容易造成审美

疲劳，这也是很多用户在使用相关APP一段时间后选择离开的重要原因。如果系统里面存在用户感兴趣的内容，但是却没有让用户方便地获取到相应的信息，说明当前的推荐系统有问题。用户越看什么系统越推什么，慢慢造成推荐的信息形式越来越窄，从而形成信息茧房，这是当前推荐系统普遍会遇到的问题。

破圈的方法之一是Exploration（探索），即通过一些方法去探索用户可能的兴趣空间，不断扩大用户的兴趣边界，甚至发掘用户自己都不曾意识到的新的兴趣点，如果推荐系统能够做到这一点，那不夸张的说，对整个社会的进步都会有不小的促进作用。不断扩大用户认知边界，是为开放，收缩用户既有的认知，是为封闭。当然，有些人知道自己需要什么，会主动的去选择信息，但是在信息爆炸的时代，绝大多数人都是在被动的吸收信息，此时信息的分发方式影响就很大了，一个好的推荐系统，应该是能够不断地扩大用户的认知边界，而不是将用户包裹在信息茧房里，这应该作为做推荐系统的初心。

## Exploration方法

常见的Exploration方法有，朴素Bandit、Epsilon-Greedy、UCB、Thompson Sampling、LinUCB、COFIBA等，但是这些方法在当前的推荐系统中其实用得很少，主要原因是Exploration方法往往有瞎猜的性质，因为不能再完全根据既往的信息做决策。Exploration的意思就是在不断的试探当中拓展用户的兴趣边界，但是试探是有代价的，如果推出来的东西用户一点兴趣没有，久而久之，用户就失望了，从而选择离开，这样的推荐系统就更没有价值了，连最起码的商业目的都没有达到。所以，做Exploration相关的尝试，往往是针对老用户、死忠粉，同时，选择的内容池子质量也更高，至少做到推出去的内容用户不喜欢，但也不能让人讨厌。

- 朴素Bandit根据历史信息选择平均回报最高的，是一种贪心算法。
- Epsilon-Greedy在朴素Bandit的基础上增强了探索能力，以 $\epsilon$ 的概率做随机选择。
- UCB算法在朴素Bandit的基础上根据选择总次数及每个Item被选择的次数计算一个置信区间，然后根据平均分及置信分的和大小做选择。
- Thompson Sampling根据每个Item的beta分布产生的随机数去做选择，同时根据选择结果更新beta分布的参数。
- LinUCB是引入特征与监督学习的UCB算法。
- COFIBA是协同过滤结合Bandit的算法，User-based协同过滤来选择要推荐的Item，选择时使用了LinUCB思想，同时根据用户反馈更新相关矩阵参数。

以上方法，就类Epsilon-Greedy的方法在强化学习推荐系统当中还会使用。对于大规模离散特征的深度学习推荐系统，值得一试的Exploration策略有：

- i. 扰动训练好的深度模型NN参数，具体操作为对NN参数加上一定的高斯噪声。

ii. 随机丢弃部分强记忆型的ID类特征，如UserId和DocId。

策略一正常训练和保存NN参数，在Serving的时候向NN参数加入高斯噪声。高斯噪声为标准分布的随机数，通过当中的来控制噪声的强度。我们尝试过几组参数，0.1，0.01，0.001，0.0001,发现0.001能够取得比较好的探索和利用的平衡。

策略二 在预估的时候，随机将部分id类特征的embedding置0，相当于不利用已经学习好的这些slot的embedding。因为像uid，did这类特征一般起到的就是记忆的作用，如果记忆能力太强了，那就容易影响到泛化能力。通过随机将部分id类特征的embedding置0，可以提升模型泛化能力，同时提升模型的探索能力。

这两个策略基本都达到了预期目的，用户的阅读散度，文章的多样性都能一定程度的提升，同时大盘实时指标并没有下降，甚至还有一定程度的提升，长留也是正向的。

## 其它思考

在信息流产品中，用户大部分的PV和时长都贡献给了主推荐页，可以通过子频道页增强对用户的探索能力。在特征构建的时候，我们谈到可以通过增加相似用户阅读文章和相似文章聚类特征来增强对新用户和新文章的探索能力。但是现实是，其它子频道页被大部分用户点开的机会很少，增加探索特征，但是模型仍然倾向于去记忆历史。

可以尝试使用个别探索性质强的召回做一下强插，比如说User-CF，因为精排就是一个贪心算法，即使召回出来了不错的内容也有可能被精排给干掉，使用强探索能力的召回做推送或者强插，结合精选内容池，说不定能够取得不错效果呢。

然后就是强化学习，以定义的长期收益作为目标，在与用户的交互过程当中持续进行学习。

提升探索能力往往意味着用户体验的降低甚至流失，但是持续给用户推荐相同类型的内容，用户也会逐渐阅读疲劳从而流失。好的推荐系统在于在Exploration & Exploitation之间取得一个比较好的平衡，在不严重影响用户阅读体验的前提下，时不时地给用户推荐一些好玩的、新奇的或者新颖的东西，探索甚至放大用户的兴趣，这也是做推荐系统的乐趣所在。

## 留言区

喜欢此内容的人还喜欢