

1. 什么是position bias

Position-Bias是指 item 在展示页面的排序位置，及其相对广告的位置等。经验和有关论文都告诉我们，这种位置会影响item的点击率。这种影响跟用户的真实兴趣无关，而是跟用户的注意力、用户对广告的情绪有关。例如有**眼动实验**表示用户会很少关注那些在列表靠下的item。离线分析显示，排在前面的文章一般比排在后面的文章点击率高，离广告近的文章点击率一般较低，这种bias被称为position-bias。**用户更愿意点击排在前面的商品，之后这些商品就越容易排在前面... 这样就形成了一个正反馈循环，让推荐生态恶化，形成“强者愈强、弱者愈弱”的马太效应。**

为了有更高的CTR预估精度，CTR预估从早期的LR、FM、FFM等支持大规模稀疏特征的模型，到XGBoost、LightGBM等树模型的结合，再到Wide&Deep、Deep&Cross、DeepFM、xDeepFM等支持高阶特征交叉的深度模型，进一步演化到DIN、DIEN、DSIN等结合用户行为序列的深度学习模型，一直作为工业界以及学术界研究的热点领域之一，被不断探索和不断创新。但是，position bias的问题却研究甚少。

各大公司现在都在强调推荐生态的理念，debias也是构建良好推荐生态中不可或缺的一个关键要素。

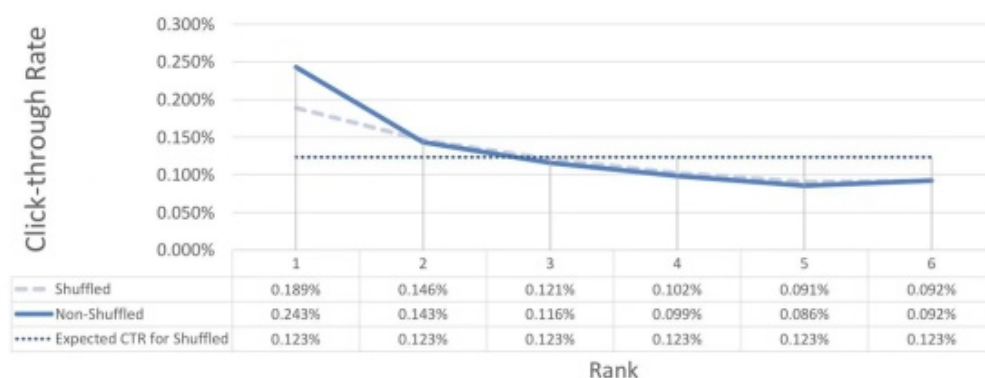


Figure 3: Users of the digital library Sowiport appear to exhibit bias in choosing items, following a random shuffling of recommendations

经过随机shuffle之后，按理说用户对各个位置的平均点击率应该是一样的，但实际上用户还是对top item的点击率高，这说明position bias确实存在

笔者将介绍一下自己在实习时使用的几种业界常用的降低position bias的方法。当时在做文献调研的时候发现了很多用统计方法来解决position bias，但数学公式太过复杂而我数学很差:(公司也招了些学统计的人来做一块。这篇不会涉及复杂的数学公式。

2. 解决方法

2.1 position作为特征(as feature)

该方法出自Airbnb的一篇经典的搜索文章 Improving Deep Learning for Airbnb Search.

给定一个用户 u ，以及一个query q 和一个list l ，以及list中的每个位置 k 。用户预订的概率是：

$$P_{booking} = P(relevant = 1|l, u, q) * P(examined = 1|k, u, q)$$

其中前半部分是这个item被用户预订的概率，后半部分是item在位置k被用户看到的概率。二者相乘就是一个item在位置k上被预订的概率。理想情况下我们只要关注于前半部分然后对list进行排序就OK。

Airbnb在训练时加入位置信息，但是在预估的时候将特征置为0。但是发现模型的NDCG跌了1.3%。文章指出，可能是训练的时候相关性的**计算过度依赖位置信息，但是在测试的时候，这个位置信息就没有了，所以导致效果变差。**

为了减少相关性计算对position feature 的依赖，文章采用了训练阶段**对position feature 进行dropout**，这样就能够减少模型对位置特征的依赖。

通过实验文章选择了0.15的dropout比例，对线上的结果有0.7%的下单率的提升。经过多次迭代之后，订单收入涨了1.8%。需要注意的是位置特征不能与其他特征做交叉。

2.2 position作为模块(as module)

(a) shallow tower

这种方法出自Youtube多目标排序论文 Recommending What Video to Watch Next: A Multitask Ranking System。

如果不去除position bias，那么用户对一个item的ctr是由user engagement(用户偏好)和position bias共同造成的。如果能够将user engagement和position bias独立开来，将有助于提升模型的推荐质量。本文提出的做法**类似wide&deep，即添加一个浅层网络(shallow tower)**，建模position bias的shallow tower就对应WDL里面的wide端。shallow tower输入position feature, device info（原因是在不同的设备上会观察到不同的位置偏差), 以及其他能够带来位置偏差的特征，输出为position bias分量（标量）。这就将user engagement和position bias 独立开来了。

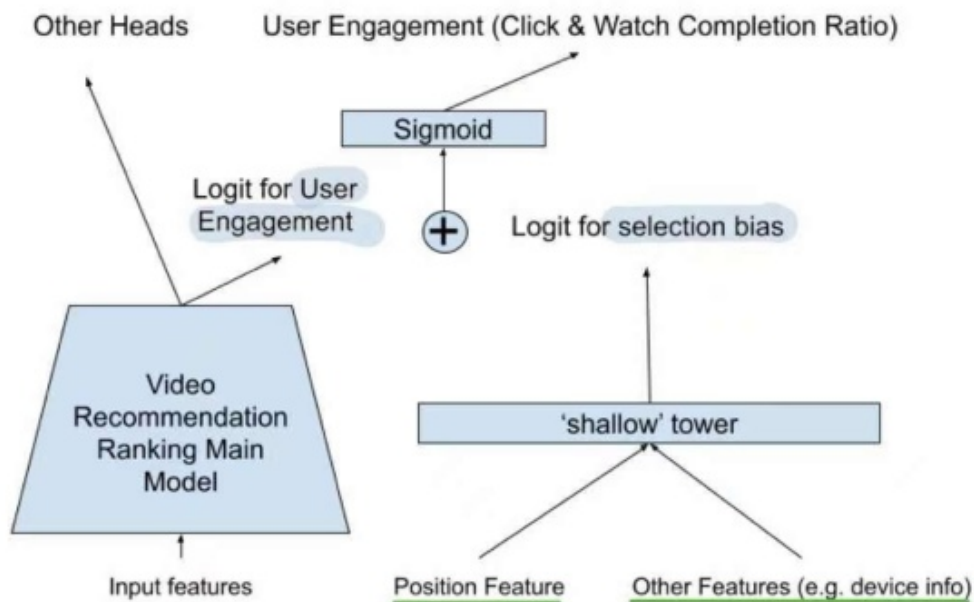


Figure 3: Adding a shallow side tower to learn selection bias (e.g., position bias).

知乎 @魔法学院的Chilia

在主模型的输出层的sigmoid激活函数之前，加上浅层网络的bias分量。训练的时候，随机丢掉10%的位置特征，防止模型过度依赖位置特征。预测的时候，直接丢掉浅层网络。

(b) PAL

出自华为Recsys 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems.

这篇文章的intuition就是，如果把position作为feature，那么有个问题就是在线上inference的时候，我们必须给这个position赋个值吧，但是赋值不同是会导致模型效果不同的。为了选择一个好的值，我们需要做线上/线下实验来找到之。但是，线上实验的成本很高（通常需要几个星期来做，而且会引一部分流量过来），所以我们只能做线下实验来找到这个最好的position值。但是，即便我们找到了“最好”的position值，如果数据分布一发生变化，或者用在另一个场景下，这个值就不再是最好的了。

所以，PAL采用的是position-as-module的方法，即使用一个子网络专门建模position bias分量，然后和“真实”CTR分量求乘积。最后线上inference的时候直接把position bias网络去掉就行了。

作者分析到，用户点击广告的概率由两部分组成：

- 广告被用户看到的概率
- 用户看到广告后，点击广告的概率

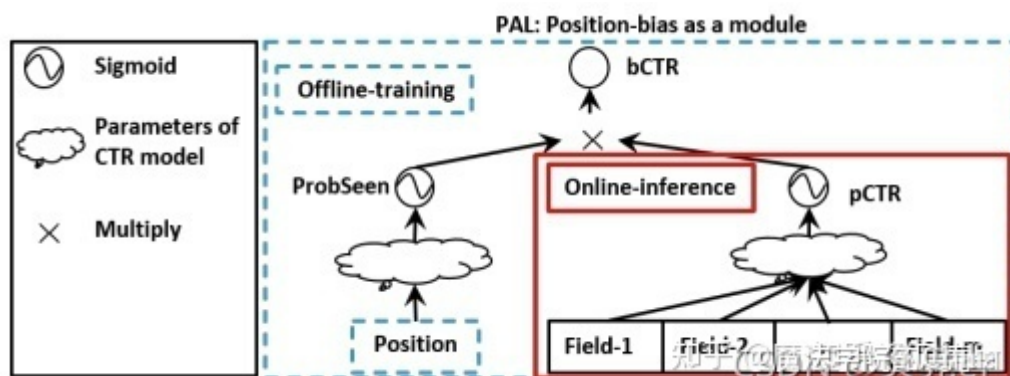
$$p(y = 1 | \mathbf{x}, pos) = p(seen | \mathbf{x}, pos) p(y = 1 | \mathbf{x}, pos, seen).$$

那么可以进一步假设：

- 用户是否看到广告只跟广告的位置有关系
- 用户看到广告后，是否点击广告与广告的位置无关

$$p(y = 1 | \mathbf{x}, pos) = p(seen | pos) p(y = 1 | \mathbf{x}, seen).$$

基于该假设，就可以分开建模：



其中：ProbSeen部分是预估广告被用户看到的概率（只和position有关），pCTR部分是用户看到广告后，点击广告的概率（和position无关），然后实际的CTR是两者的乘积，与label求loss：

$$L(\theta_{ps}, \theta_{pCTR}) = \frac{1}{N} \sum_{i=1}^N l(y_i, bCTR_i) = \frac{1}{N} \sum_{i=1}^N l(y_i, ProbSeen_i \times pCTR_i),$$

线上serving的时候，直接预估pCTR即可(ProbSeen都看作是1)。PAL和shallow tower的区别在于PAL是连乘概率，而shallow tower是类似wide&deep的相加。

AB test结果：

- 对照组：2%的流量引过来，使用baseline（即position as feature）
- 实验组：2%的流量引过来，使用PAL

做一个星期的实验，考察指标为真实CTR(下载数量/impression数量)。发现PAL相比于position-as-feature方法在线上有了很大提升，虽然PAL在线下的表现(CTR预估中的AUC)还不如position-as-feature。

但是文中并没有把PAL和不用PAL的效果进行比较。根据我的实际经验，去除position bias之后，线上CTR的指标是会降低的，因为去掉position bias相当于去掉了热点效应。但是CVR和一些satisfaction指标会有上升。

笔记：

其实，PAL的设计和ESMM有异曲同工的地方，都是将事件拆解为两个概率事件的连乘，但是PAL的假设过强，事件的关联性没有ESMM的click->conversion这样的强关联，这是因为：

第一个假设: **广告是否被用户看到只跟广告位置有关**，这个假设在广告场景是不合适的。因为他跟**广告**、以及**用户**的属性都有关系（广告大图、小图，用户个人的行为，还有context等）。

第二个假设: **用户看到广告后, 是否点击与广告位置无关**。这个实际上可能是有关的。比如在一个页面，用户同时看到了 位置1的广告和位置3的广告，但用户点击位置1的广告的概率更大。这其实还是position bias本身要解决的问题。

3. 实际应用结果

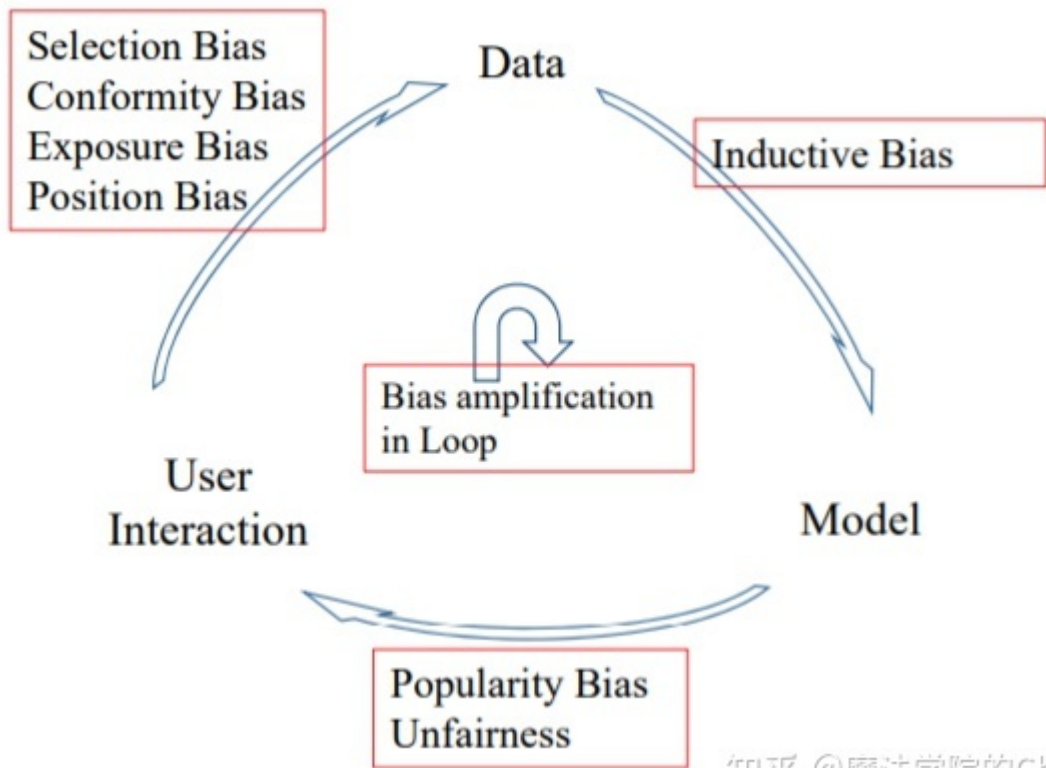
我们在不同场景下对这三种方法都有尝试。在我做的用户搜索场景，把position bias去掉之后，离线指标(auc)不可避免地会下降。在线上要取得短期指标上的收益也比较困难，因为在bias存在的情况下，一些流行的item会占据大部分流量、消费指标也很好；去掉bias之后，**长尾商品得到更多的曝光**，但是业务指标(如ctr)可能会下降。但是这样做对长期推荐系统的健康生态会有很大帮助。

4. 其他bias简介

推荐系统的bias是无处不在的，从user、data、model这三个推荐循环生态的角度出发，整体归纳起来大致有以下几个Bias：

- Position Bias：用户更倾向于和位置靠前的物品进行交互
- Exposure Bias：带标签的数据都是曝光过的，未曝光的数据无法确定其标签
- Selection Bias：用户倾向于给自己喜欢或者不喜欢的物品进行打分
- Conformity Bias：用户打分的分数倾向于和群体观点保持一致
- Popularity Bias：热门的物品获得了比预期更高的热度，长尾物品得不到足够曝光、马太效应严重
- Unfairness：因数据不均匀导致对某些弱势群体的推荐结果有偏

这些bias在推荐系统的反馈循环中会不断被加剧，导致推荐生态逐步恶化。



知乎 @魔法学院的Chilia

下一篇，我们将介绍美团在KDD cup 2020 Debiasing的解决方案。

[贝壳er：推荐系统中的bias&&debias\(二\)：_position bias的消偏](#)

[武侠超人：推荐生态中的bias和debias](#)