Optimizing Gross Merchandise Volume via DNN-MAB Dynamic Ranking Paradigm

IJCAI-2017, Workshop AI Applications in E-Commerce, 2017.

摘要

步入 Web 2.0 时代以后，人们的购物方式从传统线下实物购买向线上、移动平台购买转变。推荐系统在这样的购物场景中，起到了十分重要的作用，很大程度上可以促进消费，从而增长网站的 GMV （在线成交额）。本文提出了一种动态排序模型，DNN-MAB，使得推荐系统的结果可以更为精准，对用户更为贴切。DNN-MAB 是一个排序模型，由 DNN 作为一个（前）排序部件，再由 MAB 完成动态修正（后）排序的工作，可以将显示反馈和隐式反馈都考虑进来。这一模型也在实际场景（京东）得到了应用。

简介

为顾客生成个性化购物推荐列表，是网购平台中很重要的一项服务，直接影响用户的购物体验，也影响着购物平台的成交额。实际的场景是，挖掘用户的搜索、点击（隐式反馈）、下单（显式反馈）的历史记录，建立偏好特征，通过一个排序的模型，按照用户可能购买的概率高低给出一个尽可能精简的候选推荐列表。本文创造性地提出了一个动态排序模型，将 DNN 和多臂赌博机（MAB）算法结合起来，可以根据用户实时的操作反馈，及时调整候选推荐列表，从而提升整体的推荐准确率。

本文的主要贡献有二：

1. 创造性的提出了一种动态排序机制，pre-ranking 由 Pairwise 的深度神经网络完成，post-ranking 由 MAB 算法进行动态的调整；
2. 对 Thompson 采样算法进行了改进，使用新的参数初始化方法，从而适应实际的应用

问题建模

DNN-MAB 模型结构

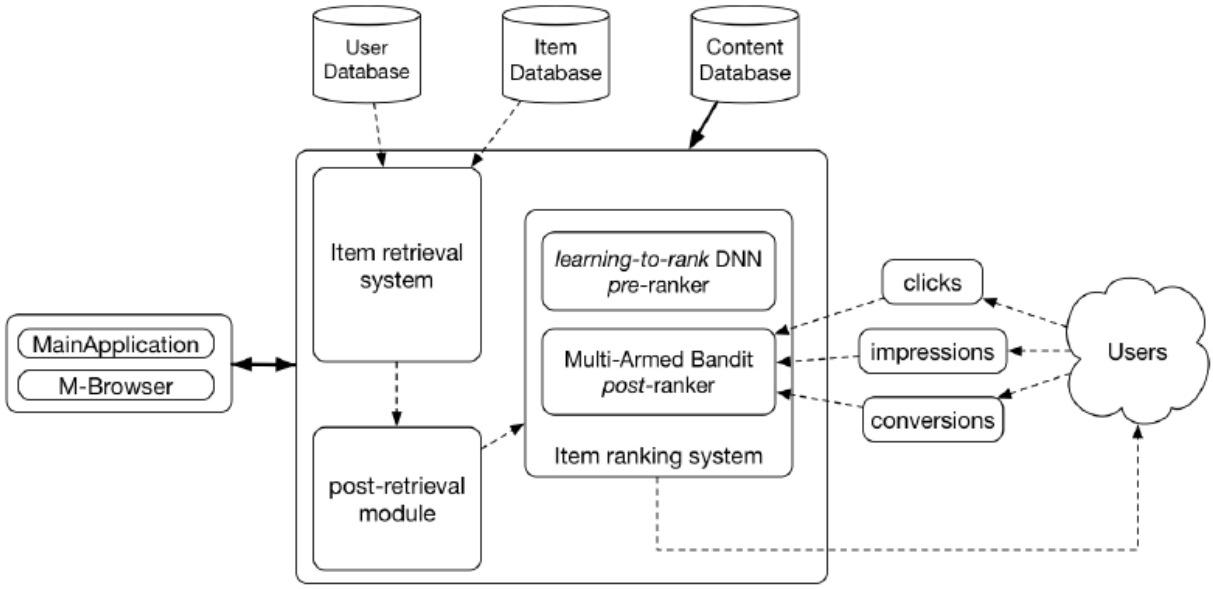

Figure 2: the recommender system flowchart

主要有三个部分：商品检索系统、过滤模块和商品排序系统。通过接收用户的信息，在后台进行信息汇总分析，从而将精心计算的候选推荐物品返回给用户，是经典的系统应用。流程如下：

- 系统获取用户的请求，同时获取用户的性别、地理位置信息、购买记录、价格敏感程度等信息，此外还有用户最近点击过的商品和购物车中的商品，均可以作为用户的输入特征；
- 商品检索系统根据这些信息，计算出一个与输入用户特征有关的较大的商品候选集；
- 通过图中的 post-retrieval 模块剔除一些明显不合适的物品（包含敏感信息、用户已经购买过的、差评较多的）；
- 本文的核心，即为对这个商品候选集进行筛选和重排序工作的两个 ranker，根据得分高低，得到最终的 K 个候选推荐物品。

优化目标

最终目标是最大化候选物品的 GMV 值，通过 dcg_K 来刻画目标：

$$dcg_K = \sum_{i=1}^{K} \frac{gmv_i \mathcal{I}(\xi_i)}{\log_2(i+1)}$$

这个目标中，gmv_i 可以简单理解为商品价格，ξ_i 表示某个商品，指示函数 $\mathcal{I}$ 用来标记用户对商品 i 是否购买。可见，这是一个 ranking-loss。

Pre-ranker

通过一个 Pair-wise 对称的DNN，共享商品的 embedding，通过最优化正负样本之间的 GVM 差值，第一次得到候选物品的打分，并且作为输入放入第二次排序过程中。概括来说，一方面使得与正样本（用户购买过的）相似的物品具有更高的可能性，此外，在通常的 Pair-wise 损失上乘一个正负样本之间的价格的权重，使得那些具有更高价格的商品能够有更高的机会出现在推荐列表中，因为本文提出的优化目标与 GVM 有关。

Post-Ranker: MAB

使用 MAB 的原因主要有三：

1. 用户实时的点击、购买的动作，应该比之前的历史的信息更为重要，这些在决策前的行为，对最终的决策有更大的影响；
2. 用户一定时间内没有反馈动作的情况，说明用户对当前品类的物品不感兴趣，这些物品在重排序过程中起的作用会更小；
3. 用户在比较同一品类下多个商品的行为，往往是最终购买的预示

使用 MAB 算法，可以尽可能减少对一些无关信息的关注，增强对用户相关行为的应用。

多臂赌博（老虎）机

给定具有 M 条臂的赌博机 $C_M = \{c_1, c_2, \cdots, c_M\}$ 和一个候选物品集合 $\mathcal{T}_N = \{x\}_N$ 及其在第一轮 pre-ranker 中的得分，其中每一个候选物品都有一条对应的臂 c_i。玩家（用户）在第 i 轮需要选择其中的一条臂，对应的收益是 $gvm_i\mathcal{I}(x_i)$，最终目标是最大化总体收益。

关于这种赌博机的介绍，可以看看这里。

收益计算

$$Reward = \sum_{i=1}^{K} gmv_{ji}\mathcal{I}(x_{ji})$$

这里的 $\{x_{j1}, x_{j2}, \cdots, x_{jK}\}$ 是通过完整的 DNN-MAB 给出的 K 个候选推荐物品。

具体算法

Algorithm 1 *post*-ranker: DNN-MAB Thompson sampling

1: **procedure** INITIALIZATION
2: **for each** $\langle \mathrm{x}, y \rangle \in \mathcal{T}_N$ **do**
3: for arm c such that $\mathrm{x} \in c$
4: $\alpha_c = \alpha_c + y$
5: $\beta_c = \beta_c + (1 - y)$
6: $\mathcal{U}_c \leftarrow \langle \mathrm{x}, y \rangle$
7: $avg_c = \alpha_c / |\mathcal{U}_c|_0$
8: **procedure** AT ROUND-i MAB RANKING
9: PULLING ARMS:
10: **for each arm** $c \in \mathcal{C}_M$ **do**
11: draw sample $r \sim \mathrm{beta}(\alpha_c, \beta_c)$
12: update all $y = y * (1 + r/\theta_1)$ for $\langle \mathrm{x}, y \rangle \in c$
13: pick $\langle \mathrm{x}_i, y_i \rangle = \arg\max_{(y, r_c)} \{\langle \mathrm{x}, y \rangle \in c\}$
14: $\mathcal{U}_c = \mathcal{U}_{c_i} - \langle \mathrm{x}_i, y_i \rangle$
15:
16: FEEDBACK:
17: **if** $\langle \mathrm{x}_i, y_i \rangle$ is exposed but not clicked **then**
18: $\mathcal{E}_c \leftarrow \langle \mathrm{x}_i, y_i \rangle$
19: $\beta_{c_i} = \beta_{c_i} + (1 - avg_{c_i}) * (1 - \exp(-\frac{|\mathcal{E}_{c_i}|_0}{\mathrm{SCALE}})) * \theta_2$
20: **if** $\langle \mathrm{x}_i, y_i \rangle$ is exposed and clicked **then**
21: $\mathcal{A}_{c_i} \leftarrow \langle \mathrm{x}_i, y_i \rangle$
22: $\alpha_{c_i} = \alpha_{c_i} + avg_{c_i} * (\frac{|\mathcal{A}_{c_i}|_0}{|\mathcal{E}_{c_i}|_0}) * \theta_3$

文中没有解释算法第 3 行 $x \in c$ 的含义，我的理解是，赌博机的臂大概是对应商品的品类（Category）

主要思想是，根据第一步 pre-ranker 的 DNN 得到的N 对物品和打分 x 和 y 作为初始化输入，在下面的第 i 轮博弈中，几个重要的参数说明如下

- $SCALE$ 是为了调整负样本（即没有发生过点击行为的）在整个排序学习过程中产生的影响；
- θ's 决定了参数每一次更新时的权重，类似学习率，对应最终的打分 y 会做出多大的调整；
- $\mathcal{U}_c$ 表示臂 c 下，还没有被用户选择物品，发生在用户行为之前，$\mathcal{E}_u$ 则表示呈献给用户但没有被点击，$\mathcal{A}_u$ 则是呈现且被点击的物品，$|\cdot|_0$ 是计算集合 size 的符号。

在第 i 轮赌博中，首先根据 beta 分布采样出 M 个样本，对应赌博机的每一条臂，计算每个臂上的 reward，选择最大的收益下对应的商品 x 及其对应的打分 y。如果在实际情况下， 用户点击了这个候选商品，就会更新 beta 分布在臂 c_i 的 α_{c_i} 参数，否则更新 β_{c_i} 参数。

实验与评价

本文偏工程，在进行 online a/b 测试前，先进行了模拟测试。选择了三种赌博机算法：ϵ-greedy、Upper Confidence Bound (UCB) 、Thompson sampling。其中第三种有分为原本的方式和算法 1 中的改进版本。

模拟实验

在模拟实验中，设置 $M=5$，表示每个赌博机有 5 条手臂，用户点击时，收益回报为 1，否则为 0；这里的点击是通过概率阈值 $f_{threshold}$ 控制的，在测试过程中对商品的点击生成一个概率值 f_{item}，如果比阈值大，则表示点击行为的发生。通过 10 组实验，经过 10000 次赌博，经过修改的 Thompson Sampling 赌博机算法能获得的收益是最大的，且模型的收敛速度是最快的。这是因为在这种机制下，初始化时考虑到了用户的信息（从 DNN 处获得）。

实验设置

在京东的真实场景下，进行了 a/b 测试，选取 7 天的数据。采用的评价标准有：GVM、订单量、总体和分页的 dcg（常用来衡量搜索引擎算法）。

$$dcg_{p,page-k}=\sum_{i=1,k\in page}^{i=p}\frac{gvm_{ki}\mathcal{I}(x_{ki})}{\log_2(i+1)}$$

且实际的候选物品展示，是分页显示的，所以这个分页的 dcg 是一个更为合理的指标。

实验结果

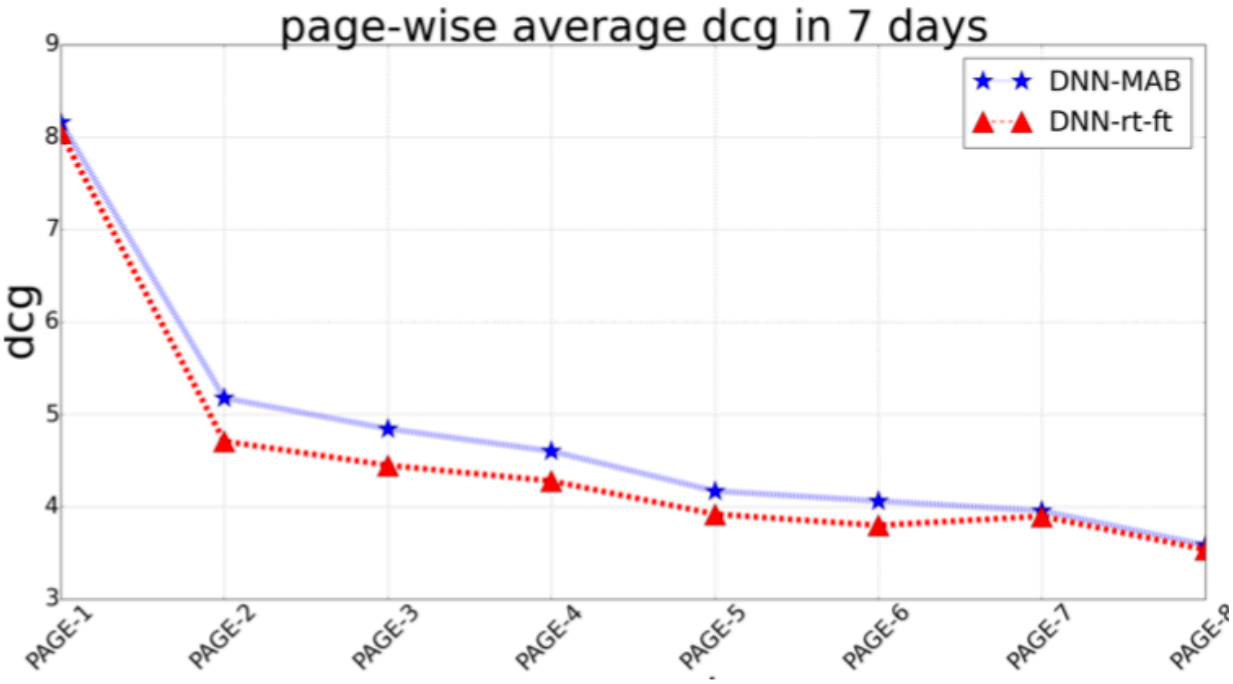

Figure 6: page-wise dcg gain: DNN-MAB v.s. DNN-*rt-ft*

Page	DNN-MAB	DNN-*rt-ft*	Gain
Page-0	8.164	8.046	+1.47%
Page-1	5.177	4.708	+9.96%
Page-2	4.844	4.448	+8.90%
Page-3	4.602	4.279	+7.55%
Page-4	4.171	3.920	+6.40%
Page-5	4.062	3.798	+6.95%
Page-6	3.957	3.897	+1.54%
Page-7	3.584	3.536	+1.34%

Table 4: page-wise dcg a/b test: DNN-MAB v.s. DNN-*rt-ft*

图中的 DNN-rt-ft 是指不通过 MBA 进行 post-ranking 的结果，是目前采用的方法。DNN-MAB 的带来的收益更高，可见带有强化学习思想的赌博机进行收益最大化的重排序是有效果的。

Date	Day1	Day2	Day3	Day4
GMV	+22.87%	+45.45%	+20.20%	+2.73%
Orders	-2.14%	-1.57%	+5.18%	+0.42%
Date	Day5	Day6	Day7	**Summary**
GMV	+0.91%	+23.15%	+1.50%	**+16.69%**
Orders	-2.79%	+4.19%	+2.20%	**+0.78%**

Table 1: GMV and orders gain / loss for DNN-MAB

Date	Day1	Day2	Day3	Day4
GMV	-12.08%	-9.33%	-4.74%	-3.24%
Orders	0.30%	-4.72%	-1.34%	-0.67%
Date	Day5	Day6	Day7	**Summary**
GMV	-18.31%	-7.49%	-1.43%	**-8.08%**
Orders	-10.67%	-4.69%	-0.81%	**-3.23%**

Table 2: GMV and orders gain / loss for DNN + *normal*-Thompson

另一方面，改进的 Thompson 采样方法，主要是在初始化阶段融合了一些用户的特征信息，不仅加快了收敛速度，提升模型的稳定性，也使得收益进一步提高。

总结与点评

本文提出了一个动态的排序框架：DNN-MBA，是一个深度神经网络作为前排序、多臂赌博机根据最大化收益对排序进行动态调整，从而完成了更精准的用户推荐，为电商平台带来收益。综合利用了 DNN 的优势：可以从历史数据（显式、隐式反馈）中抽象出更完整、准确的用户、商品特征；以及 MBA 博弈最大化收益的策略，捕捉用户短时兴趣的迁移，调整优化目标，从而对候选推荐列表根据预测的收益，进行重排序。与传统算法相比，该框架带来了更快的收敛速度及效果提升，在线上的真实流量实验中，推荐带来的成交额增长的幅度达到10+%。

本文的实验结果中，Table 1 显示第一、二天中出现了订单量减少的情况，可见如何保证复杂模型的稳定性和鲁棒性，提高关键性综合指标，仍然是一个很大的问题。

该文是比较偏工程的工作，从 MBA 中 Reward 的设计，可以看到强化学习的思想，直接最大化这个收益的方式，更新采样策略的分布参数。强化学习在采样中起到的作用还是很大的，尤其是涉及到正负样本的情况下。在推荐系统中，强化学习可以用来增强用户点击过的信息的强度，而那些没有交互的信息可以被进一步弱化。

对这里，直接优化 GVM 等指标的做法感觉十分生硬和投机，如果不是强化学习的帽子，可能真的不合适把最终的评价指标放在损失函数中。虽然也看到过对于二分类问题直接优化 AUC 的做法...阅读本文的另一小收获是，好的初始化方式，能够加快收敛速度和模型的性能，但必定不是每一处都适用。