Self-Play to Debias LLM

SPRec: Self-Play to Debias LLM-based Recommendation

1. Motivation

在基于大型语言模型（LLMs）的推荐系统中，现有方法主要通过有监督微调（SFT） 或直接偏好优化（DPO） 使模型适应推荐任务，但存在显著局限性：

- SFT 的缺陷：SFT 仅依赖正向样本（用户交互过的物品）训练，限制了模型对用户偏好的全面理解，难以捕捉细粒度的个性化需求。
- DPO固有的偏差：DPO 通过离线偏好排序数据（正向和负向样本对）显式对齐用户偏好，但研究发现其会加剧模型对少数物品的偏向性，具体如1.1和1.2所示。

1.1 偏差定义

- Token-level bias：由于模型通常通过最大化目标词元概率进行优化，包含常见token（如名含"the"的电影）的item可能被过度推荐，而忽略其实际用户相关性。
- Item-level bias: 这种类型的偏差来源就比较广泛了，在LLM时代之前就被深入研究过。在LLM中，item-level的偏差特指是经过微调后的大语言模型倾向于过度推荐热门项目（如蝙蝠侠系列电影）。这种现象会引发过滤Filter bubble，导致用户被局限在有限的流行内容范围内，既降低了推荐多样性，也损害了用户体验。

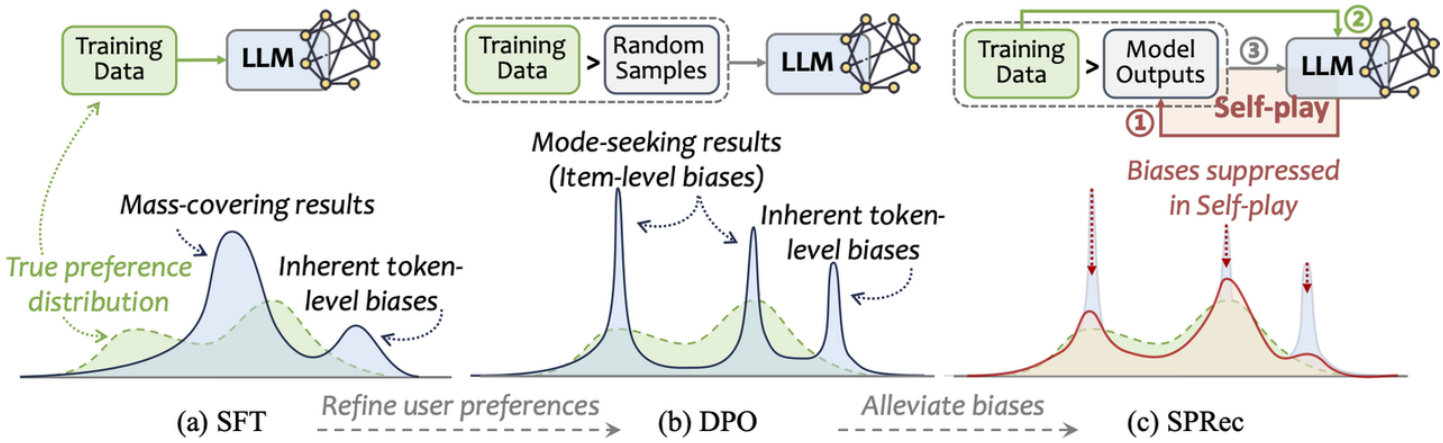
1.2 偏差度量

设 R 表示真实的用户偏好（例如项目列表），遵循 $P(R)$ 分布，而 $\hat{R}$ 表示LLM模型预测的偏好，来自 $P(\hat{R})$ 分布。偏差则通过这两个分布之间的不匹配来量化。其实就是：只要LLM推荐结果与训练数据的分布不一致，就是有偏。

具体偏多少呢？我们又可以分为item层面与item category层面的度量方式：

- Item层面的度量方式：用DivRatio与ORRatio，DivRatio即推荐的unique的item占总推荐次数的比例，比如推荐的列表为[1,1,2,3,4]，则DivRatio = 4/5=0.8。而ORRatio则是Over-recommendation的一个度量，此处简单定义为推荐结果中最频繁的3个item占所有推荐列表中的比例。
- Category层面的度量方式：这个比item层面的粒度要粗一些。可沿用WWW 24的一个metric：MGU@K，即推荐出来的K个商品，在item category层面上与历史数据的category的平均差异。如果MGU=0，代表推荐的商品在category层面上与历史数据的category分布完全一致。

1.3 偏差来源



SFT有一个较平缓的单峰，而DPO有一些比较“尖”的峰，两者都有一些内在的token-level bias。首先如何理解这里的SFT的大平峰与DPO的尖峰。这就要从两者的损失函数讲起。

SFT与DPO中的损失函数中，刚好也都有KL散度这一项。然而区别在于，SFT里的是Forward KL散度 $D_{KL}(p_D(y|x), \pi_\theta(y|x))$ ，而DPO里的Reverse KL散度 $D_{KL}(\pi_\theta||\pi_{\text{ref}})$ 。

- Forward KL散度强调的是mass covering，即模型的分布 π_θ 会尽量去覆盖数据分布 p_D 的所有区域（否则会导致KL项分母出现0，从而整项变为无穷）。这种优化方式的会生成更“全”的结果，但也可能在数据的低概率区域产生偏差。
- Reverse KL散度强调的是mode seeking，即去找数据的高概率区域去拟合，从而形成尖峰。

2. Prelimiary

2.1 SFT

为了使开源大语言模型（LLM）能够有效地学习推荐任务，一种可行的方法是使用离线推荐日志中的示范数据对其全部或部分参数进行微调。目标是通过在训练数据集D上最大化对数似然，使模型的行为与推荐任务保持一致。

$$\pi_{SFT} = \arg\max_{\pi_\theta} \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}} \log \pi_\theta(y_i|x_i), \quad (1)$$

其中 (x_i, y_i) 是来自D的输入-输出对， x_i 代表用户上下文和交互历史， y_i 代表目标项目。将 $p_D(y|x)$ 定义为经验概率（即项目流行度），监督微调（SFT）通过最小化正向KL散度来校准模型预测：

$$\begin{aligned}\pi_{SFT} &= \arg\min_{\pi_\theta} \mathbb{D}_{KL}(p_\mathcal{D}(y|x), \pi_\theta(y|x)) \\ &= \arg\min_{\pi_\theta} \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}} - log\pi_\theta(y_i|x_i) + H(p_\mathcal{D}), \quad (2)\end{aligned}$$

其中 $H(p_D)$ 是 p_D 的常数熵。 其实这也是behavior cloning的目标。现实中，由于SFT还不够好，我们还需要进一步偏好对齐，在推荐系统中，DPO肯定是一个很好的选择。

2.2 DPO

为确保模型输出符合复杂的用户偏好，研究人员提出了直接偏好优化DPO，该方法优化以下目标函数：

$$min_{\pi_\theta} - \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log\sigma\left[\beta\log\left(\frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}\right) - \beta\log\left(\frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}\right)\right],$$

其中 (x, y_w, y_l) 表示带有选定（偏好）答案 y_w 和拒绝答案 y_l 的提示 x 。参数 β 用作正则化因子，用于控制学习到的策略 π_θ 偏离参考策略 π_{ref} 的程度。在推荐任务的背景下，x 代表用户上下文，通常由用户特征和历史交互序列组成，而 y_w 和 y_l 分别对应正样本和负样本。目标是促使模型对更受偏好的物品 (y_w) 赋予比不太受欢迎的物品 (y_l) 更高的概率，从而有效地捕捉用户偏好。直接偏好优化（DPO）为偏好对齐提供了一种高效且稳定的解决方案，无需基于强化学习的方法中通常所需的复杂奖励模型。它能够自然地纳入正样本和负样本，这使其特别适合推荐系统，在推荐系统中，从对比用户交互中学习至关重要。

2.3 Evaluating Bias via Distribution Alignment

在对齐用户偏好时，大语言模型可能会无意中学习到有偏差或不公平的结果。为了评估语言推荐系统中的偏差和公平性问题，一个主流观点是将这些问题表述为分布不匹配问题。具体来说，设R表示真实用户偏好（例如，一个项目列表），遵循分布 $P(R)$ ，设 $\hat{R}$ 表示模型预测的偏好，其来自分布 $P(\hat{R})$ 。然后，偏差或不公平性通过这两个分布之间的不匹配来量化： $P(R) \neq P(\hat{R})$

3. SPRec Method

SPRec（Self-Play to Debias LLM-based Recommendation）是一种自博弈框架，无需额外数据或人工干预，通过迭代优化缓解过度推荐问题并提升公平性。核心思路是让模型 “与自身博弈”，利用前一轮的输出作为负样本，动态抑制偏差物品。

3.1 Solution: Suppress Biases through Self-Play

由于直接偏好优化（DPO）损失本质上会导致策略 π_θ 学习到尖锐的“峰值”，从而产生偏差，一个直观的解决办法是直接抑制这些习得的峰值。为了解决这个问题，采用SPDPO的自博弈框架，该框架在策略学习和偏差抑制之间交替迭代。具体来说，在第t+1次迭代中，从模型在第t次迭代时的预测分布 $\pi_{\theta_t}(\cdot|x)$ 中抽取负样本，从而产生以下学习范式：

$$\pi_{\theta_{t+1}} \leftarrow argmax_{\pi_\theta} \mathbb{E}_{(x,y_w)\sim D, y_l \sim \pi_{\theta_t}(\cdot|x)} l\left(\pi_\theta; \pi_{\theta_t}; x, y_w, y_l\right). (5)$$

通过将公式（5）与公式（4）进行比较，我们得到在\((t+1)\)次迭代中，目标函数SPDPO可以看作是由 $\frac{\pi_{\theta_t}(y_l|x)}{q_D(y_l|x)}$ 加权的CDPO，其可以表示如下：

$$\mathcal{L}_{SPDPO} = -\mathbb{E}_{(x,y_w)\sim\mathcal{D}, y_l\sim q_\mathcal{D}(\cdot|x)} \frac{\pi_{\theta_t}(y_l|x)}{q_\mathcal{D}(y_l|x)} l\left(\pi_\theta; \pi_{\theta_t}; x, y_w, y_l\right).$$

同样，如果直接偏好优化（DPO）使用来自离散均匀分布 $q_D(y|x) = U = \frac{1}{|I|}$ 的负样本，那么目标函数$SPDPO$可以看作是由 $\pi_{\theta_t}(y_l|x)$ 加权的直接偏好优化（DPO）。这突出表明，如果有偏差的项目在模型输出分布中具有更高的概率，该目标会通过提高它们的学习率，自适应地更加关注这些项目。

备注：与传统推荐方法预先定义负样本或提前分配权重不同，本文方法在学习过程中动态选择负样本。这具有显著优势，使模型能够自适应地调整其学习范式，以有效抑制偏差。因此，该方法减轻了过滤气泡问题，并提高了推荐的多样性。

3.2 Architecture of SPRec

利用公式(5)中的损失函数，提出了一种自博弈推荐调优框架SPRec，该框架通常包括多个监督微调（SFT）步骤和近端策略优化（DPO）步骤的迭代。工作流程如图3（c）所示，在每次迭代中依次执行三个关键步骤：

(1) 数据集构建：对于离线数据集中的每个正样本 (x^i, y_w^i)，通过运行当前模型 π_{θ_t}，并将其预测推荐作为 y_l^i，来采样一个负样本 y_l^i。这样，每个样本获得了成对偏好数据，即 (x^i, y_w^i, y_l^i)。

(2) 监督微调步骤：仅使用正样本 (x^i, y_w^i) 微调模型，通过最大化对数似然强化模型对用户真实偏好的对齐：$\pi_{SFT} = \arg\max_{\pi_\theta} \mathbb{E}_{(x_i,y_i)\sim\mathcal{D}} \log \pi_\theta (y_i|x_i)$

(3) 使用构建的偏好对 (x^i, y_w^i, y_l^i) 优化模型，目标是让模型对正向物品的打分高于负样本（前一轮的偏误预测）.

这个过程在每个时期重复T次迭代。自博弈机制适用于任何基于大语言模型的推荐系统。为确保与现有的基于直接偏好优化（DPO）的推荐器具有可比性，我们可以从单个负样本扩展到多个负样本，并在实验中分析结果。

关键机制

- 动态负样本：负样本来自模型前一轮的输出，使模型自适应地 “惩罚” 自身过度推荐的物品，抑制偏差。
- 迭代优化：通过多轮 SFT 与 DPO 的交替，平衡 “强化真实偏好” 与 “抑制偏误预测”，最终实现推荐准确性与公平性的提升。