Popularity-aware Distributionally Robust Optimization for Recommendation System

Jujia Zhao
zhao.jujia.0913@gmail.com
National University of Singapore

Wenjie Wang*
wenjiewang96@gmail.com
National University of Singapore

Xinyu Lin
xylin1028@gmail.com
National University of Singapore

Leigang Qu
leigangqu@gmail.com
National University of Singapore

Jizhi Zhang
cdzhangjizhi@mail.ustc.edu.cn
University of Science and Technology of China

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore

ABSTRACT

Collaborative Filtering (CF) has been widely applied for personalized recommendations in various industrial applications. However, due to the training strategy of Empirical Risk Minimization, CF models tend to favor popular items, resulting in inferior performance on sparse users and items. To enhance the CF representation learning of sparse users and items without sacrificing the performance of popular items, we propose a novel Popularity-aware Distributionally Robust Optimization (PDRO) framework. In particular, PDRO emphasizes the optimization of sparse users/items, while incorporating item popularity to preserve the performance of popular items through two modules. First, an implicit module develops a new popularity-aware DRO objective, paying more attention to items that will potentially become popular over time. Second, an explicit module that directly predicts the popularity of items to help the estimation of user-item matching scores. We apply PDRO to a micro-video recommendation scenario and implement it on two representative backend models. Extensive experiments on a real-world industrial dataset, as well as two public benchmark datasets, validate the efficacy of our proposed PDRO. Additionally, we perform an offline A/B test on the industrial dataset, further demonstrating the superiority of PDRO in real-world application scenarios.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation, Popularity, Distributionally Robust Optimization

ACM Reference Format:
Jujia Zhao, Wenjie Wang, Xinyu Lin, Leigang Qu, Jizhi Zhang, and Tat-Seng Chua. 2023. Popularity-aware Distributionally Robust Optimization for Recommendation System. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3583780.3615492

*Corresponding author. This research is supported by Huawei International Pte Ltd.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615492

1 INTRODUCTION

In the era of information explosion, extensive industrial applications employ recommendation systems to sift through vast amounts of items for personalized recommendations. Collaborative Filtering (CF) stands as the foremost method in the realm of recommendation. Typical, CF models learn the representations of users and items through historical user-item interactions and then rank item candidates for each user by the matching scores between the user and item representations [16]. However, due to the optimization by Empirical Risk Minimization (ERM), these CF models are more inclined to learn the representations of dense users and popular items with a larger number of interactions [41]. As such, they typically suffer from suboptimal performance when dealing with sparse users and items [4].

To tackle this issue, previous studies primarily fall into two categories: 1) Debiasing recommendation reduces the effects of popular items on model training by causal methods or intuitive strategies [5, 9, 14, 15, 35], balancing the representation learning between popular and sparse items. And 2) disentanglement learning endeavors to disentangle user preferences from item popularity, and then leverage the actual user preferences for recommendations [27, 51]. Despite their success, these works mainly regulate the effects of popular items while overlooking the effective representation learning of sparse users and items explicitly.

To improve the representation of sparse users and items, a recent work [41] applies Distributionally Robust Optimization (DRO) to the recommender learning process, aiming to emphasize the optimization of the worst group in the uncertainty set of the training data[1] [25, 34]. However, this work reveals two critical weaknesses by directly employing DRO for recommendation. First, it only considers the worse user groups and thus fails to simultaneously emphasize the representation learning on sparse items. Second, directly applying DRO might enhance the worst-case performance at the expense of popular items [31], degrading the overall recommendation accuracy. In light of these, our objective is to boost the representation learning of both the sparse users and items without sacrificing the popular ones.

[1] The uncertainty set denotes a family of pre-determined distributions encompassing both the empirical distributions and the potentially shifted distributions.

To this end, we introduce a Popularity-aware DRO (PDRO) framework, which additionally considers popularity in the DRO. Initially, we partition the user-item interaction pairs into groups by their loss values, where the loss values reflect the quality of the representations of users and items [3]. Subsequently, to improve the representation learning on sparse users and items while maintaining the performance of popular items, we incorporate item popularity into DRO via two modules: **implicit module** devises a new popularity-aware DRO objective, which pays more attention to the items that will potentially become popular over time. After the training with the popularity-aware DRO, an **explicit module** is designed to directly predict the temporal item popularity for estimating the user-item matching scores.

We implement PDRO on two competitive backend models, LightGCN [11] and ACVAE [42], covering the traditional CF model and sequential model. Besides, we collect a real-world recommendation dataset from the Huawei micro-video APP that covers rich user and item features. Rigorous experiments on this industrial dataset, along with two public datasets, validate the effectiveness of our proposed PDRO framework. Additionally, we conduct an offline A/B test in this micro-video recommendation scenario, revealing the superiority of PDRO over the previous recommender model. This new industrial Micro-video dataset is released along with our codes to facilitate recommendation session at https://github.com/Polaris-JZ/PDRO.

2 PRELIMINARY

• **Distributionally Robust Optimization** (DRO) aims to minimize the loss of the worst case in the uncertainty set $\mathcal{U}$ defined by the training data [21]. The objective of DRO can be defined as:

$$\theta^*_{\text{DRO}} := \arg\min_{\theta\in\Theta} \left\{ \max_{\mathbb{U}\in\mathcal{U}} \mathbb{E}_{(x,y)\in\mathbb{U}} [\, \mathcal{L}\,(\theta; (x, y))\,] \right\}, \tag{1}$$

where $\mathcal{U}$ represents a pre-determined family of training data distribution, and $\mathbb{U}$ denotes one possible distribution in $\mathcal{U}$. This objective function aims to pay more attention to optimizing $\mathbb{U}$ with the large loss, enhancing the generalization ability to possible testing data with distribution shifts.

Afterward, to avoid over-fitting some outliers with noisy labels, [34] proposes a novel approach called Group-DRO, which defines uncertainty set by dividing the distribution of training data into N subgroups. The objective function of Group-DRO is formulated as:

$$\theta^*_{\text{Group-DRO}} := \arg\min_{\theta\in\Theta} \left\{ \max_{\boldsymbol{w}\in\Delta_N} \sum_{i=1}^{N} w_i \mathbb{E}_{(x,y)\sim P_i} [\, \mathcal{L}\,(\theta; (x, y))\,] \right\}, \tag{2}$$

where w_i is the weight of group distribution P_i, $\boldsymbol{w}$ is a weight vector denoted as $[w_1, w_2, ..., w_N]$, and Δ_N is a $(N-1)$-dimensional probability simplex, satisfying that $w_i \geq 0$ and $\sum_{i=1}^{N} w_i$=1 [34]. The primary goal of Group-DRO is to enhance the worst-group performance by emphasizing the optimization of the groups with the large loss.

Although Group-DRO has the potential ability to learn better representations for sparse users and items, it may inadvertently compromise the performance of popular items [31]. Additionally, the performance of Group-DRO heavily relies on the grouping strategy since the optimization is performed at the group level.

3 METHOD

In order to enhance the representation learning of sparse users/items while maintaining the performance of popular items, we group user-item interactions by their loss values, and then integrate item popularity into the DRO framework via implicit and explicit modules. Next, we will detail the proposed PDRO framework.

3.1 Popularity-aware DRO

The implicit module enhances the DRO objective function via two key components: 1) a worst-group component that aims at emphasizing the optimization of the groups with the large loss, and 2) a popularity component that considers the items' popularity shifts over time, and pays more attention to the items that are likely to become popular in the future.

• **Group partition.** We employ the Group-DRO setting, which has demonstrated effectiveness in recommendation scenarios [41]. Due to the infrequent occurrence of sparse users and items in the training interactions, CF models fail to effectively learn their representations, resulting in elevated losses [41]. We justify this phenomenon via the empirical evidence in Figure 1, showing an inverse relationship between popularity and loss. Motivated by this observation, we perform group partition based on the losses of user-item interaction pairs in each training epoch, thereby ensuring that the group partition can be dynamically adjusted, enabling the model to focus on the sparse user-item groups that have inadequately learned representations with a large loss in each epoch.

We divide the user-item interactions into N groups, and further categorize them into T stages chronologically using the timestamps of user-item interactions. Consequently, each user-item pair is assigned a unique group and stage number. The new popularity-aware DRO objective is formulated as:

$$\theta^*_{\text{PDRO}} := \arg\min_{\theta\in\Theta} \sum_{n=1}^{N} w_n \mathcal{L}_n(\theta). \tag{3}$$

Here, $\mathcal{L}_n(\cdot)$ is the loss of group n, and w_n is the weight of group n calculated by the worst-group component and popularity component[2], that satisfies $w_n \geq 0$ and $\sum_{n=1}^{N} w_n = 1$. Let $\boldsymbol{w}$ be the weight vector denoted as $[w_1, w_2, ..., w_N]$, which is optimized by:

$$\boldsymbol{w} := \arg\max_{\boldsymbol{w}\in\Delta_N} \sum_{n=1}^{N} w_n [\ \underbrace{(1-\lambda)\cdot\mathcal{L}_n\,(\theta)}_{\text{(worst-group component)}} \underbrace{-\lambda\cdot\sum_{t=1}^{T}\sum_{j=1}^{N}\beta_t\mathcal{L}_j^t(\theta-\eta\nabla_\theta\mathcal{L}_n(\theta))}_{\text{(popularity component)}}], \tag{4}$$

where $\boldsymbol{w}$ is adjusted to maximize the summed loss, and assigned larger weights to the groups with larger loss due to $\sum_{n=1}^{N} w_n = 1$. Besides, the hyper-parameter λ is employed to balance the strength between the two components. Specifically,

- The worst-group component focuses more on the groups with the large loss, helping to improve the performance of sparse users and items.

[2]The gradient of w_n is excluded from participating in the model update process.

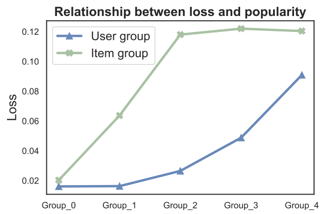


Figure 1: Relationship between loss and popularity. We divide users and items in Micro-video into 5 groups separately, and the popularity decreases gradually from group 0 to group 4.

- The popularity component pays more attention to the groups that can reduce the overall loss after one-step of gradient descent. $\mathcal{L}_j^t(\theta - \eta\nabla_\theta\mathcal{L}_n(\theta))$ signifies the loss of interactions in group j and stage t after one-step parameter updates by optimizing on group n. By considering the one-step parameter updates, we can discover the group n that can better minimize the overall loss. The negative sign preceding the popularity component indicates the desire to assign larger weights to the groups that can minimize the overall loss. Besides, $\beta_t = \exp(p \cdot t)$ controls the weights of stages, where β_t enables the model to concentrate on interactions in the later stages, and p is the hyper-parameter to modulate the magnitude of the weight discrepancy across distinct stages. Additionally, the summation of the weighted losses in the popularity component allows the model to implicitly inject popularity, as interactions involving popular items contribute to a higher proportion of the overall loss. By combining the summation process with the stage weight β_t, the model will prioritize the items that potentially become popular in the future,

However, directly applying Eq. (4) incurs a significant computational burden as it requires the system to consider the updated models under all possible group selection scenarios. To alleviate this computational complexity, we employ a First-order Taylor approximation to simplify Eq. (4) into a gradient-based formulation:

$$\boldsymbol{w} := \arg\max_{\boldsymbol{w}\in\Delta_N} \sum_{n=1}^{N} w_n \quad \underbrace{[(1-\lambda)\cdot\mathcal{L}_n(\theta)}_{\text{(worst-group component)}} \quad + \quad \underbrace{\lambda \cdot g_n \sum_{t=1}^{T}\sum_{j=1}^{N}\beta_t g_j^t}_{\text{(popularity component)}} \quad], \tag{5}$$

where g_n is the loss gradient of group n, and g_j^t is the loss gradient of the interactions from group j and stage t. Eq. (5) only needs to calculate the gradients for different groups, significantly reducing the computational burden.

The update of w_n depends on the worst-group component and popularity component. Following [41], we use a step size factor η to control the strength of the update:

$$w_n^e = w_n^{e-1} \cdot \exp(\eta \cdot ((1-\lambda)\cdot\mathcal{L}_n(\theta) + \lambda \cdot g_n \sum_{t=1}^{T}\sum_{j=1}^{N}\beta_t g_j^t)). \tag{6}$$

where w_n^e represents the weight of group n in the current batch e, and $\boldsymbol{w}^0$ follows a uniform distribution. Additionally, followed by [41], a streaming algorithm is employed to smooth the loss update for specific groups, mitigating the batch-to-batch variances:

$$\mathcal{L}_n^e \leftarrow \alpha \cdot \mathcal{L}_n^e + (1-\alpha)\cdot\mathcal{L}_n^{e-1}, \tag{7}$$

Table 1: Statistics of three datasets. #Inter./#user denotes the average number of interactions per user.

Dataset	#User	#Item	#Interaction	#Inter./#user	Density
Micro-video	25,871	44,503	210,550	7.40	1.66e-04
KuaiRand	22,128	7,076	621,064	22.42	3.17e-03
Amzon-book	64,907	88,027	1,991,329	25.02	2.84e-04

where α serves as the streaming factor controlling the smoothing degree between the group losses of two consecutive batches, and $\mathcal{L}_n^e$ represents the overall loss of group n in the current batch e.

To summarize, we implicitly incorporate popularity into Group-DRO, ensuring that DRO also focuses on the items that will potentially become popular in the future, instead of blindly enhancing the worst-group performance.

3.2 Explicit Module

In addition to implicitly injecting popularity into DRO, we also incorporate popularity directly to assist the model in calculating the user-item matching score. Inspired by [50], we use the stages divided by Section 3.1 to calculate the popularity of all items in each stage separately:

$$p_i^t = M_i^t / \sum_{k\in\mathcal{K}_t} M_k^t, \tag{8}$$

where p_i^t is the popularity of item i in stage t, M_i^t represents the interaction number of item i in stage t, and $\mathcal{K}_t$ represents all items in this stage t. After we get the item popularity in different stages, we employ a time-series forecasting method to predict the popularity:

$$p_i^t = p_i^{t-1} + \sigma(p_i^{t-1} - p_i^{t-2}), \tag{9}$$

where σ is the hyper-parameter that is used to adjust the popularity trend drift. Note that we only predict the popularity of the items with stage $t \geq 3$. For the items in the first and second stages, we use their own popularity calculated by Eq. (8) for the subsequent estimation. After getting item popularity, we estimate the user-item matching score by[3]:

$$s(u,i) = (1 + p_i^t)^\gamma \cdot s_o(u,i), \tag{10}$$

where $s_o(u,i)$ represents the matching score for user u and item i from the backend model optimized by PDRO, and γ is a hyper-parameter that adjusts the strength of the popularity factor. In this way, we effectively integrate item popularity into the user-item matching scores, thereby ensuring the performance of popular items. To maintain consistency in the training and testing procedures, we employ Eq. (10) for both phases.

4 EXPERIMENTS

In this section, we perform a series of experiments aimed at addressing the following research questions:

- **RQ1:** How does the performance of PDRO compare with other baselines across the datasets?
- **RQ2:** What is the impact of different components within the PDRO on overall performance?
- **RQ3:** How does the performance of PDRO in groups with different sparsity levels compare with that of backend models?

[3]To avoid the extremely small popularity of some sparse items significantly affecting the range of the final matching scores, we add one to the popularity score in Eq. (9) to regulate the magnitude of popularity.

Table 2: Overall performance of PDRO and other baselines. Bold signifies the best performance among the backend models, backend-DRO and backend-PDRO while underline represents the second-best model. The row of percentage improvement (% Improve.) quantifies the performance gain of PDRO in comparison to the second-best method. * denotes statistically significant improvements of PDRO over the backend models, according to the t-tests with a significance level of $p < 0.01$.

Models	Micro-video				KuaiRand				Amazon-book			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
NCF [12]	0.0801	0.1197	0.0584	0.0709	0.0740	0.1259	0.0498	0.0669	0.0233	0.0389	0.0156	0.0205
COR [38]	0.0677	0.1016	0.0473	0.0579	0.0722	0.1207	0.0481	0.0641	0.0268	0.0440	0.0180	0.0234
MultVAE [19]	0.0857	0.1286	0.0595	0.0735	0.0837	0.1385	0.0554	0.0738	0.0319	0.0507	0.0213	0.0272
MacridVAE [23]	0.0845	0.1243	0.0607	0.0737	0.0773	0.1308	0.0510	0.0687	0.0320	0.0509	0.0213	0.0273
BC-Loss [46]	0.0701	0.1162	0.0485	0.0630	0.0708	0.1204	0.0474	0.0636	0.0276	0.0442	0.0185	0.0237
InvCF [47]	0.0850	0.1293	0.0576	0.0722	0.0754	0.1287	0.0505	0.0680	0.0244	0.0399	0.0166	0.0215
LightGCN [11]	0.0888	0.1321	0.0593	0.0733	0.0741	0.1283	0.0496	0.0674	0.0285	0.0465	0.0188	0.0246
LightGCN-DRO [41]	0.0893	0.1300	0.0603	0.0734	0.0748	0.1299	0.0499	0.0679	0.0292	0.0481	0.0193	0.0252
LightGCN-PDRO	**0.0901***	**0.1356***	**0.0636***	**0.0780***	**0.0908***	**0.1494***	**0.0605***	**0.0798***	**0.0355***	**0.0539***	**0.0240***	**0.0299***
% Improve.	0.90%	2.65%	5.47%	6.27%	21.39%	15.01%	21.24%	17.53%	21.58%	12.06%	24.35%	18.65%
DIB [20]	0.0740	0.1106	0.0533	0.0650	0.0712	0.1237	0.0478	0.0650	0.0225	0.0374	0.0148	0.0194
CauseRec [49]	0.0638	0.0982	0.0439	0.0591	0.0534	0.0958	0.0398	0.0540	0.0254	0.0406	0.0171	0.0219
ACVAE [42]	0.0647	0.0910	0.0463	0.0546	0.0818	0.1382	0.0545	0.0730	0.0320	0.0487	0.0234	0.0285
ACVAE-DRO [41]	0.0629	0.0907	0.0474	0.0559	0.0828	0.1380	0.0553	0.0735	0.0320	0.0481	0.0231	0.0281
ACVAE-PDRO	**0.0674***	**0.0944***	**0.0493***	**0.0576***	**0.0904***	**0.1472***	**0.0590***	**0.0777***	**0.0338***	**0.0509***	**0.0238**	**0.0292***
% Improve.	4.17%	3.74%	4.01%	3.04%	9.18%	6.51%	6.69%	5.71%	5.62%	4.52%	1.71%	2.46%

• **Datasets.** We assess the effectiveness of PDRO on a real-world industrial dataset as well as two popular benchmark datasets.

1) **Micro-video** is an industrial dataset collected from the Huawei micro-video APP, which contains user-item interactions on extensive micro-videos for one month. Notably, this dataset stands out due to its abundant side information on users/items. In terms of user features, the dataset includes information on age, gender, and countries with different customs and religious backgrounds. As for the items, the dataset comprises diverse attributes, such as duration spanning from less than 2 minutes to 10 minutes, uploader details, and other relevant characteristics. Furthermore, there is a rich array of visual features associated with items, including videos and thumbnails with different resolutions. Additionally, there are diverse item textual features, containing titles and descriptions in multiple languages, are collected. Micro-video also captures abundant interaction behaviors between users and items, such as exposure, click, play, and like. In this work, positive samples are identified as micro-videos played by users for durations exceeding 8 seconds.

2) **KuaiRand**[4] is a popular video recommendation dataset, which contains diverse user feedback signals [6]. 3) **Amazon-book**[5] covers rich user interactions with books. We consider clicking behavior as positive samples in KuaiRand, and interactions with ratings ≥ 4 as positive samples in Amazon-book.

For the three datasets, we first sort the interactions chronologically according to the timestamps. And then we partition the sorted samples into training, validation, and testing sets by the ratio of 8:1:1. The statistics of datasets are shown in Table 1.

• **Evaluation.** We adopt full-ranking protocol [11] to evaluate the performance of all methods. Specifically, as for evaluation metrics, we employ Recall@K and NDCG@K for performance comparison, where $K = 10$ or 20 on three datasets.

[4]https://kuairand.com/.
[5]https://jmcauley.ucsd.edu/data/amazon/.

• **Baselines.** We compare our proposed PDRO with several competitive baselines, including non-sequential (NCF, COR, MultVAE, MacridVAE, BC-Loss, InvCF, and LightGCN) and sequential methods (DIB, CauseRec, and ACVAE). 1) **NCF** [12] leverages neural networks to learn the user-item interaction function. 2) **COR** [38] proposes a causal representations learning framework to enhance robustness under shifts of user features. 3) **MultVAE** [19] introduces a Variational Auto-Encoder (VAE) framework with multinomial likelihood. 4) **MacridVAE** [23] learns disentangled user representation learning at both intention- and preference-level. 5) **BC-Loss** [46] introduces a bias-aware margin into contrastive loss to alleviate the popularity bias. 6) **InvCF** [47] disentangles the popularity semantics and invariant user representations for robust prediction. 7) **LightGCN** [11] leverages high-order neighbors information to enhance the user and item representations. 8) **DIB** [20] employs information theory to disentangle representations into biased and unbiased parts. 9) **CauseRec** [49] integrates counterfactual thinking to synthesize user sequences. 10) **ACVAE** [42] proposes a VAE-based learning framework injected with contrastive learning and adversarial training. 11) **Group-DRO** [41] utilizes DRO to optimize the worst-group performance. We implement Group-DRO on both LightGCN and ACVAE.

• **Hyper-parameter Settings.** We instantiate PDRO on two competitive baselines of non-sequential and sequential methods (*i.e.,* LightGCN and ACVAE). The best hyper-parameters are selected based on the Recall of the validation set, with the searching scopes as follows: Group number N and streaming step size α are tuned in $\{2, 3, 5, 7\}$ and $\{0.1, 0.3, 0.5, 0.7\}$, respectively. We search the stage number T, strength factor λ, and stage importance factor p in the range of $\{2, 4, 6, 8, 10\}$, $\{0.1, 0.3, 0.5, 0.7\}$, and $\{0.2, 0.5, 1, 1.5\}$, respectively. The strength of popularity trend σ is selected in $\{0.1, 0.2, 0.4, 0.6, 0.8\}$, and the strength of popularity factor γ is searched in $\{1, 2, 3, 4, 5, 6\}$.

4.1 Overall Performance (RQ1)

The overall performance of PDRO and other baselines are shown in Table 2, from which we can observe that:

- The VAE-based methods (MultVAE, MacridVAE, and ACVAE) consistently yield competitive results on three datasets. It may be attributed to 1) the modeling of user preference over all items in the latent space, and 2) the consideration of random noises in the interaction generation process during training [32].
- LightGCN usually outperforms other baselines among non-sequential methods, which is attributed to the incorporation of high-order neighbors' information by graph propagation. As for sequential methods, ACVAE exceeds DIB and CauseRec in most cases. The reason may be that ACVAE employs contrastive learning and adversarial training, pushing the model to learn robust representations learning for sparse items.
- In most cases, the Group-DRO improves the performance of the backend model, which is ascribed to the optimization over the worst group. Nevertheless, the relatively minor improvements compared to the backend model are due to the sacrifice on the popular items (*cf.* Section 4.2.2). In contrast, PDRO significantly outperforms the backend model, validating the effectiveness of simultaneously considering sparse and popular items.

• **Offline A/B Test.** We also conduct an offline A/B test on Micro-video to assess the efficacy of PDRO within realistic industry scenarios [8]. The utilization of offline A/B test enables a comparative evaluation of the new system's performance against the current system in an industrial setting [8]. Furthermore, this approach offers notable advantages in terms of efficiency and cost-effectiveness. The objective of the A/B test is to estimate the potential uplift generated by our new system, denoted as π_t, compared to the current system, denoted as π_p. The current system π_p is trained on exposed data [6], while the new system π_t is trained on played data (*cf.* Section 4).

To prevent high variance and bias in the recommendation setting, we employ the capped importance sampling [8]. Specifically, we estimate the expected rewards R^{IS} using the following formula:

$$R^{IS} = \frac{1}{n} \sum_{(u,i,r \in S_n)} min(cap, \frac{\pi_t(i|u)}{\pi_p(i|u)})r. \tag{11}$$

Additionally, we compute the uplift as the difference between R^{IS} and the average reward over the exposure data:

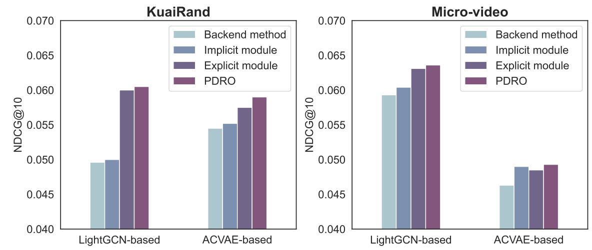
$$Uplift = R^{IS} - \frac{1}{n} \sum_{(u,i,r \in S_n)} r. \tag{12}$$

Here, S_n represents the distribution of exposure data, n is the total number of exposure data, and r indicates whether the user played with the item. $\pi_t(i|u)$ denotes the ranking score of user u and item i in the new system, while $\pi_p(i|u)$ represents the ranking score in the current system.[7] cap is the capping value, searched from $\{1\%t, 2\%t, 5\%t, 10\%t\}$, where $t = max(\{\frac{\pi_t(i|u)}{\pi_p(i|u)}|u, i \in S_n\})$. A positive uplift indicates the new system yields greater rewards than the current system. Furthermore, the magnitude of the uplift corresponds to the magnitude of the additional rewards obtained.

[6]In the exposed data, micro-videos exposed to users are considered as positive samples.
[7]In order to ensure that these two scores are comparable within the same range, we apply the sigmoid function to normalize them.

Table 3: Offline A/B test results.

Metrics	LightGCN	LightGCN-PDRO	ACVAE	ACVAE-PDRO
RIS	0.1967	0.2015	0.2077	0.2147
Uplift	-0.0019	0.0030	0.0149	0.0220



(a) Ablation study in KuaiRand (b) Ablation study in Micro-video

Figure 2: Performance of implicit module and explicit module compared with backend model and PDRO.

Following [8], we obtain the exposure data from Micro-video and compare R^{IS} and $Uplift$ between the backend model and PDRO [28]. From the results in Table 3, it is evident that PDRO exhibits superior performance compared to both current system π_t and the backend model, thus further confirming the effectiveness of PDRO in real-world industrial scenarios.

4.2 In-depth Analysis

4.2.1 ***Ablation Study (RQ2).*** We conduct an ablation study to assess the individual contributions of the implicit module and explicit module in the PDRO framework. The results are presented in Figure 2. From these results, we have the following findings: 1) The performance of PDRO declines when either the implicit module or explicit module is removed, which validates the effectiveness of each of these modules. 2) Discarding either one of the modules still outperforms the backend model, indicating the superiority of implicitly considering popular items in DRO and explicitly leveraging popularity in score ranking. 3) The effectiveness of the two modules is consistent across different datasets and backend models, further demonstrating their robustness and generalizability.

4.2.2 ***Group Performance Analysis (RQ3).*** In this section, we present a comprehensive evaluation of our model's performance across different user groups, as depicted in Table 4. As discussed in Section 3.1, the user groups are defined based on the average loss of their interactions, which ensures the group division *w.r.t.* different levels of sparsity. The results in Table 4 demonstrates the superior performance of our proposed method compared to both backend models with and without DRO among sparse and popular groups (Group 0 and Group 4, respectively).

4.2.3 ***Hyper-parameters Analysis (RQ2).*** We further select some sensitive hyper-parameters and vary them in the ranges presented in Section 4 and report the results in Figure 3. We can find that: 1) The selection of group number and stage number warrants careful consideration, as they determine the level of granularity of group and stage division in DRO. The representation learning for sparse items suffers from insufficient distinctions between a small number of groups (*e.g.,* 2 groups) while a too-large group number may result in training instability (*e.g.,* 8 groups), which

Table 4: Group evaluation results. The users are grouped according to the average loss of their interactions within the backend model. Specifically, Group 0 comprises users with the largest loss, while Group 1 consists of users with the second-largest loss, and so forth. Bold signifies the best performance among the listed models.

NDCG@20	Micro-video					KuaiRand				
	Group 0	Group 1	Group 2	Group 3	Group 4	Group 0	Group 1	Group 2	Group 3	Group 4
LightGCN	0.0550	0.0870	0.0834	0.0840	0.0679	0.0622	0.0673	0.0702	0.0709	0.0668
LightGCN-DRO	0.0545	0.0890	0.0825	0.0850	0.0673	0.0614	0.0679	0.0708	0.0717	0.0680
LightGCN-PDRO	**0.0630**	**0.0972**	**0.0845**	**0.0864**	**0.0689**	**0.0749**	**0.0786**	**0.0811**	**0.0815**	**0.0821**
ACVAE	0.0397	0.0613	0.0635	0.0766	0.0579	0.0787	0.0609	0.0656	0.0745	0.0856
ACVAE-DRO	0.0412	0.0627	0.0639	0.0769	0.0562	0.0783	0.0613	0.0658	0.0741	0.0852
ACVAE-PDRO	**0.0425**	**0.0636**	**0.0659**	**0.0831**	**0.0587**	**0.0814**	**0.0662**	**0.0665**	**0.0796**	**0.0896**

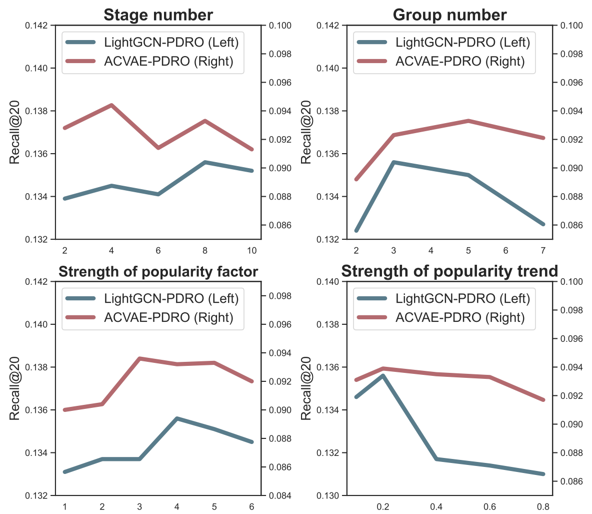


Figure 3: Hyper-parameters analysis of Micro-video.

is similar to the stage number. 2) In general, augmenting the strength of the popularity factor and trend improves performance, as incorporating popularity aids predictions on popular items. However, excessive emphasis on popularity risks may disregard user preferences, resulting in inferior performance.

5 RELATED WORK

• **Distributionally Robust Optimization.** DRO aims to minimize the loss of the worst case within a pre-determined uncertainty set, offering the potential to effectively learn the representations of sparse users/items [25, 26, 45, 52]. The exploration of DRO methods primarily revolves around defining the uncertainty set [3, 10, 33].

In order to mitigate noise within the uncertainty set, Group-DRO defines the uncertainty set as a combination of subgroups within the training set, focusing on optimizing group-level worst-case performance [13, 29, 34]. For instance, [41] reduces uncertainty in loss estimation by designing a streaming algorithm based on Group-DRO. Additionally, [31] selects subgroups that lead to the most significant reduction in global loss after gradient descent, rather than prioritizing the minority group. However, prior methods either focus on worst-case performance or ignore the popularity trend. To address this, there is a need to incorporate popularity information into DRO, maintaining the performance of popular items while improving the representation learning of the sparse users/items.

• **Debiasing Recommendation.** The objective of debiasing recommendation is to address inherent biases that exist in recommendation scenarios, such as popularity bias [2, 22, 36, 40, 43, 53, 54], exposure bias [17, 18, 24], and filter bubble issues [30, 37].

Existing approaches can be broadly categorized into two primary groups. 1) Intuitive methods [44, 53] address bias by suppressing the learning of popular items intuitively. For example, [44] considers oversampling popular negative samples in order to reduce popularity bias. 2) Causal methods [39, 50] apply causal technologies to eliminate the negative effect of item popularity. For instance, [50] proposes a Popularity-bias Deconfounding and Adjusting diagram, which removes the influence of popularity via causal intervention.

Similarly, Disentanglement Learning [7, 23, 27, 51] aims to disentangle a user's genuine interests to enhance the quality of recommendations and minimize the influence of diverse biases. For instance, [23] uses VAE to learn the disentangled user interests in order to facilitate effective recommendations. However, existing works focus on regulating the popularity of items, which neglects the representation learning of sparse users/items.

6 CONCLUSION AND FUTURE WORK

We proposed a Popularity-aware DRO (PDRO) framework to enhance the representation learning of sparse users and items while maintaining the performance of popular items. The PDRO framework incorporates item popularity into the DRO optimization process through two modules. In the implicit module, we devise a new DRO objective to implicitly consider popularity, ensuring the enhancement of sparse user/item representations while emphasizing the optimization of items likely to become popular in the future. Besides, the explicit module explicitly predicts item popularity and utilizes popularity to estimate user-item matching scores. To evaluate the effectiveness of PDRO, we conducted comprehensive experiments on an industry dataset and two public datasets. The results demonstrate the superiority of PDRO.

In future research, we plan to complete an online A/B test to assess the performance of PDRO. What's more, there are several promising directions to further enhance PDRO. For instance, advanced time-series forecasting methods can be explored to improve item popularity prediction. Additionally, utilizing a more flexible and fine-grained grouping method can enhance the robustness of PDRO. Lastly, the emerging direction of using large language models for recommendation [1, 48] has attracted extensive attention. It is promising to explore the effectiveness of DRO-based methods in enhancing the robustness of these large recommender models.

REFERENCES

[1] Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. In *RecSys*. ACM.

[2] Rocío Cañamares and Pablo Castells. 2017. A probabilistic reformulation of memory-based collaborative filtering: Implications on popularity biases. In *SIGIR*. ACM, 215–224.

[3] John Duchi and Hongseok Namkoong. 2018. Learning models with uniform performance via distributionally robust optimization. *arXiv:1810.08750*.

[4] Chenjiao Feng, Jiye Liang, Peng Song, and Zhiqiang Wang. 2020. A fusion collaborative filtering method for sparse data in recommender systems. *Inf. Sci.* 521 (2020), 365–379.

[5] Zuohui Fu, Yikun Xian, Shijie Geng, Gerard De Melo, and Yongfeng Zhang. 2021. Popcorn: Human-in-the-loop Popularity Debiasing in Conversational Recommender Systems. In *CIKM*. ACM, 494–503.

[6] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *CIKM*. ACM, 3953–3957.

[7] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal inference in recommender systems: A survey and future directions. *arXiv:2208.12397*.

[8] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *WSDM*. ACM, 198–206.

[9] Siyuan Guo, Lixin Zou, Yiding Liu, Wenwen Ye, Suqi Cheng, Shuaiqiang Wang, Hechang Chen, Dawei Yin, and Yi Chang. 2021. Enhanced Doubly Robust Learning for Debiasing Post-Click Conversion Rate Estimation. In *SIGIR*. ACM, 275–284.

[10] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. 2018. Fairness without demographics in repeated loss minimization. In *ICML*. PMLR, 1929–1938.

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

[12] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *WWW*. ACM, 173–182.

[13] Weihua Hu, Gang Niu, Issei Sato, and Masashi Sugiyama. 2018. Does distributionally robust supervised learning give robust classifiers?. In *ICML*. PMLR, 2029–2037.

[14] Jin Huang, Harrie Oosterhuis, Maarten De Rijke, and Herke Van Hoof. 2020. Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems. In *RecSys*. ACM, 190–199.

[15] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*, Maarten de Rijke, Milad Shokouhi, Andrew Tomkins, and Min Zhang (Eds.). ACM, 781–789.

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] Qian Li, Xiangmeng Wang, Zhichao Wang, and Guandong Xu. 2023. Be causal: De-biasing social network confounding in recommendation. *KDD*, 1–23.

[18] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *WWW*. ACM, 951–961.

[19] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. ACM, 689–698.

[20] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. Mitigating confounding bias in recommendation via information bottleneck. In *RecSys*. ACM, 351–360.

[21] Jiashuo Liu, Jiayun Wu, Bo Li, and Peng Cui. 2022. Distributionally robust optimization with data geometry. In *NeurIPS*. Curran Associates, Inc., 33689–33701.

[22] Zhongzhou Liu, Yuan Fang, and Min Wu. 2023. Mitigating popularity bias for users and items with fairness-centric adaptive recommendation. *ACM Transactions on Information Systems* 41, 3 (2023), 1–27.

[23] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*. Curran Associates, Inc., 5712–5723.

[24] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. 2021. A graph-based approach for mitigating multi-sided exposure bias in recommender systems. *TOIS* 40, 2 (2021), 1–31.

[25] Paul Michel, Tatsunori Hashimoto, and Graham Neubig. 2021. Modeling the second player in distributionally robust optimization. In *ICLR*.

[26] Paul Michel, Tatsunori Hashimoto, and Graham Neubig. 2022. Distributionally robust models with parametric likelihood ratios. In *ICLR*.

[27] Preksha Nema, Alexandros Karatzoglou, and Filip Radlinski. 2021. Disentangling Preference Representations for Recommendation Critiquing with ß-VAE. In *CIKM*. ACM, 1356–1365.

[28] Liqiang Nie, Leigang Qu, Dai Meng, Min Zhang, Qi Tian, and Alberto Del Bimbo. 2022. Search-oriented micro-video captioning. In *MM*. ACM, 3234–3243.

[29] Yonatan Oren, Shiori Sagawa, Tatsunori B Hashimoto, and Percy Liang. 2019. Distributionally robust language modeling. *arXiv:1909.02060*.

[30] Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you.* Penguin UK.

[31] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. 2022. Focus on the common good: group distributional robustness follows. In *ICLR*.

[32] Leigang Qu, Meng Liu, Jianlong Wu, Zan Gao, and Liqiang Nie. 2021. Dynamic modality interaction modeling for image-text retrieval. In *SIGIR*. ACM, 1104–1113.

[33] Hamed Rahimian and Sanjay Mehrotra. 2019. Distributionally robust optimization: A review. *arXiv:1908.05659*.

[34] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks for group shifts: on the importance of regularization for worst-case generalization. In *ICLR*.

[35] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as Treatments: Debiasing Learning and Evaluation. In *ICML*, Vol. 48. JMLR.org, 1670–1679.

[36] Harald Steck. 2011. Item popularity and recommendation accuracy. In *RecSys*. ACM, 125–132.

[37] Wenjie Wang, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. 2022. User-controllable recommendation against filter bubbles. In *SIGIR*. ACM, 1251–1261.

[38] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *WWW*. ACM, 3562–3571.

[39] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2020. Causal inference for recommender systems. In *RecSys*. ACM, 426–431.

[40] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *KDD*. ACM, 1791–1800.

[41] Hongyi Wen, Xinyang Yi, Tiansheng Yao, Jiaxi Tang, Lichan Hong, and Ed H Chi. 2022. Distributionally-robust recommendations for improving worst-case user experience. In *WWW*. ACM, 3606–3610.

[42] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *WWW*. ACM, 449–459.

[43] Guipeng Xv, Chen Lin, Hui Li, Jinsong Su, Weiyao Ye, and Yewang Chen. 2022. Neutralizing Popularity Bias in Recommendation Models. In *SIGIR*. ACM, 2623–2628.

[44] Hsiang-Fu Yu, Mikhail Bilenko, and Chih-Jen Lin. 2017. Selection of negative samples for one-class matrix factorization. In *SDM*. SIAM, 363–371.

[45] Runtian Zhai, Chen Dan, Zico Kolter, and Pradeep Ravikumar. 2021. DORO: distributional and outlier robust optimization. In *ICML*. PMLR, 12345–12355.

[46] An Zhang, Wenchang Ma, Xiang Wang, and Tat seng Chua. 2022. Incorporating Bias-aware Margins into Contrastive Loss for Collaborative Filtering. In *NeurIPS*. Curran Associates, Inc.

[47] An Zhang, Jingnan Zheng, Xiang Wang, Yancheng Yuan, and Tat-Seng Chua. 2023. Invariant Collaborative Filtering to Popularity Distribution Shift. In *WWW*. ACM, 1240–1251.

[48] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is ChatGPT Fair for Recommendation? Evaluating Fairness in Large Language Model Recommendation. In *RecSys*. ACM.

[49] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In *SIGIR*. ACM, 367–377.

[50] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *SIGIR*. ACM, 11–20.

[51] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2020. Disentangling user interest and popularity bias for recommendation with causal embedding. *arXiv:2006.11011* (2020).

[52] Chunting Zhou, Xuezhe Ma, Paul Michel, and Graham Neubig. 2021. Examining and combating spurious features under distribution shift. In *ICML*. PMLR, 12857–12867.

[53] Ziwei Zhu, Yun He, Xing Zhao, and James Caverlee. 2021. Popularity bias in dynamic recommendation. In *KDD*. ACM, 2439–2449.

[54] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee. 2021. Popularity-opportunity bias in collaborative filtering. In *WSDM*. ACM, 85–93.