Disentangling Interest and Conformity for Eliminating Popularity Bias in Session-Based Recommendation

Qidong Liu
Xi'an Jiaotong University
Feng Tian (fengtian@mail.xjtu.edu.cn)
Xi'an Jiaotong University
Qinghua Zheng
Xi'an Jiaotong University

Research Article

Keywords: Recommender System, Session-based Recommendation, Popularity Bias, Disentangling Learning

Posted Date: June 3rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1715647/v1

Disentangling Interest and Conformity for Eliminating Popularity Bias in Session-Based Recommendation

Qidong Liu[1], Feng Tian[1*] and Qinghua Zheng[1]

[1*]Faculty of Electronic & Information Engineering, Xi'an Jiaotong University, Xianning West Road, Xi'an, 710049, Shaanxi, China.

*Corresponding author(s). E-mail(s): fengtian@mail.xjtu.edu.cn;
Contributing authors: liuqidong@stu.xjtu.edu.cn;
qhzheng@mail.xjtu.edu.cn;

Abstract

Session-**B**ased **R**ecommendation (SBR) is to predict next item, given an anonymous interaction sequence. Recently, many advanced SBR models show great recommending performance, but few studies note that they suffer from popularity bias seriously: the model tends to recommend popular items and fails to recommend long-tail items. The only few debias works relieve popularity bias indeed. However, they ignore individual's conformity towards popular items and thus decrease recommending performance on popular items. Besides, conformity is always entangled with individual's real interest, which hinders extracting one's comprehensive preference. To tackle the problem, we propose a SBR framework with **D**isentangling **I**nte**R**est **A**nd **C**onformity (DIRAC) for eliminating popularity bias in SBR. In this framework, two group of item encoders and session modeling modules are devised to extract interest and conformity respectively, and a fusion module is designed to combine these two types of preference. Also, a discrepancy loss is utilized to disentangle representation of interest and conformity. Besides, our devised framework can integrate with several SBR models seamlessly. We conduct extensive experiments on two real-world datasets with three advanced SBR models. The results show that our framework outperforms other state-of-art debias methods consistently.

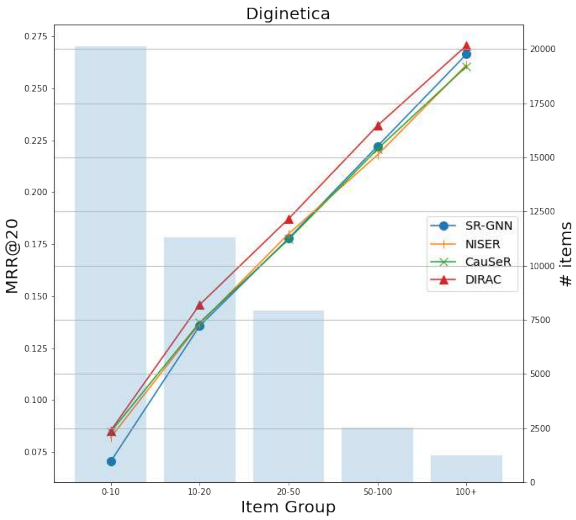


Fig. 1 An illustration of popularity bias in SBR. Items are grouped by their popularity in Diginetica dataset, which is commonly used in SBR experiment. Popularity is defined as the number of interaction of one item in dataset. The vertical axis represents mean reciprocal rank of SR-GNN and corresponding debias methods. The histogram shows the number of items in each item group.

Keywords: Recommender System, Session-based Recommendation, Popularity Bias, Disentangling Learning

1 Introduction

Session-**B**ased **R**ecommendation (SBR) has attracted much attention, because of its wide usage in numerous application fields, such as e-commerce[1], music[2] and so on. Recently, many SBR works, equipped with deep learning techniques, promote recommending performance to a high level[3, 5, 6, 8]. However, existing of popularity bias in SBR models leads to a poor performance on those tail items and we conduct an experiment on a real-world dataset Diginetica to illustrate it. We train a state-of-art SBR model SR-GNN[6] on the dataset and report mean reciprocal rank(MRR) in the top-20 recommendation list, where items are grouped by its popularity. Results are shown in Figure 1: more popular items get better recommending performance, which reveals severe popularity bias in SBR. At the same time, popularity bias will make user's recommendation list full of popular items, which causes filter bubble[7] and harms long-term profit for both of users and platforms. Therefore, eliminating popularity bias for SBR is extremely vital.

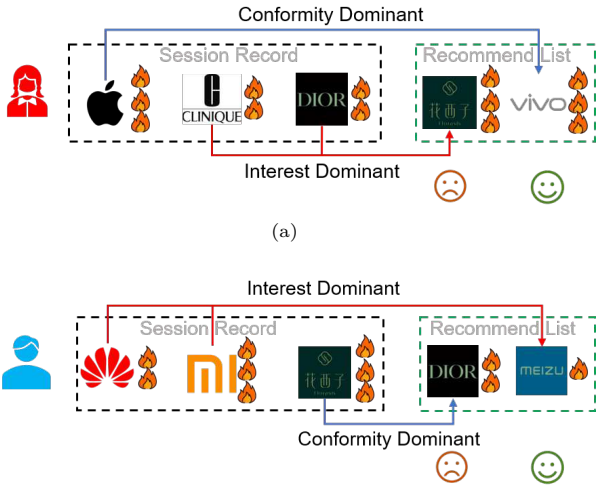


Fig. 2 Examples of how interest and conformity affect user's satisfaction.

Recent studies related to popularity bias achieve some success in the field of **C**ollborative **F**iltering (CF). They elevate recommending performance on tail items by various techniques, such as re-weighting[9] and causality[10]. Nevertheless, as far as we know, only three works[11–13] focus on popularity bias in SBR. Liu et.al.[11] devises a preference mechanism to scale recommending probability of tail and popular items respectively. Gupta et.al.[12] normalizes item and session embedding for SR-GNN[6] to promote recommending performance of tail items. Another most recent work[13] utilizes causality technique to eliminate popularity bias for SR-GNN. Though these works relieve popularity bias for SBR well, they ignore individual's conformity towards popular items and thus decrease recommending performance on popular items. As Figure 1 shows, NISER[12] and CauSeR[13] elevate recommending performance on long-tail item group, but the performance on 50-100 and 100+ item group degrades compared with SR-GNN. Besides, these methods are designed for SR-GNN only, but we aim to explore a general method for more SBR models.

As is well known, the key of SBR is to capture user preferences[14]. Individual preference always contains both of conformity and interest, which are frequently mentioned in recommendation research field, and are focus in this paper. In detail, conformity is a tendency that users rate homogeneously to others in a group, whose existence has been proved in [15, 16]. However, interest is one's attitude to item itself[17]. Therefore, we can assume that someone clicks or buys one item due to various degree of interest and conformity. To illustrate it, we can see the example in Figure 2. On the one hand, if we always recommend popular related items, we find that it does not always work from

example of the female user. She feels satisfied with the recommended popular electronic because conformity is dominant in her preference on electronic. However, her preference on makeup is led by interest, so the popular but a little unrelated makeup cannot satisfy her. The phenomenon proves that overestimation of popularity effects causes popularity bias. On the other hand, if we intentionally recommend more unpopular items, the results are not always pleased too. We can see example of the male. He is content with the unpopular but related electronic, because interest mainly leads to his preference. Conversely, the unpopular makeup disappoints him due to his huge conformity on makeup, which illustrates that ignoring of conformity causes poor recommending performance on popular item group, as Figure 1 shows. Besides, from the example, it can be concluded that interest and conformity are always entangled, which hinders precise estimation of individual's preference.

However, most of current studies extract preference regardless of entanglement between interest and conformity, so we propose a SBR framework with **D**isentangling **I**nte**R**est **A**nd **C**onformity (DIRAC) to tackle this problem. Firstly, an item encoding module is applied to encode item id into interest embedding and conformity embedding respectively. Then, an interest modeling module and a conformity modeling module are devised for extracting interest and conformity preference. Especially, conformity modeling module is designed as an encoder-decoder structure to help modeling conformity. At last, we design a gate fusion module to fuse preference of interest and conformity adaptively. In summary, the main contribution of this paper can be concluded as follows:

- We model and disentangle interest and conformity for session-based recommendation models, which is much useful to eliminate popularity bias for SBR models and avoid nuisance for popular items.
- We propose a general framework for session-based recommendation to eliminate popularity bias, which can integrate with most of SBR models.
- We have conducted extensive experiments on two real-world datasets with three SBR models to evaluate the effectiveness of the proposed DIRAC. The results show the superiority of DIRAC over other state-of-art methods.

The rest of paper is organized as follows. Section 2 reviews related works about session-based recommendation models and techniques to eliminate popularity bias for recommender system. Then, section 3 describes proposed DIRAC in detail. Experiments and result analysis are shown in section 4. At last, we conclude this paper and look forward to the promising future work in section 5.

2 Related Work

In this section, we first review the session-based recommendation models, then summarize current progress of eliminating popularity bias for recommender system.

2.1 Session-Based Recommendation

Session-based recommendation (SBR) models can be categorized into three groups, i.e., traditional models, RNN-based models and GNN-based models. Firstly, traditional models also have two types. One focus on finding out the most similar item or session to current session, such as STAN[18] and WH[19]. The other type of models regard transitions over items within a session as a Markov Chain. For example, FPMC[20] combines Markov chain with matrix factorization to learn a transition probability matrix for each user. PME[21] computes transition probability based on Euclidean distances between users and items. Secondly, as deep learning broadcasts to several fields, GRU4Rec[3] firstly applies Recurrent Neural Network (RNN) to SBR, which shows brilliant performance. Later, some RNN-based models, like STAMP[4] and NARM[5], adopt attention mechanisms to model long-term and short-term preference of a session. Lastly, GNN-based models take complex transitions of items into account to promote SBR performance by using Graph Neural Network (GNN). SR-GNN[6] is the first proposed model in this category, and some works follow it applying more elaborate graph techniques, such as DHCN[22] and HG-GNN[23].

Most of RNN-based SBR and GNN-based SBR models can be concluded into one SBR framework, which will be illustrated in section 3. Based on it, we propose a general debias framework to eliminate popularity bias for many SBR models.

2.2 Popularity Bias

Recently, popularity bias in recommender system attracts much attention[24]. A bunch of methods to eliminate popularity bias is re-weighting method, which re-weights each instance according to inverse propensity score (IPS)[9, 25]. These methods impose a higher weight on tail items during training process and can guarantee zero expected value of bias. However, high variance of IPS is harmful, so some works[26, 27] focus on decreasing it to promote performance. Another bunch of works regard popularity bias as a problem of spurious effects in causality field. PD[29] and DecRS[30] both adopt back-door criterion[28] to eliminate confounder bias. MACR[31] designs a multi-task learning method to calculate direct effects of users and items. Perhaps the closet work to ours is DICE[10], which consider user's interest and conformity as cause of click. However, they use causality technique to disentangle interest and conformity, which is totally different from us. Also, all of methods mentioned above, including DICE, aim at eliminating popularity bias for collaborative filtering, and cannot be applied to SBR directly.

To best of our knowledge, only three works focus on popularity bias in SBR. Liu et.al[11] proposes TailNet, which imposes a scale factor on recommending probability of tail items. NISER[12] adds a normalized regularization on item embedding and session embedding. As for CauSeR[13], it adopts do-calculus[28] to eliminate bad causal effects. Though, these works perform well

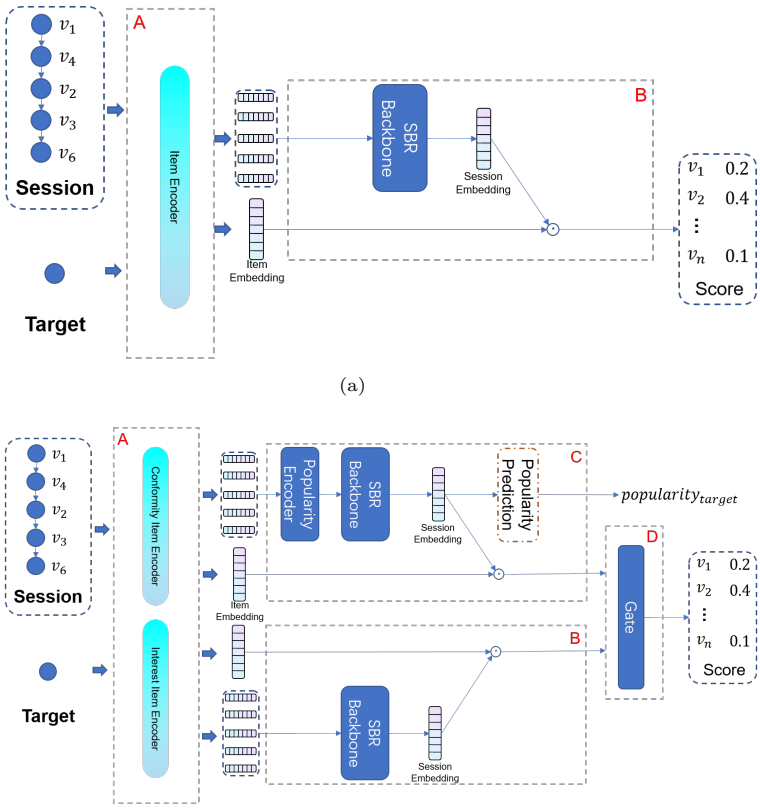


Fig. 3 (a)The architecture of general SBR models, which contains two parts: A.Item Encoder; B.SBR Backbone; (b)The architecture of proposed framework, which contains four parts: A.Item Encoding Module; B.Interest Modeling Module; C.Conformity Modeling Module; D.Fusion Module

on eliminating popularity bias, they do not consider users' conformity, which decreases recommending performance on popular items.

3 Method

As Figure 3(a) shows, most of SBR models can be divided into two parts: one is item encoder, which transforms item id into dense embedding; the other is SBR backbone, which generates session embedding as representation of individual's preference. By comparison, there are four modules in our proposed framework, as Figure 3(b) shows: Item Encoding Module, Interest Modeling Module, Conformity Modeling Module and Fusion Module. In this section, we will illustrate the proposed framework according to these four modules in detail.

3.1 Notations

The task of session-based recommendation is to predict the next item that user will interact with, only based on user's current interactive sequence. In such a task, let $V = \{v_1, v_2, ..., v_N\}$ denotes the set of items that appear in all sessions, where N represents the number of items in the dataset. The current session m can be defined as $S_m = [v_{m,1}, v_{m,2}, ..., v_{m,n}]$, where $v_{m,i} \in V$. The aim of the task is to predict the recommending probability $\hat{y}_m$ for all candidate items. According to the list of probability $\hat{y}_m = \{\hat{y}_{m,1}, \hat{y}_{m,2}, ..., \hat{y}_{m,N}\}$, we can give out the recommendation list.

3.2 Item Encoding Module

To disentangle interest and conformity preference of a session, we apply two sets of item encoders to encode item id into embedding. One is Interest Item Encoder, which is to capture user's interest, and the other is Conformity Item Encoder. The two encoders have identical structure, but own different parameters. Therefore, each item and each session have two kinds of embedding:

$$\begin{aligned} \mathbf{x}_i^{con} &= Encoder_{con}(v_i) \\ [\mathbf{x}_{m,1}^{con}, \mathbf{x}_{m,2}^{con}, ..., \mathbf{x}_{m,n}^{con}] &= Encoder_{con}([v_{m,1}, v_{m,2}, ..., v_{m,n}]) \end{aligned} \tag{1}$$

$$\begin{aligned} \mathbf{x}_i^{int} &= Encoder_{int}(v_i) \\ [\mathbf{x}_{m,1}^{int}, \mathbf{x}_{m,2}^{int}, ..., \mathbf{x}_{m,n}^{int}] &= Encoder_{int}([v_{m,1}, v_{m,2}, ..., v_{m,n}]) \end{aligned} \tag{2}$$

where $\mathbf{x}_i^{con}$ and $\mathbf{x}_i^{int}$ denote the conformity and interest embedding vector of item i respectively.

As mentioned above, item encoder of different type of SBR model varies. For RNN-based SBR, the structure of encoder is always embedding layer. It embeds each item $v \in V$ into an embedding vector $\mathbf{x} \in \mathbb{R}^d$, where d is the dimensionality. By contrast, GNN-based SBR models adopt an embedding layer and following an GNN layer to get item embedding vector, such as GGNN[34] for SR-GNN[6].

3.3 Interest Modeling Module

The Interest Modeling Module aims to model individual's real interest via interaction sequence in current session. Firstly, we apply a SBR backbone to encode the series of item embedding vectors in current session into a session embedding, which represents interest preference of current session:

$$\mathbf{S}_m^{int} = Backbone_{int}([\mathbf{x}_{m,1}^{int}, \mathbf{x}_{m,2}^{int}, ..., \mathbf{x}_{m,n}^{int}]) \tag{3}$$

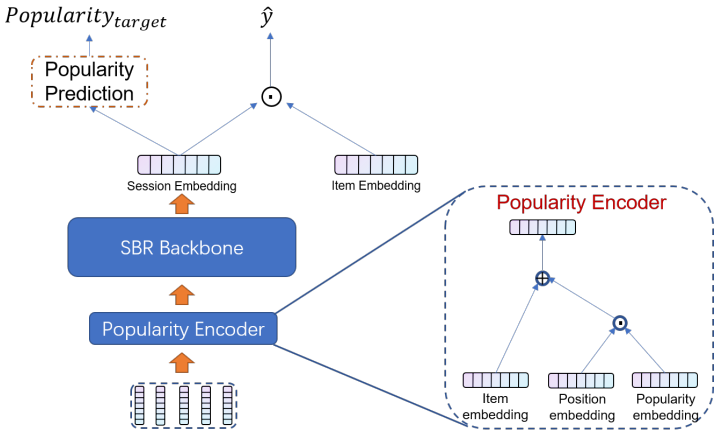


Fig. 4 The architecture of Conformity Modeling Module.

where $\mathbf{S}_m^{int}$ is the interest session embedding for session m. Note that *Backbone* represents the part of models that computes session embedding from a series of item embedding in SBR models. For example, GRU layer is the *Backbone* of GRU4Rec[3].

To balance magnitude of learned embedding of popular and long-tail items[12], then we normalize target item embedding and session embedding before calculate the recommending probability score.

$$\tilde{\mathbf{x}}_i^{int} = \frac{\mathbf{x}_i^{int}}{\|\mathbf{x}_i^{int}\|_2}, \ \tilde{\mathbf{S}}_m^{int} = \frac{\mathbf{S}_m^{int}}{\|\mathbf{S}_m^{int}\|_2} \tag{4}$$

$$\hat{y}_{m,k}^{int} = \frac{exp(\mu \tilde{\mathbf{x}}_k^{int^T} \tilde{\mathbf{S}}_m^{int})}{\sum_{j=1}^{n} exp(\mu \tilde{\mathbf{x}}_j^{int^T} \tilde{\mathbf{S}}_m^{int})} \tag{5}$$

where $exp(\cdot)$ is exponential function, and μ is a scaling factor for better convergence. Besides, $\hat{y}_{m,k}^{int}$ denotes the recommending probability of item k for session m from interest preference.

3.4 Conformity Modeling Module

Next, we introduce the Conformity Modeling Module to model individual's conformity preference in current session and the detailed structure is shown in Figure 4. Inspired by popularity-based news recommendation[35], popularity and position information can be utilized to help model user's conformity preference. Therefore, we propose a Popularity Encoder to inject popularity and position information into session embedding:

$$\mathbf{x}_i^{'con} = \mathbf{x}_i^{con} + \mathbf{pop}_i \odot \mathbf{pos}_i \tag{6}$$

where $\odot$ denotes the Hadamard product. Besides, $\mathbf{pop}_i \in \mathbb{R}^d$ and $\mathbf{pos}_i \in \mathbb{R}^d$ are popularity embedding and position embedding of item v_i respectively,

where d is the dimensionality. In detail, they are mapped from discretized popularity and position value via embedding layer.

Further to ensure session embedding encoding conformity preference, we propose an encoder-decoder structure in this module. The series of item embedding in current session are encoded by SBR backbone, which is so-called encoder, and output the session embedding which contains individual's conformity preference:

$$\mathbf{S}_m^{con} = Backbone_{con}([\mathbf{x}_{m,1}^{'con}, \mathbf{x}_{m,2}^{'con}, ..., \mathbf{x}_{m,n}^{'con}]) \tag{7}$$

where $\mathbf{S}_m^{con}$ is the conformity session embedding for session m. Note that $Backbone_{int}$ and $Backbone_{con}$ have identical structure but with different parameters.

Then, if the session embedding contains conformity preference, it can be used to predict the popularity of target item. Therefore, we devise the decoder as a feed-forward network which aims to predict popularity of the target item:

$$\hat{pop}_{target} = \mathbf{W}_2(\mathbf{W}_1\mathbf{S}^{con} + b_1) + b_2 \tag{8}$$

where $\hat{pop}_{target}$ denotes predicted popularity of the target item. $\mathbf{W}_1 \in \mathcal{R}^{d\times d}$, $\mathbf{W}_2 \in \mathcal{R}^{1\times d}$, $b_1 \in \mathcal{R}^{d\times 1}$ and $b_2 \in \mathcal{R}^1$ are all parameter matrix.

Finally, we normalize the item embedding and session embedding, and calculate recommending probability of conformity preference via softmax:

$$\tilde{\mathbf{x}}_i^{con} = \frac{\mathbf{x}_i^{con}}{\|\mathbf{x}_i^{con}\|_2},\ \tilde{\mathbf{S}}_m^{con} = \frac{\mathbf{S}_m^{con}}{\|\mathbf{S}_m^{con}\|_2} \tag{9}$$

$$\hat{y}_{m,k}^{con} = \frac{exp(\mu\tilde{\mathbf{x}}_k^{con^T}\tilde{\mathbf{S}}_m^{con})}{\sum_{j=1}^{n} exp(\mu\tilde{\mathbf{x}}_j^{con^T}\tilde{\mathbf{S}}_m^{con})} \tag{10}$$

where $\hat{y}_{m,k}^{con}$ denotes the conformity recommending probability of item k for session m. At last, $\hat{y}_{m,k}^{con}$ and $\hat{y}_{m,k}^{int}$ will be input into fusion module to get the recommending probability of each item.

3.5 Fusion Module

Here, we adopt a fusion gating mechanism to combine interest preference with conformity preference. The significance of interest and conformity preference vary in different session, so we calculate the weight for recommending probability of interest and conformity preference via a gate fusion mechanism:

$$g = \sigma(\mathbf{W}_3[\mathbf{S}^{int};\ \mathbf{S}^{int} \odot \mathbf{S}^{con};\ \mathbf{S}^{con}] + b_3) \tag{11}$$

where $\sigma(\cdot)$ is sigmoid function and $\mathbf{W}_3 \in \mathbb{R}^{1\times 3d}$, $b_3 \in \mathbb{R}^1$ are learnable parameters. g denotes the weight for recommending probability of interest preference.

At last, recommending probability of interest and conformity preference can be fused as follow:

$$\hat{y}_m = g \cdot \hat{y}_m^{int} + (1 - g) \cdot \hat{y}_m^{con} \tag{12}$$

where $\hat{y_m}$ is the recommending probability for session m.

3.6 Train Loss

For the task of recommendation, we use cross-entropy loss as main loss function, which can be written as follow:

$$\mathcal{L}_{click} = -\sum_{i=1}^{N} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \tag{13}$$

As for auxiliary task in Conformity Modeling Module, which aims to predict popularity of target item, the loss function is mean square error (MSE):

$$\mathcal{L}_{auxiliary} = \|\hat{pop} - pop\|^2 \tag{14}$$

where *pop* is the true popularity of the target item.

Also, according to the field of disentangle learning[10, 36], we impose direct supervision on the distribution of item embedding and session embedding to disentangle interest and conformity. For a batch of training data, we take out embedding of unique item as a set, denoted as $\mathbf{X}$, and denote a batch of session embedding as $\mathbf{S}$. Therefore, the discrepancy loss function can be written as:

$$\mathcal{L}_{discrepancy} = \|\mathbf{X}^{int} - \mathbf{X}^{con}\|^2 + \|\mathbf{S}^{int} - \mathbf{S}^{con}\|^2 \tag{15}$$

In a word, we can optimize the whole framework on sum of the above three loss functions:

$$\mathcal{L} = \mathcal{L}_{click} + \alpha \cdot \mathcal{L}_{auxiliary} + \beta \cdot \mathcal{L}_{discrepancy} + \lambda \cdot \|\theta\|^2 \tag{16}$$

where α, β, λ are hyper-parameters, and θ is the set of all learnable parameters.

4 Experiment

4.1 Datasets

We conduct experiments on the following two real-world e-commerce datasets, i.e. *Diginetica*[1] and *RetailRocket*[2]. The Diginetica dataset was published in CIKM Cup 2016, which collects anonymous logs and transactions. The Retail-Rocket dataset came from a Kaggle contest, which contains users' behavior

[1] https://competitions.codalab.org/competitions/11161#participate
[2] https://www.kaggle.com/retailrocket/ecommerce-dataset

Table 1 Statistics of datasets used in experiments

Dataset	# train	# test	# item	avg.len
Diginetica	719470	60858	43097	5.70
RetailRocket	716878	27704	48971	3.83

data in a real-world e-commerce website. We use transactional data and view log in Diginetica and RetailRocket respectively.

4.1.1 Preprocessing

Following previous works[6, 32], we filter out items appearing less than five times and remove sessions with length shorter than two. Next, we set session data of last week as test data, and remaining data as train data. Furthermore, we conduct a data augmentation to generate sequences and corresponding labels by splitting the input sequence. In detail, for each session $S = [v_1, v_2, ..., v_n]$, we generate $n-1$ sessions with labels as: $([v_1], v_2)$, $([v_1, v_2], v_3)$, $([v_1, v_2, ..., v_{n-1}], v_n)$. After preprocessing, the statitics of datasets are shown in Table 1.

4.2 Baselines and Backbones

To evaluate performance of proposed DIRAC, we compare it with following baselines, which focus on popularity bias in SBR:

- **TailNet[11]:** designs a preference mechanism to calculate a scale factor, which is used to adjust recommending probability of tail and head items respectively.
- **NISER[12]:** imposes L2 regularization on item embedding and session embedding respectively.
- **CauSeR[13]:** adopts do-calculus[28] to eliminate bad causal effects of momentum in SGD optimizer, which is biased towards the head items.

To prove generality of our framework, we integrate following SBR models into above popularity debias methods and proposed DIRAC:

- **GRU4Rec[3]:** uses GRU to model user's preference in current session.
- **NARM[5]:** adopts attention mechanism to model local preference, adopts GRU to model global preference, and combines both of local and global preference.
- **SR-GNN[6]:** utilizes GNN to capture transitions of items, and gets representation of the session by combining global and current preference.

4.3 Evaluation Metrics and Experimental Setup

4.3.1 Evaluation Metrics

In the experiments, we split the itemset into head itemset and tail itemset according to Pareto Principle[33] and adopt two common metrics as follows:

- **Recall@20:** (Recall) is the ratio of ground-truth items in top-20 recommending item list.
- **MRR@20:** (Mean Reciprocal Rank) is average inverse ranking of the first ground-truth item in top-20 recommending item list. It can evaluate ranking skill of SBR models.

4.3.2 Experimental Setup

In training process, we initialize all parameters with normal distribution ($mean = 0, deviation = 0.1$), and adopt Adam optimizer, except for CauSeR. SGD optimizer is used for CauSeR, because momentum is calculated by algorithm. Following previous works[6, 12], the dimensionality of latent vectors is set as $d = 100$ for all baselines on both dataset. However, we set $d = 50$ for both of two item encoders in proposed DIRAC for fair comparison. Also, we employ grid search to find the best hyper-parameters, i.e. α in {0.1, 1, 10}, β in {0.1, 1, 10} and μ in {14, 16, 18, 20}. We fix batch size to 100 and learning rate to 0.001.

4.4 Results and Analysis

We conduct experiments on all baselines and proposed framework to answer following research questions(RQ):

- **RQ1:** How does proposed DIRAC perform compared with other state-of-art debias methods, especially on tail itemset, which indicates the performance of eliminating popularity bias?
- **RQ2:** Will DIRAC degrade recommending performance on popular items?
- **RQ3:** How is the generality of DIRAC to eliminate popularity bias?
- **RQ4:** Does DIRAC perform well on sessions with different lengths?
- **RQ5:** How does each component for disentanglement affect performance of DIRAC? Does DIRAC disentangle interest and conformity?

4.4.1 Comparison Against Baselines (RQ1 & RQ2 & RQ3)

Performances of all baselines and proposed DIRAC on Diginetica and Retail-Rocket are shown in Table 2 and Table 3 respectively.

Overall Performance and Popularity Debias (RQ1). The results illustrate that proposed DIRAC consistently outperforms other baselines in terms of both Recall@20 and MRR@20 on two datasets with three SBR models, which proves the effectiveness of DIRAC. We firstly analyze performance of each SBR model. SR-GNN undoubtedly outperforms NARM and GRU4Rec, but performance of GRU4Rec on head itemset is pretty good. It can be concluded that performance on popular itemset contributes more to overall in GRU4Rec, indicating that it is affected by popularity bias more severely. As for baselines mentioned above, TailNet performs slightly worse than each SBR model, which is consistent with [11], because it focus on improving novelty but not accuracy. In terms of CauSeR, it achieves good performance with

Table 2 Performances of all methods on Diginetica dataset. The boldface and italic are the best and the second best results over all methods respectively.

Models	Recall@20(%)			MRR@20(%)		
	Overall	Head	Tail	Overall	Head	Tail
SR-GNN	50.5833	61.9673	38.2825	17.6272	22.9641	11.8604
SR-GNN–TailNet	50.3319	62.1729	37.5372	16.7829	22.0976	11.0401
SR-GNN–NISER	51.6350	*63.2233*	39.1132	*17.7878*	*23.0869*	12.0620
SR-GNN–CauSeR	*51.6366*	62.7867	*39.5884*	17.6168	22.6445	*12.1843*
SR-GNN–DIRAC	**52.2314**	**63.7580**	**39.7764**	**18.6495**	**23.7220**	**12.7940**
NARM	48.9254	61.4073	35.4381	16.3257	21.8273	10.3809
NARM–TailNet	48.6066	60.7017	35.5372	16.0545	21.4867	10.1843
NARM–NISER	*49.8303*	*62.0401*	36.6381	*16.5527*	*21.9670*	10.7023
NARM–CauSeR	42.0306	46.3568	*37.3560*	13.9535	16.1715	**11.5567**
NARM–DIRAC	**51.0253**	**63.4543**	**37.7149**	**17.5124**	**23.1057**	*11.4686*
GRU4Rec	45.8066	**64.9467**	25.1239	14.3497	*23.0813*	4.9149
GRU4Rec–TailNet	39.2389	56.7090	20.3617	11.5815	17.6829	4.9885
GRU4Rec–NISER	*50.7181*	62.1508	38.3645	*17.2835*	22.1048	12.0738
GRU4Rec–CauSeR	43.9285	46.9991	**40.6106**	14.8505	16.4148	**13.1603**
GRU4Rec–DIRAC	**51.9373**	*63.6948*	*39.2328*	**18.2425**	**23.3162**	*12.7602*

Table 3 Performances of all methods on RetailRocket dataset. The boldface and italic are the best and the second best results over all methods respectively.

Models	Recall@20(%)			MRR@20(%)		
	Overall	Head	Tail	Overall	Head	Tail
SR-GNN	59.6881	67.7302	46.9392	33.3581	**38.2037**	25.6764
SR-GNN–TailNet	58.7316	67.0650	45.5207	32.1390	36.9939	24.4425
SR-GNN–NISER	*61.2583*	*67.7597*	50.6346	*35.7171*	37.0178	*31.9114*
SR-GNN–CauSeR	60.8179	67.1239	*50.8212*	35.1494	37.1494	31.3946
SR-GNN–DIRAC	**61.6265**	**67.8243**	**51.8010**	**36.1386**	*38.0844*	**33.0540**
NARM	53.0284	62.9562	37.2900	29.2180	35.4191	19.3876
NARM–TailNet	45.1250	55.5082	31.9576	22.7829	28.5194	15.5083
NARM–NISER	*60.0635*	*66.9826*	*49.0948*	*34.0895*	*37.4140*	**28.8198**
NARM–CauSeR	54.2593	57.9232	48.4509	29.9830	32.0168	26.7589
NARM–DIRAC	**61.0562**	**68.3482**	**49.4961**	**34.1707**	**38.0345**	*28.0456*
GRU4Rec	49.5308	65.5816	24.0855	25.6102	35.7326	9.5633
GRU4Rec–TailNet	42.5823	61.9378	11.8981	23.5178	33.8762	7.0967
GRU4Rec–NISER	*60.3379*	*67.1121*	49.5987	*35.0529*	*37.5570*	31.0832
GRU4Rec–CauSeR	55.8836	57.6583	**53.0702**	32.3431	32.2294	**32.5234**
GRU4Rec–DIRAC	**61.1067**	**67.7890**	*50.5133*	**35.2231**	**37.7194**	*31.2657*

SR-GNN model, but has a poor overall performance in other situation. By comparison, only NISER and DIRAC are able to promote overall performance on both of datasets among three SBR models. Then, we analyze how baselines and DIRAC perform on eliminating popularity bias. The results on two datasets show that NISER, CauSeR and DIRAC all can promote the performance of tail itemset, which indicates that these three methods all relieve popularity bias efficiently. Besides, CauSeR and DIRAC achieve better performance than NISER under most of conditions. Though CauSeR performs better in eliminating popularity bias on GRU4Rec, it decreases performance on head itemset at the same time, which causes a poor overall performance. On

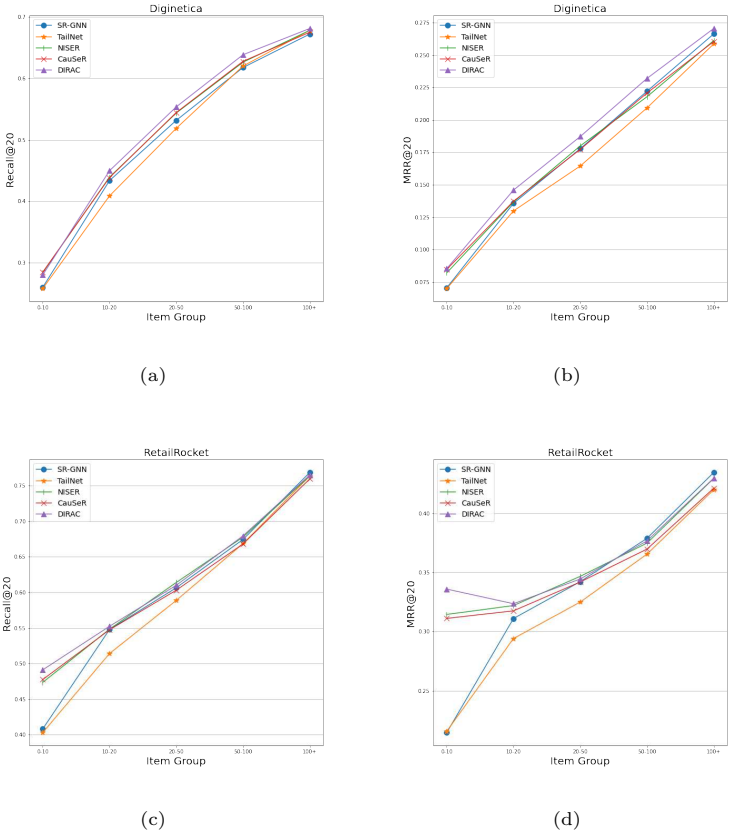


Fig. 5 Performances of all methods with SR-GNN model in each item group on Diginetica and RetailRocket dataset.

the contrary, DIRAC not only performs well on tail itemset, but also can elevate performance on head itemset. Therefore, we can conclude that proposed DIRAC eliminate popularity bias efficiently and achieve better overall performance compared with other baselines. Also, we can observe that GRU4Rec gets the most increasement on overall performance compared with NARM and SR-GNN, because it is affected by popularity bias most seriously.

Performance on Popular Items (RQ2) Results on two datasets in Table 2 and Table 3 show that our proposed DIRAC outperforms other baselines on popular items. For more detailed analysis, we also show recommending performance with SR-GNN model on different item groups, which are categorized by popularity of items. As Figure 5 shows, recommending performance of three baselines, i.e. TailNet, NISER and CauSeR, on 50-100 and 100+ item group all decline with different degree compared with SR-GNN model. On the contrary, DIRAC promotes performance of these two item groups on Diginetica. Though DIRAC performs a little worse than SR-GNN on RetailRocket

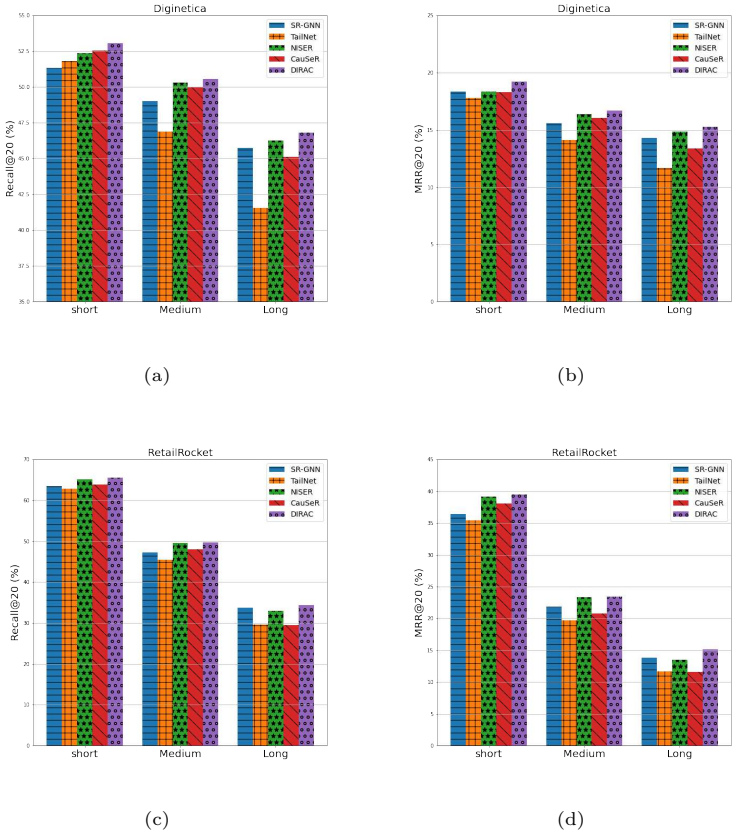


Fig. 6 Performances of all methods with SR-GNN model for various length sessions.

dataset under MRR@20 metric, it still exceeds other baselines and is close to SR-GNN. Therefore, we can conclude that our proposed DIRAC overcome the problem of performance loss on popular items while eliminating popularity bias.

Generality (RQ3) In this paragraph, we will answer the question of generality. As shown in Table 1 and Table 2, proposed DIRAC achieves better overall performance and promote tail itemset performance on both of two datasets with all of three SBR models, which proves its good generality. NISER also has generality as DIRAC, but it performs worse on both of overall itemset and tail itemset under most of conditions. As for CauSeR, its overall performance varies largely with different SBR models, however, it can eliminate popularity bias consistently.

Table 4 Ablation experiment of DIRAC with SR-GNN model on Diginetica dataset

Method	Recall@20(%)			MRR@20(%)		
	Overall	Head	Tail	Overall	Head	Tail
DIRAC	**52.2314**	**63.7580**	*39.7764*	**18.4695**	**23.7220**	**12.7940**
w/o Pop-Encoder	*52.1328*	*63.7169*	39.6157	*18.4097*	*23.6184*	12.7814
w/o Auxiliary	51.9751	63.1474	**39.9029**	17.8994	23.0931	12.2875
w/o Discrepancy	52.0950	63.6915	39.5645	18.4045	23.5854	*12.6063*

Table 5 Ablation experiment of DIRAC with SR-GNN model on RetailRocket dataset

Method	Recall@20(%)			MRR@20(%)		
	Overall	Head	Tail	Overall	Head	Tail
DIRAC	**61.6265**	*67.8243*	**51.8010**	**36.1386**	*38.0844*	**33.0540**
w/o Pop-Encoder	*61.4604*	**67.8714**	51.2971	*35.8507*	**38.2330**	32.0739
w/o Auxiliary	61.3088	67.5595	51.3998	35.8143	37.8095	32.6513
w/o Discrepancy	61.1825	67.2946	*51.4931*	35.8085	37.7315	*32.7601*

4.4.2 Analysis on sessions with different lengths (RQ4)

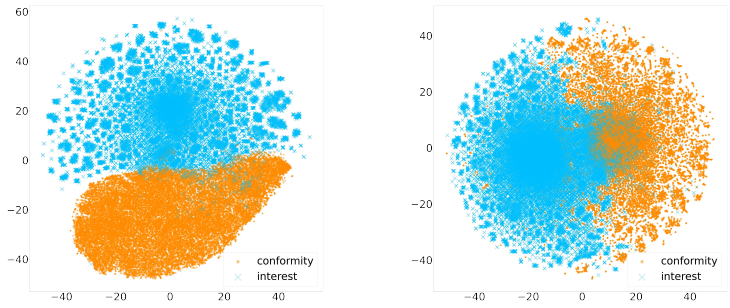
To further investigate how session length affects the performance, we compare DIRAC with other baselines among different session groups. The entire sessions are split into three groups: short sessions (≤ 5 items), medium sessions (> 5 and ≤ 10 items) and long sessions (> 10 items). As Figure 6 shows, proposed DIRAC framework outperforms other baselines with all of three length of session groups on both of Recall@20 and MRR@20 metrics. Even improvements on long session group are larger than the other two groups.

4.4.3 Study on DIRAC(RQ5)

We analyze the effects of each components for disentangling interest and conformity, i.e., popularity encoder, auxiliary task and discrepancy loss. As shown in Table 4 and Table 5, the results show that each of these three parts is beneficial for overall performance, which illustrates that they can help disentangle interest and conformity preference well. Also, we visualize the learned item embedding in proposed DIRAC using t-SNE[37]. From the Figure 7, we observe that two sets of embedding, which represents interest and conformity respectively, are separated well. It can prove the assumption that individual's preference consists of various degree of interest and conformity, and they should be disentangled for more precise estimation of preference.

5 Conclusion and Future Work

In this paper, we find that recommending performance on popular items degrades while eliminating popularity bias, due to ignoring of individual's conformity. Therefore, we propose a general SBR framework with **D**isentangling **I**nte**R**est **A**nd **C**onformity (DIRAC) to tackle the problem. In our proposed DIRAC, item encoding module is used to get two sets of item embedding which



(a) (b)

Fig. 7 Visualization of the learned item embedding in DIRAC with SR-GNN model on (a) Diginetica and (b) RetailRocket dataset.

represents interest and conformity respectively. Then, an interest modeling module and a conformity modeling module are devised to model and disentangle the preference of interest and conformity. At last, a fusion module combine these two types of preference for recommending. Extensive experiments verify that proposed DIRAC can eliminate popularity bias and promote overall performance better for several session-based recommendation models compared with other state-of-arts, and avoid performance loss on popular items.

As we know, popularity of one item is changing in its life cycle, which indicates that popularity bias is dynamic especially in session-based recommendation or sequential recommendation. We will explore solutions to this problem in the future.

References

[1] Jannach, D., Ludewig, M., Lerche, L. (2017) Session-based item recommendation in e-commerce: on short-term intents, reminders, trends and discounts. User Model User-Adap Inter, 27(3), 351-392 https://doi.org/10.1007/s11257-017-9194-1

[2] Ludewig, M., Jannach, D. (2018) Evaluation of session-based recommendation algorithms. User Model User-Adap Inter, 28(4), 331-390 https://doi.org/10.1007/s11257-018-9209-6

[3] Bal´azs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, Domonkos Tikk. (2016) Session-based Recommendations with Recurrent Neural Networks. In: Proceedings of the 4th International Conference on Learning Representations(ICLR) https://arxiv.org/abs/1511.06939

[4] Liu, Q., Zeng, Y., Mokhosi, R., and Zhang, H. (2018) STAMP: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD), pp. 1831-1839 https://doi.org/10.1145/3219819.3219950

[5] Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., and Ma, J. (2017) Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management(CIKM), pp. 1419-1428 https://doi.org/10.1145/3132847.3132926

[6] Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., and Tan, T. (2019) Session-based recommendation with graph neural networks. In: Proceedings of the AAAI conference on artificial intelligence(AAAI), Vol. 33, No. 01, pp. 346-353 https://doi.org/10.1609/aaai.v33i01.3301346

[7] Nguyen, T. T., Hui, P. M., Harper, F. M., Terveen, L., Konstan, J. A. (2014) Exploring the filter bubble: the effect of using recommender systems on content diversity. In: Proceedings of the 23rd international conference on World wide web(WWW), pp. 677-686. https://doi.org/10.1145/2566486.2568012

[8] Zeng, J. et al. (2020). User Sequential Behavior Classification for Click-Through Rate Prediction. In: Nah, Y., Kim, C., Kim, SY., Moon, YS., Whang, S.E. (eds) Database Systems for Advanced Applications. DASFAA 2020 International Workshops. DASFAA 2020. Lecture Notes in Computer Science(), vol 12115. Springer, Cham. https://doi.org/10.1007/978-3-030-59413-8_22

[9] Joachims, T., Swaminathan, A., and Schnabel, T. (2017) Unbiased learning-to-rank with biased feedback. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining(WSDM), pp. 781-789 https://doi.org/10.1145/3018661.3018699

[10] Zheng, Y., Gao, C., Li, X., He, X., Li, Y., and Jin, D. (2021) Disentangling user interest and conformity for recommendation with causal embedding. In: Proceedings of the 30th international conference on World wide web(WWW), pp. 2980-2991 https://doi.org/10.1145/3442381.3449788

[11] Liu, S., and Zheng, Y. (2020) Long-tail session-based recommendation. In: Proceedings of the 14th ACM conference on recommender systems(RecSys), pp. 509-514 https://doi.org/10.1145/3383313.3412222

[12] Gupta, P., Garg, D., Malhotra, P., Vig, L., and Shroff, G. (2019) NISER: Normalized item and session representations to handle popularity bias. arXiv preprint arXiv:1909.04276

[13] Gupta, P., Sharma, A., Malhotra, P., Vig, L., and Shroff, G. (2021) CauSeR: Causal Session-based Recommendations for Handling Popularity Bias. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management(CIKM), pp. 3048-3052 https://doi.org/10.1145/3459637.3482071

[14] Wang, S., Cao, L., Wang, Y., Sheng, Q. Z., Orgun, M. A., Lian, D. (2021) A survey on session-based recommender systems. ACM Computing Surveys(CSUR), 54(7), 1-38. https://doi.org/10.1145/3465401

[15] Krishnan, S., Patel, J., Franklin, M. J., Goldberg, K. (2014) A methodology for learning, analyzing, and mitigating social influence bias in recommender systems. In: Proceedings of the 8th ACM Conference on Recommender Systems(RecSys), pp. 137-144 https://doi.org/10.1145/2645710.2645740

[16] Liu, Y., Cao, X., Yu, Y. (2016) Are you influenced by others when rating? improve rating prediction by conformity modeling. In: Proceedings of the 10th ACM conference on Recommender Systems(RecSys), pp. 269-272 https://doi.org/10.1145/2959100.2959141

[17] Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., Yang, K. (2019) Deep session interest network for click-through rate prediction. arXiv preprint arXiv:1905.06482.

[18] Garg, D., Gupta, P., Malhotra, P., Vig, L., Shroff, G (2019) Sequence and time aware neighborhood for session-based recommendations: Stan. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR), pp. 1069-1072 https://doi.org/10.1145/3331184.3331322

[19] Jannach, D., Ludewig, M (2017) When recurrent neural networks meet the neighborhood for session-based recommendation. In: Proceedings of the 11th ACM Conference on Recommender Systems(RecSys), pp. 306-310 https://doi.org/10.1145/3109859.3109872

[20] Rendle, S., Freudenthaler, C., Schmidt-Thieme, L (2010) Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web(WWW), pp. 811-820 https://doi.org/10.1145/1772690.1772773

[21] Wu, X., Liu, Q., Chen, E., He, L., Lv, J., Cao, C., Hu, G (2013) Personalized next-song recommendation in online karaokes. In: Proceedings of the 7th ACM Conference on Recommender Systems(RecSys), pp. 137-140 https://doi.org/10.1145/2507157.2507215

[22] Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., Zhang, X. (2021) Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence(AAAI), Vol. 35, No. 5, pp. 4503-4511 https://ojs.aaai.org/index.php/AAAI/article/view/16578

[23] Pang, Y., Wu, L., Shen, Q., Zhang, Y., Wei, Z., Xu, F. et.al. (2022) Heterogeneous global graph neural networks for personalized session-based recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining(WSDM), pp. 775-783 https://doi.org/10.1145/3488560.3498505

[24] Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X. (2020) Bias and debias in recommender system: A survey and future directions. arXiv preprint arXiv:2010.03240

[25] Agarwal, A., Takatsu, K., Zaitsev, I., Joachims, T. (2019) A general framework for counterfactual learning-to-rank. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR), pp. 5-14 https://doi.org/10.1145/3331184.3331202

[26] Bottou, L., Peters, J., Quiñonero-Candela, J., Charles, D. X., Chickering, D. M., Portugaly, E. et.al. (2013) Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising. Journal of Machine Learning Research, 14(11) http://jmlr.org/papers/v14/bottou13a.html

[27] Gruson, A., Chandar, P., Charbuillet, C., McInerney, J., Hansen, S., Tardieu, D., Carterette, B. (2019) Offline evaluation to make decisions about playlistrecommendation algorithms. In: Proceedings of the 12nd ACM International Conference on Web Search and Data Mining(WSDM), pp. 420-428 https://doi.org/10.1145/3289600.3291027

[28] Pearl, J. (2009) Causality. Cambridge university press

[29] Zhang, Y., Feng, F., He, X., Wei, T., Song, C., Ling, G., Zhang, Y. (2021) Causal intervention for leveraging popularity bias in recommendation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR), pp. 11-20 https://doi.org/10.1145/3404835.3462875

[30] Wang, W., Feng, F., He, X., Wang, X., Chua, T. S. (2021) Deconfounded recommendation for alleviating bias amplification. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining(KDD), pp. 1717-1725 https://doi.org/10.1145/3447548.3467249

[31] Wei, T., Feng, F., Chen, J., Wu, Z., Yi, J., He, X. (2021) Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining(KDD), pp. 1791-1800 https://doi.org/10.1145/3447548.3467289

[32] Xu, C., Zhao, P., Liu, Y., Sheng, V. S., Xu, J., Zhuang, F. et.al (2019) Graph Contextualized Self-Attention Network for Session-based Recommendation. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence(IJCAI), pp. 3940-3946 https://doi.org/10.5555/3367471.3367589

[33] Armstrong, R. (2008) The long tail: Why the future of business is selling less of more. Canadian Journal of Communication, 33(1), 127

[34] Li, Y., Zemel, R., Brockschmidt, M., Tarlow, D. (2016) Gated Graph Sequence Neural Networks. In: Proceedings of the 4th International Conference on Learning Representations(ICLR) https://arxiv.org/abs/1511.05493

[35] Qi, T., Wu, F., Wu, C., Huang, Y (2021) PP-Rec: News Recommendation with Personalized User Interest and Time-aware News Popularity. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing(ACL), Volume 1: Long Papers, pp. 5457-5467 https://doi.org/10.18653/v1/2021.acl-long.424

[36] Wang, X., Jin, H., Zhang, A., He, X., Xu, T., Chua, T. S. (2020) Disentangled graph collaborative filtering. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval(SIGIR), pp. 1001-1010 https://doi.org/10.1145/3397271.3401137

[37] Van der Maaten, L., Hinton, G. (2008) Visualizing data using t-SNE. Journal of machine learning research, 9(11). http://jmlr.org/papers/v9/vandermaaten08a.html

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- KAISDIRAC.zip