Self-supervised Learning for Alleviating Selection Bias in Recommendation Systems

Haochen Liu*
Michigan State University
East Lansing, MI, USA
liuhaoc1@msu.edu

Da Tang
ByteDance Inc.
Mountain View, CA, USA
da.tang@bytedance.com

Ji Yang
ByteDance Inc.
Mountain View, CA, USA
ji.yang@bytedance.com

Xiangyu Zhao
Michigan State University
East Lansing, MI, USA
zhaoxi35@msu.edu

Jiliang Tang
Michigan State University
East Lansing, MI, USA
tangjili@msu.edu

Youlong Cheng
ByteDance Inc.
Mountain View, CA, USA
youlong.cheng@bytedance.com

ABSTRACT

Recommendation systems are trained on historical rating data explicitly provided by users. Since users have the freedom to select what items to rate by themselves, the collected recommendation datasets typically suffer from selection bias. As a result, recommendation models trained on such biased data perform not well on unobserved data. Traditional solutions to selection bias such as data imputation and inverse propensity score are sensitive to the quality of the additionally introduced imputation model or the propensity estimation model. In this work, we propose a novel self-supervised learning (SSL) framework **Rating Distribution Calibration (RDC)** to alleviate the negative impacts of selection bias on recommendation models. In addition to the original training objective, we introduce a rating distribution calibration loss which aims to correct the predicted rating distribution of a biased user by forcing them to be close to that of similar unbiased users. Extensive experiments are conducted on a real-world recommendation dataset and results show that our proposed framework outperforms the original model as well as state-of-the-art debiasing approaches by a significant margin. A detailed parameter sensitivity analysis is also included to help us understand the influence of the choices of the hyperparameters on the debiasing performance.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

recommendation system, unbiased recommendation, self-supervised learning

*This work was done when Haochen Liu worked as an intern at Bytedance, USA.

IRS 2021, August 15–16, 2021, Virtual

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Haochen Liu, Da Tang, Ji Yang, Xiangyu Zhao, Jiliang Tang, and Youlong Cheng. 2021. Self-supervised Learning for Alleviating Selection Bias in Recommendation Systems. In *IRS '21: International Workshop on Industrial Recommendation Systems at SIGKDD'21, August 15–16, 2021, Virtual.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

By learning users' preferences, recommendation systems accurately find the items or services that they are interested in, which effectively combats the problem of information overloading. Many recommendation systems learn users' preferences for items based on their historical explicit feedback, such as binary or multi-scale ratings. However, recent studies have shown that user rating data suffer from selection bias [2]. Specifically, users can select which items to rate by themselves. They typically don't select and rate items by random, but instead, their selection behaviors follow certain patterns [11]. For example, some users are more likely to rate items that they strongly like or dislike. As a result, the ratings collected under selection bias are not representative enough to reflect the preferences of users to all items, and thereby mislead recommendation models to provide biased predictions. Another equivalent formulation of the selection bias is called the missing-not-at-random (MNAR) problem. Given that users select which items to rate by themselves, the unobserved ratings are not missing randomly. But a desirable recommendation model seeks to make accurate predictions on all user-item pairs, which can only be precisely evaluated on missing-at-random (MAR) test data [16]. However, directly training a recommendation model on MNAR training data leads to defective performances on MAR test data.

Figure 1 shows a simple and intuitive example of selection bias. Table (a) shows the underlying preferences of six users on six items. Based on certain selection patterns, the six users select partial items to rate, which results in the observed user ratings in Table (b), where the unobserved ratings are denoted as interrogation marks. There are three types of selection behaviors. First, user A and user D randomly choose items to rate and their choice is independent of the underlying rating of an item. Thus, the observed ratings are MAR. Second, users B and F tend to choose the items they like to rate. Third, users C and E tend to choose the items they dislike to rate. A recommendation model trained on such observed ratings tends to produce accurate predictions for users without selection bias (type

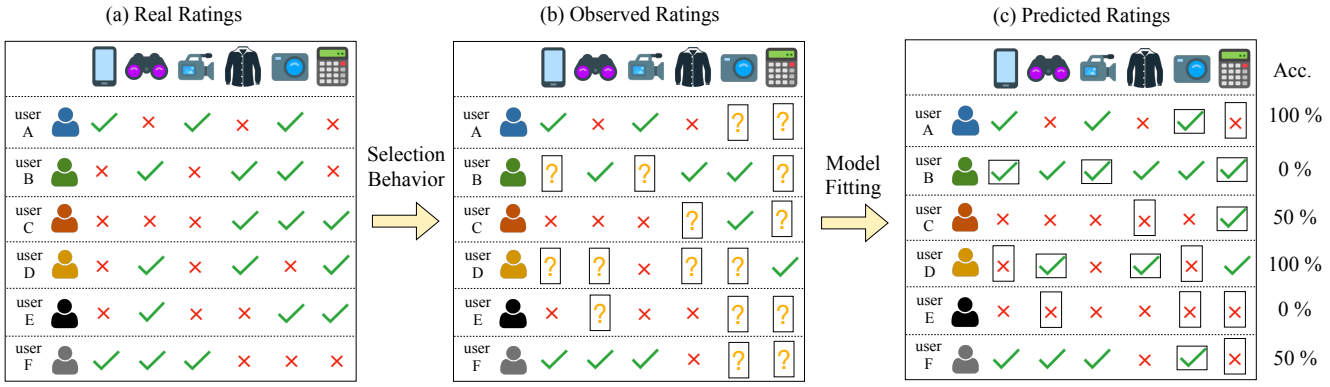

Figure 1: An example of selection bias in recommendation systems.

1), and inaccurate predictions for users with selection bias (type 2 and 3), as shown in Table (c). Since the observed ratings of users A and D are MAR, the data are representative of their preferences. Then the model can get a comprehensive understanding of their preferences, so as to provide more accurate predictions. User F and C are more likely to rate the items they like/dislike, respectively. As a result, the model tends to predict more positive/negative ratings for them, which leads to errors in predictions. More extremely, users B and E only provide ratings to items they like/dislike, and then the model learns significantly biased preferences and tends to predict all the items as positive/negative, which leads to very bad performances.

Traditional methods dealing with selection bias in recommendation systems are mainly based on two ideas. First, since the collected training rating data are MNAR, some works propose to include a data imputation module to estimate the missing data, where heuristic methods [17] and learning-based methods [18] are applied. Second, researchers propose to leverage inverse propensity score (IPS) to reweight training data during training, based on the estimated probability of observing them. Furthermore, Wang et al. [18] propose to combine the above two strategies to mitigate selection bias. Although existing methods alleviated the negative impacts of selection bias to some extent, the debiasing performances of the aforementioned two approaches heavily rely on the quality of the data imputation model and the propensity score estimation model, which is hard to guarantee in practice [2].

In this paper, we seek to solve the problem of selection bias using a novel strategy. Recommendation models are typically trained via supervised learning. The observed rating data with selection bias provide biased supervision signals, which leads to biased models. Hence, we need additional debiased signals to correct the model. We propose an SSL framework to provide such supervision signals by introducing self-defined learning tasks. Our proposal is primarily based on the following observations: (1) The degree of selection bias of different users varies. For example, some users show strong selection bias when interacting with recommendation systems (e.g. the user B in Figure 1) while some users have no selection bias and their rating data are almost missing at random (e.g. the user A in Figure 1). (2) The negative impact of selection bias on the recommendation model is directly reflected in its predicted rating distribution of a user. For example, due to the selection bias of user B, the recommendation model predicts a rating distribution (all items are positive) that is largely deviated from the real distribution (half positive and half negative). Thus, an intuitive idea for dealing with selection bias is to introduce an SSL task to correct the predicted rating distributions of users with selection bias based on the distributions of similar users without selection bias. Specifically, we iteratively find the users with less selection bias, which we call "pivot" users, and force the predicted rating distribution of non-pivot users to be close to that of their similar pivot users. Without introducing new models and new data, the SSL task directly regularizes the biased rating predictions by the contrast between different users and has been empirically validated to achieve better debiasing performance compared with competitive baseline approaches. We summarize our contributions in this work as follows.

- To the best of our knowledge, we first investigate alleviating selection bias in recommendation systems via self-supervised learning.
- We propose a novel SSL-based framework **Rating Distribution Calibration (RDC)** to reduce the influence of selection bias in training recommendation models.
- We empirically evaluate our methods on one public dataset. The experimental results demonstrate the superiority of our proposed model over state-of-the-art baselines. We also include extensive experiments to analyze the sensitivity of the hyper-parameters in our framework.

The remainder of this paper is organized as follows. First, in Section 2, we review the related works. Then, we will present some preliminary knowledge in Section 3. Next, we describe the proposed framework in detail in Section 4. Then, we carry out our experimental setups and results with discussions in Section 5. Finally, Section 6 concludes the work with possible future research directions.

2 RELATED WORKS

In this section, we go through the related works. We summarize the latest studies in two research lines which are tied to our work,

namely, selection bias in recommendation systems and self-supervised learning.

2.1 Selection Bias in Recommendation Systems

As an important application in artificial intelligence (AI), recommendation systems are required to be unbiased [1, 8]. Selection bias is one kind of important bias that needs to be eliminated [2]. Previous studies on selection bias in recommendation tackle this problem on two families of methods. The first category is known as data imputation. Given that selection bias results in a biased data missing mechanism, Steck [17] propose a heuristic strategy to impute missing rating data to alleviate the impacts of such bias. Specifically, they use a constant value as the imputed ratings and optimize the recommendation model with a weighted objective function where the contributions of missing values are adaptively down-weighted. Based on the assumption that the probability of observing a rating depends on its underlying value, Hernández-Lobato et al. [6] propose to jointly learn an MF-based complete data model as well as an MF-based missing data model to explain the generation of complete rating data and the data missing mechanism. With the guidance of the data missing model, the complete data model learns unbiased and more accurate predictions. The second category is called inverse propensity scoring. Since the ratings that are less likely to be observed are underrepresented in biased data, we can up-weight these ratings during training as a means of compensation. Schnabel et al. [16] propose two methods to estimate the propensity score of a rating (i.e., the probability of observing this rating), and utilize the inverse propensity score (IPS) to weigh the corresponding loss term of that rating.

Based on the above two categories of debiasing methods, Wang et al. [18] propose a novel framework to integrate them in a doubly robust manner to overcome the high-bias issue of the data imputation methods and the high-variance issue of the IPS-based methods. Moreover, based on this doubly robust framework, Wang et al. [19] further propose a framework called Learning to Debias (LTD). In the LTD, a propensity estimation model is introduced to learn more accurate propensity scores. The propensity estimation model is optimized according to the performance of the resulting rating model on a small amount of unbiased data. However, the aforementioned two types of debiasing methods, as well as their combination, heavily rely on the quality of the data imputation model or the propensity estimation model, which is hard to guarantee in practice [2]. In contrast, our method avoids the introduction of additional models but only develops a novel SSL task based on the original model and data, which makes our method more robust than the existing methods.

2.2 Self-supervised Learning

As an emerging technology for representation learning via unlabeled data, self-supervised learning (SSL) has drawn increasing attention from various communities. The principle idea of SSL is to improve the generalization ability of the model by training it on auxiliary tasks where no human-annotated labels are needed. In the computer vision (CV) field, various SSL techniques are developed for learning high-quality image representations. Doersch et al. [3] first propose to predict the relative locations of image patches. Following this line of research, Noroozi and Favaro [12] design a pretext task called Jigsaw Puzzle. More types of pretext tasks have also been investigated, such as image rotation [4], image inpainting [14], image colorization and motion segmentation prediction Pathak et al. [13]. SSL is also used in the graph domain to enhance representation learning with full exploitation of unlabeled data [7]. In the domain of recommendations, there are few works incorporating SSL. Zhou et al. [20] employ the mutual information maximization principle to define four auxiliary tasks to improve sequential recommendation. Liu et al. [9] propose a graph contrastive learning to improve the generalization ability of graph neural network (GNN) based recommendation models.

3 PRELIMINARIES

In this section, we introduce some preliminary knowledge about the problem under study. We first state the problem, and then describe two common recommendation models that will be studied in our research.

3.1 Problem Statement

In general, recommendation models seek to predict the "rating" or "preference" a user would give to an item [15]. Specifically, recommendation models have various forms, including rating models, which predict a binary or numeric rating of a user to an item; and ranking models, which sort a set of items according to a user's preference. Both these two kinds of recommendation models suffer from selection bias in the training data. In this work, we focus on the former type, and note that our method and conclusion can also be extended to the latter type.

Let $\mathcal{U} = \{u_1, ..., u_N\}$ be a set of users, $\mathcal{I} = \{i_1, ..., i_M\}$ be a set of items. A rating model can be viewed as a function $\mathbf{M} : (u, i) \rightarrow \hat{r}_{u,i}$ that maps a pair of user and item to a numeric rating. Let $\mathcal{R} = \{(u, i, r_{u,i}) : (u, i) \in \mathcal{U} \times \mathcal{I}\}$ be the underlying full rating set, where $r_{u,i}$ is the rating of user u to item i. Given a set of observed rating data $\mathcal{R}^{\text{o}} \subset \mathcal{R}$, we seek to find a rating model $\mathbf{M}$ that can make an accurate prediction $\hat{\mathcal{R}}$ on the full rating set $\mathcal{R}$, which is evaluated by the prediction error:

$$\mathcal{E} = \mathcal{E}(\hat{\mathcal{R}}, \mathcal{R}) = \frac{1}{|\mathcal{R}|} \sum_{(u,i,r_{u,i}) \in \mathcal{R}} (\hat{r}_{u,i} - r_{u,i})^2$$

In practice, the full ratings $\mathcal{R}$ is not available since we cannot collect the true ratings of each user to each item. Instead, we can evaluate the prediction error of a model on a collected MAR test set $\mathcal{R}^{\text{test}} \subset \mathcal{R}$ as an unbiased estimation of that on the full rating matrix:

$$\mathcal{E} \approx \mathcal{E}^{\text{test}} = \mathcal{E}(\hat{\mathcal{R}}^{\text{test}}, \mathcal{R}^{\text{test}}) = \frac{1}{|\mathcal{R}^{\text{test}}|} \sum_{(u,i,r_{u,i}) \in \mathcal{R}^{\text{test}}} (\hat{r}_{u,i} - r_{u,i})^2$$

It can be shown that $\mathcal{E}^{\text{test}}$ is an unbiased estimation of $\mathcal{E}$. The detailed proof can be found in Appendix A.

Formally, given a set of observed rating data $\mathcal{R}^{\text{o}}$, the problem is to find an optimal rating model $\mathbf{M}$ that can minimize the prediction error $\mathcal{E}^{\text{test}}$ on the MAR test set $\mathcal{R}^{\text{test}}$. However, due to selection bias, the observed rating data are MNAR, which leads rating models trained on them to perform badly on the MAR test data. Following the work [19], we assume that we have a small amount of MAR ratings which can provide guidance for us to fix the bias of the

model trained with a large MNAR dataset. Thus, the problem can be formulated as – given a set of observed ratings $\mathcal{R}^{\text{o}} = \mathcal{R}^{\text{o}}_{\text{MNAR}} \cup \mathcal{R}^{\text{o}}_{\text{MAR}}$, where $\left|\mathcal{R}^{\text{o}}_{\text{MNAR}}\right| \gg \left|\mathcal{R}^{\text{o}}_{\text{MAR}}\right|$, we seek to find a rating model $\mathbf{M}$ that minimizes the prediction error $\mathcal{E}^{\text{test}}$ on $\mathcal{R}^{\text{test}}$.

3.2 Rating models

The last section describes the general problem of building an unbiased rating model regardless of its specific architecture. In practice, different types of rating models can be used. In this section, we introduce two common rating models: matrix factorization (MF) and neural factorization machine (NFM) [5]. We will apply these two models as examples to show how they are debiased under our proposed RDC framework. Note that our framework can be incorporated with any rating model. In the rest of the paper, we refer to the rating model as the "base" model, in order to distinguish it from the debiasing model.

The MF model is parameterized by a user feature matrix $P \in \mathbb{R}^{N\times K}$, an item feature matrix $Q \in \mathbb{R}^{M\times K}$, and bias vectors for users $b_u \in \mathbb{R}^N$ and items $b_i \in \mathbb{R}^M$ to predict the full ratings R:

$$\hat{r}_{u,i} = \sum_{k=1}^{K} P_{uk}Q_{ki} + b_u + b_i$$

The NFM model includes an embedding layer that maps a user-id u or an item-id i to their corresponding embeddings e_u and e_i, respectively. Then a bi-interaction layer calculates the element-wise dot product of e_u and e_i, which is then passed through several layers of feed-forward neural networks to predict the numeric rating $\hat{r}_{u,i}$.

4 THE PROPOSED FRAMEWORK

In this section, we detail our proposed SSL framework RDC. We first present the overview of the RDC framework, and then describe the SSL task designed for rating distribution calibration. Finally, we present the optimization method.

4.1 Overview

The original rating model is optimized via supervised learning by minimizing the mean square error (MSE) between the predicted ratings and the observed ratings:

$$\mathcal{L}_M = \frac{1}{|\mathcal{R}^{\text{o}}|} \sum_{(u,i,r_{u,i})\in\mathcal{R}^{\text{o}}} (\hat{r}_{u,i} - r_{u,i})^2 + \lambda \cdot Reg(\theta) \tag{1}$$

where $Reg(\theta)$ indicates the regularization term of the model parameters θ, and λ is a hyper-parameter for adjusting the weight of the regularization term.

As discussed in the introduction, the selection bias leads a rating model trained on the vanilla MSE loss to predict biased rating distributions for users. In RDC, we seek to alleviate the negative impacts of selection bias on the rating model by calibrating the predicted rating distribution of the users, and alleviate the impacts by an additional SSL task. Given that different users show different levels of selection bias, and users with less/more selection bias can achieve more/less accurate predicted rating distribution, respectively, we can use the rating distributions of the former as standards to calibrate that of the latter. Formally, we define the users who leave ratings in a MAR way (i.e., users with less selection bias) as "pivot" users, and accordingly, the users who provide ratings in an MNAR way (i.e., users with more selection bias) as "non-pivot" users. Hereby, we design an SSL framework which (1) extracts the rating distribution features from the users; (2) dynamically finds similar pivot users for non-pivot users by comparing their rating distributions; and (3) calibrates the rating distributions of non-pivot users with their similar pivot users as an SSL task.

4.2 The SSL task

Rating Distribution Feature Extraction. We denote the predicted ratings of a user u to all items $I = \{i_1, \ldots, i_M\}$ as $\hat{\mathbf{r}}_u = \{\hat{r}_{u_1}, \ldots, \hat{r}_{u_M}\}$. We define an aggregation function $g(\hat{\mathbf{r}}_u)$ that maps the predicted ratings $\hat{\mathbf{r}}_u$ to its distribution features. In order to ensure distribution features to be differentiable to the model parameters so that the SSL objective can be optimized via gradient back-propogation, we cannot use heuristic discrete distribution features like the frequency distribution $\mathbf{g}_u = g(\hat{\mathbf{r}}_u) = \left\{\Pr\left[\hat{r}_u = r\right]\right\}_{r=1}^{R}$ or the statistical features $\mathbf{g}_u = g(\hat{\mathbf{r}}_u) = \{\text{mean}(\hat{\mathbf{r}}_u), \text{var}(\hat{\mathbf{r}}_u)\}$. Instead, we propose a smooth distribution feature function as an approximation of the distribution density of the predicted ratings $\hat{\mathbf{r}}_u$:

$$\mathbf{g}_u = g(\hat{\mathbf{r}}_u) = \left[f^{(1)}(\hat{\mathbf{r}}_u), \ldots, f^{(K)}(\hat{\mathbf{r}}_u)\right],$$
$$\text{where } f^{(k)}(\hat{\mathbf{r}}_u) = \frac{1}{M}\sum_{i=1}^{M} \frac{1}{1+\exp^{-\tau(\hat{r}_{u,i}-k)}}, k \in \{1, \ldots, K\}$$

where K indicates the number of features and τ is the temperature value that controls the smoothness of the function.

Finding Pivot Users. As shown in the introduction, a user with less selection bias is able to obtain more accurate predictions on MAR data. Thus, we propose to choose pivot users based on their performances on the observed MAR rating set $\mathcal{R}^{\text{o}}_{\text{MAR}}$. In iteration n, the pivot user set $\mathbf{P}^{(\mathbf{n})}_{+}$ is defined as the users whose MAE on the MAR rating set $\mathcal{R}^{\text{o}}_{\text{MAR}}$ is less than a threshold t, which serves as a hyper-parameter:

$$\mathbf{P}^{(\mathbf{n})}_{+} = \{u \in \mathcal{U} : \text{MAE}\left(u; \mathcal{R}^{\text{o}}_{\text{MAR}}\right) < t\} \tag{2}$$

where $\text{MAE}(u; \mathcal{R}^{\text{o}}_{\text{MAR}})$ is the mean absolute error (MAE) of user u on the MAR rating set $\mathcal{R}^{\text{o}}_{\text{MAR}}$.

Thereby we can have the non-pivot user set $\mathbf{P}^{(\mathbf{n})}_{-} = I/\mathbf{P}^{(\mathbf{n})}_{+}$. Given a non-pivot user $u \in \mathbf{P}^{(\mathbf{n})}_{-}$, we compare the rating distribution similarity between u and each pivot user and try to find the closest one u^*:

$$u^* = \text{argmin}_{u'\in\mathbf{P}^{(\mathbf{n})}_{+}} ||\mathbf{g}_{u'} - \mathbf{g}_u||_2^2 \tag{3}$$

where $||\cdot||_2$ indicates the l^2-norm.

Rating Distribution Calibration. The SSL loss is defined as follows to forces the distribution features of the non-pivot users to be close to that of their similar pivot users:

$$\mathcal{L}_{SSL} = \frac{1}{|\mathbf{P}^{(\mathbf{n})}_{-}|} \sum_{u\in\mathbf{P}^{(\mathbf{n})}_{-}} ||\mathbf{g}_{u^*} - \mathbf{g}_u||_2^2 \tag{4}$$

Algorithm 1: RDC optimization

Input: Observed MNAR rating set $\mathcal{R}^{o}_{\text{MNAR}} = \{u_t, v_t, r_t\}_{t=1}^{N_1}$, MAR training rating set $\mathcal{R}^{o}_{\text{MAR}} = \{(u_t, v_t, r_t)\}_{t=1}^{N_2}$, validation rating set $V = \{u_t, v_t, r_t\}_{t=1}^{|V|}$, p, τ, K, λ, μ, t

Output: a rating model $\mathbf{M}$

1 Initialize the rating model $\mathbf{M}$ by pre-training it on the observed rating set $\mathcal{R}^{o} = \mathcal{R}^{o}_{\text{MNAR}} \cup \mathcal{R}^{o}_{\text{MAR}}$.
2 **repeat**
3 Evaluate the performances of all the M users on $\mathcal{R}^{o}_{\text{MAR}}$, to obtain the current pivot user set $\mathbf{P}_{+}^{(\mathbf{n})}$ and the non-pivot user set $\mathbf{P}_{-}^{(\mathbf{n})}$
4 For each non-pivot user $u \in \mathbf{P}_{-}^{(\mathbf{n})}$, find its closest pivot user $u^* \in \mathbf{P}_{+}^{(\mathbf{n})}$ via Eq 3
5 Calculate the SSL loss $\mathcal{L}_{SSL}$ in Eq 4
6 Calculate the MSE loss $\mathcal{L}_M$ in Eq 1
7 Update $\mathbf{M}$ by optimizing the overall loss $\mathcal{L}$ in Eq 5
8 Evaluate the rating model $\mathbf{M}$ on the validation rating set V
9 **until** the performance of $\mathbf{M}$ on V doesn't improve for consecutive p epochs;

4.3 Optimization

With the updating of the rating model, the set of the pivot and non-pivot users are also changing. We empirically find that with the improvement of the rating model's performance, more and more users can achieve better performance on the MAR rating set, and thus, some non-pivot change to pivot users. Hence, we alternatively update the rating model and divide the pivot user set during the optimization process, so as to leverage the increasing number of pivot users to provide more guidance for debiasing. The optimization algorithm is presented in Algorithm 1.

First, we initialize the rating model by pre-training it on the original MSE objective function (Eq 1), which has been empirically shown to accelerator the convergence of the whole framework (line 1). Next, we iteratively update the RDC framework for N epochs (from line 2 to line 7). In each iteration, we first evaluate the performances of all the M users on the MAR training set, to obtain the current pivot user set $\mathbf{P}_{+}^{(\mathbf{n})}$ and non-pivot user set $\mathbf{P}_{-}^{(\mathbf{n})}$ (line 3). Then, for each non-pivot user, we find its most similar pivot user (line 4). Then, we calculate the SSL loss $\mathcal{L}_{SSL}$ (line 5). Then we obtain the overall loss $\mathcal{L}$ as follows to combine the original objective function and the SSL loss (Line 6 in Algorithm 1):

$$\mathcal{L} = \mathcal{L}_M + \mu \cdot \mathcal{L}_{SSL} \tag{5}$$

where μ is a hyper-parameter for weighting the contribution of the SSL loss. Afterward, we update the rating model $\mathbf{M}$ by optimizing the overall loss $\mathcal{L}$ (line 7). Finally, we evaluate the performance of the current rating model on the validation set. The iteration will stop when the validation performance doesn't improve for consecutive p epochs.

5 EXPERIMENT

In order to validate the performance of our proposed framework, we conduct extensive experiments on a real-world recommendation dataset. Through the experiments, we seek to answer three questions:

Table 1: Performance comparison on Yahoo! dataset. In the table, we report the results averaged over 5 runs with different random seeds with the 95% confidence intervals.

Methods	MF		NFM	
	MAE	MSE	MAE	MSE
Vanilla	1.071 ± 0.002	1.461 ± 0.004	1.191 ± 0.006	2.025 ± 0.018
CPT-v	0.914 ± 0.003	1.181 ± 0.004	0.985 ± 0.012	1.527 ± 0.010
IPS	0.877 ± 0.001	1.153 ± 0.003	0.866 ± 0.023	1.211 ± 0.057
HEI	1.158 ± 0.003	1.895 ± 0.005	1.225 ± 0.009	2.117 ± 0.023
DR	0.842 ± 0.001	1.257 ± 0.014	0.841 ± 0.014	1.066 ± 0.016
DR-LTD	0.792 ± 0.001	0.983 ± 0.002	0.802 ± 0.006	1.063 ± 0.011
RDC	**0.769** ± 0.003	**0.981** ± 0.003	**0.790** ± 0.006	**0.982** ± 0.003

- **RQ1**: How does our model perform compared with traditional recommendation models that don't explicitly deal with selection bias?
- **RQ2**: How does our method perform compared with other representative recommendation models that explicitly deal with selection bias?
- **RQ3**: How do the choice of hyper-parameters influence the performance of our proposed model?

5.1 Dataset

To compare the performance of various methods on mitigating selection bias, we should evaluate the prediction performance of the resulting recommendation models on unbiased data (i.e. MAR data). Thus, we conduct the experiment on a real-world recommendation dataset with MAR test data.

Yahoo! Dataset. The Yahoo dataset [10] collects the 5-star scale ratings for 1,000 songs from 15,400 users. In the dataset, there are 311,704 biased ratings which are rated by all 15,400 users in a self-selection manner. These ratings make up our training set. In addition, a subset of 5,400 users are asked to rate 10 randomly selected songs, which results in 54,000 unbiased ratings as the test data.

5.2 Baselines

We compare our model with the vanilla recommendation model, as well as several representative debiased models: CPT-v [10], HEI [17], IPS [16], DR [18], LTD [19]. A brief introduction of the baselines can be found in Appendix B.

5.3 Experimental Results

The details of the implementation of our proposed model and the baselines can be found in Appendix C. The main experimental results are shown in Table 1. We report the mean-absolute-error (MAE) and the mean-square-error (MSE) of the rating models trained under various debiasing approaches on the MAR test data. We make the following observations.

First, among all the approaches, our RDC achieves the best performance. Compared with the above data imputation or inverse propensity score based approaches, our SSL framework focuses on the data themselves and mitigates the selection bias through the calibration and correction among different users, which avoids the

heavy dependence on the quality of the imputation model and IPS estimation model. Thus, it achieves better and more robust debiasing performance. Second, the performances of different debiasing methods are quite various. CPT-v and IPS significantly mitigate the selection bias and improves the performance of the rating model by leveraging data imputation and IPS weighting technologies. DR further improves the performance by combining these two tricks, which ensures the model's inference quality when one of the two components cannot work well. Based on DR, DR-LTD introduces an adaptive propensity score estimation model, which further strengthens its debiasing ability. Third, our RDC achieves the best performances on both MF and NFM, which demonstrates that our debiasing framework can be flexibly and successfully applied to different base models. Finally, the deviations of all the results are small, which suggests that the performances of all the approaches are stable and convincing.

5.4 Parameter Sensitivity Analysis

In this section, we investigate the sensitivity of the proposed framework RDC w.r.t three key parameters, i.e, μ, that is the weight of the SSL loss in the overall loss function; τ, which is the temperature value in the distribution feature function; and k, which is the threshold for determining pivot users. Specifically, we evaluate how RDC performs with changing the value of one parameter, while keeping other parameters fixed.

In Figure 2, we report the performances of RDC in terms of MAR on the Yahoo! dataset. The model achieves a high MAE when $\mu = 0$, that is, the SSL loss is not used. The MAE decreases with the weight of SSL loss increases to 1, and when the weight continues to increase to 2, the MAE maintains stable. The observation indicates that the SSL loss plays an important role in improving the performance of the model on MAR data. For τ, we find when it varies in the interval $[1.0, 2.0]$, the performance of the model is stable, which demonstrates that when τ changes in a reasonable range, the model is insensitive to this parameter. Finally, RDC achieves the best performance when the threshold for determining pivot users $t = 0.3$. When $t \to 0$, the performance drops significantly. This is because when t gets smaller, we have a more stringent standard for selecting pivot users, which leads to fewer pivot users but they cannot provide sufficient calibration guidance.

6 CONCLUSION

Selection bias, which is introduced by the self-selection behavior of the users in providing explicit feedback, has been shown to seriously damaged the performance of recommendation models. In this work, we investigate the mitigation of selection bias via a novel self-supervised learning based framework Rating Distribution Calibration (RDC) to alleviate the negative impacts of selection bias on training classical recommendation models. In addition to the original training objective, we introduce a rating distribution calibration loss which aims to correct the predicted rating distribution of a biased user by forcing them to be close to that of similar unbiased users. Extensive experiments are conducted on a real-world recommendation dataset and the results show that our proposed framework outperforms the original models as well as state-of-the-art debiasing approaches by significant margins. We also conduct a parameter sensitivity analysis for our proposed framework. As future work, we are going to investigate the possibility of leveraging SSL to alleviate selection bias in other recommendation scenarios, such as social recommendation.

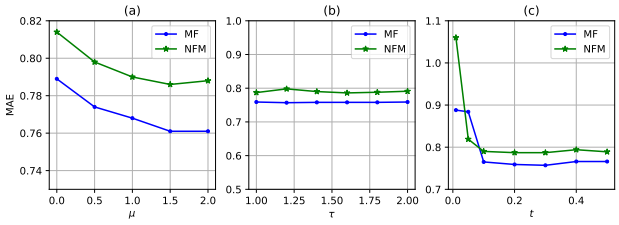

Figure 2: Parameter sensitiveness. (a) μ for the SSL loss weight; (b) τ for the temperature value; (c) t for the pivot user threshold.

REFERENCES

[1] Simon Caton and Christian Haas. 2020. Fairness in machine learning: A survey. arXiv preprint arXiv:2010.04053 (2020).

[2] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. arXiv preprint arXiv:2010.03240 (2020).

[3] Carl Doersch, Abhinav Gupta, and Alexei A Efros. 2015. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision. 1422–1430.

[4] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018).

[5] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. 355–364.

[6] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In International Conference on Machine Learning. PMLR, 1512–1520.

[7] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. 2020. Self-supervised learning on graphs: Deep insights and new direction. arXiv preprint arXiv:2006.10141 (2020).

[8] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Anil K. Jain, and Jiliang Tang. 2021. Trustworthy AI: A Computational Perspective.

[9] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, and Zhang Xiong. 2021. Contrastive Learning for Recommender System. arXiv preprint arXiv:2101.01317 (2021).

[10] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In Proceedings of the third ACM conference on Recommender systems. 5–12.

[11] Benjamin M Marlin, Richard S Zemel, Sam Roweis, and Malcolm Slaney. 2007. Collaborative filtering and the missing at random assumption. In Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence. 267–275.

[12] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised learning of visual representations by solving jigsaw puzzles. In European conference on computer vision. Springer, 69–84.

[13] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. 2017. Learning features by watching objects move. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2701–2710.

[14] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition. 2536–2544.

[15] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In Recommender systems handbook. Springer, 1–35.

[16] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In international conference on machine learning. PMLR, 1670–1679.

[17] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 713–722.

[18] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In International Conference on Machine Learning. PMLR, 6638–6647.

[19] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2021. Combating Selection Biases in Recommender Systems with a Few Unbiased Ratings. In Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 427–435.

[20] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 1893–1902.

A PROOF ON THE UNBIASED ESTIMATION OF $\mathcal{E}$

We use an observation matrix $O^T = (O_{u,i})$ to denote the rating $r_{u,i}$ is observed ($O^T_{u,i} = 1$) or not ($O^T_{u,i} = 0$) in the MAR test set $\mathcal{R}^{\text{test}}$. Given $\mathcal{R}^{\text{test}}$ is missing at random, we have $P(O^T_{u,i} = 1) = \frac{|\mathcal{R}^{\text{test}}|}{|\mathcal{R}|}$. Then

$$\begin{aligned}
\mathbb{E}_{O^T_{u,i}}(\mathcal{E}^{\text{te}}) &= \frac{1}{|\mathcal{R}^{\text{test}}|} \sum_{(u,i,r_{u,i})\in\mathcal{R}} \mathbb{E}_{O^T_{u,i}}[O^T_{u,i} \times (\hat{r}_{u,i} - r_{u,i})^2] \\
&= \frac{1}{|\mathcal{R}^{\text{test}}|} \sum_{(u,i,r_{u,i})\in\mathcal{R}} P(O^T_{u,i} = 1) \times (\hat{r}_{u,i} - r_{u,i})^2 \\
&= \frac{1}{|\mathcal{R}^{\text{test}}|} \sum_{(u,i,r_{u,i})\in\mathcal{R}} \frac{|\mathcal{R}^{\text{test}}|}{|\mathcal{R}|} \times (\hat{r}_{u,i} - r_{u,i})^2 \\
&= \frac{1}{|\mathcal{R}|} \sum_{(u,i,r_{u,i})\in\mathcal{R}} (\hat{r}_{u,i} - r_{u,i})^2 = \mathcal{E}
\end{aligned}$$

Hence, the error on the MAR test set $\mathcal{E}^{\text{test}}$ is an unbiased estimation of the error on the full rating matrix $\mathcal{E}$.

B BASELINES

In this section, we introduce the representative baseline approaches that we compare our method with.

Vanilla: The base recommendation model that doesn't explicitly deal with the selection bias.

CPT-v [10]: A generative model that jointly models the generative process of rating values and the missing mechanism. The missing mechanism is based on the assumption that the probability that a rating is observed depends only on that underlying rating value.

Heuristic error imputation (HEI) [17]: In HEI, a heuristic error imputation module is employed to estimate the prediction errors of the rating model. Specifically, the prediction error is imputed as a constant hyper-parameter. The rating model is trained by optimizing the real errors on the observed data as well as the imputed errors on the unobserved data.

Inverse Propensity Score (IPS) [16]: The IPS method estimates the probability of observing a rating (i.e. propensity score) and uses the inverse probability score to weight the objective function.

Doubly Robust (DR) [18]: A framework that combines the techniques of error imputation and inverse propensity score. An imputation model and a rating model are trained alternatively and affect each other. Both two models are trained with IPS weighting.

Learning to Debias (LTD) [19]: Based on the above IPS and DR methods, a propensity score estimation model is introduced to adjust the propensity adaptively according to the performance of the rating model on MAR data. Here we implement the LTD model with DR as the basis, which achieves the best performance among all the models using LTD.

C EXPERIMENTAL SETUP

In our experiments, we compare different debiasing tricks on two classical base recommendation models: MF and NFM. For the MF model, we set the size of latent factors for users and items as 40. For the NFM model, the embedding size of users and items is set as 40. Following the bi-interaction layer, an one-layer feed-forward neural network with a hidden size of 100 is adopted to compute the rating prediction. We use Sigmoid[1] as the nonlinear activation function. For the approaches which need to use MAR data in the training phase, i.e., IPS, DR, LTD, and our proposed method, we randomly set aside 5% test data as the required MAR data. We also include these MAR data in the training set to train the rating models, which empirically improves the model performance [19]. As for other baselines, the 5% test data are also used for training to have a fair comparison. For all approaches, we randomly set aside 10% MNAR training data as the validation set and the hyper-parameters are fine-tuned based on the rating model's performance on such validation set.

For baselines using IPS (IPS, DR, DR-LTD), we employ a Naive Bayes estimator [16] to calculate the propensity scores, given there are no available additional features of users and items in the Yahoo! dataset. Our proposed model also uses IPS to weigh the ratings during training.

[1] $\text{Sigmoid}(t) = 1/(1 + \exp(t))$.