Causal Recommendation via Machine Unlearning with a Few Unbiased Data

Meng Li[1] **Haochen Sui**[2]

[1] Baidu, Inc. [2] University of Michigan - Ann Arbor
limeng65@baidu.com hcsui@umich.edu

Abstract

Recommender systems (RS) are increasingly important in social media, entertainment, and e-commerce in the information explosion era. However, the collected data contains many biases such as selection bias, as users are free to choose items to rate, making the collected data not representative of the target population. Recently, many methods such as relabeling-based and reweighting-based have been proposed to mitigate the selection bias. However, the effectiveness of these methods relies on strong assumptions, which are difficult to satisfy in real-world scenarios, leading to sub-optimal debiasing performance. In this paper, we propose a debiasing method from the machine unlearning perspective. Specifically, we first propose a user unlearning rate network to determine which user needs to be unlearned. Then we generate the error-maximizing pseudo-labels for each user and fusion such pseudo-labels and the observed labels based on the learned user unlearning rate to mitigate the selection bias. In addition, we further propose an unlearning to debias training algorithm to achieve unbiased learning of the prediction model. Finally, we conduct extensive experiments on three real-world datasets to validate the effectiveness of our method.

Introduction

Recommender systems (RS) are increasingly important in social media, entertainment, and e-commerce in the information explosion era (Shi, Larson, and Hanjalic 2014; Zhang et al. 2023; Wang et al. 2020a). It provides personalized recommendations to users by analyzing their historical behavior (Koren, Bell, and Volinsky 2009; He et al. 2017; Zhang, Liu, and Wu 2018; Wang et al. 2021a). However, the collected data contains many biases such as selection bias, as users are free to choose items to rate, making the collected data not representative of the target population (Marlin and Zemel 2009; Marlin et al. 2007; Wang et al. 2024b; Yang et al. 2023), which challenges the unbiased learning of RS. That is, due to the difference between the training and inference space, simply adopting empirical risk minimization (ERM) on the training data cannot achieve superior prediction performance on all user-item pairs.

Recently, many methods have been proposed to mitigate the selection bias. The error-inputation-based (EIB) method (Steck 2010) first uses the imputation model to estimate the missing labels and then trains the prediction model using observed labels and pseudo-labels. However, it is difficult to obtain accurate pseudo-labels in practice (Dudík, Langford, and Li 2011; Dai et al. 2022; Li et al. 2023a). The inverse propensity scoring (IPS) based methods adopt the reweighting strategy to the observed data to adjust the distribution (Schnabel et al. 2016; Saito et al. 2020), but due to the data sparsity, it is hard to obtain accurate propensities (Wang et al. 2022a). The doubly robust (DR) based methods use the imputation and propensity models simultaneously, leading to low variance (Wang et al. 2019; Saito 2020), but with shortcomings that the debiasing performance heavily relies on the correct specification of the imputation and the propensity models (Kweon and Yu 2024; Li et al. 2024c). By jointly modeling rating values and user selection, generative models have high explainability but usually with complex and sophisticated models that are hard to train in practice (Marlin and Zemel 2009; Hernández-Lobato, Houlsby, and Ghahramani 2014; Chen et al. 2018). In addition, Wang et al. (2020b) adopts information bottleneck with a variational approximation, and Liu et al. (2022a) and Ding et al. (2022) adopt the knowledge distillation approach to mitigate selection bias but lack guarantees of achieving unbiasedness.

In this paper, we propose a machine unlearning approach opening a new perspective for RS debiasing, with the core observation that different users have different selection mechanisms when choosing items to be rated, and that users with larger selection bias should be forgotten at a greater rate. For example, Figure 1 illustrates a toy example where there are 6 users and 6 items in total, in which each user likes 3 items out of all 6 items, and dislikes the other 3 items. Due to the selection behavior, we can only collect the partial ratings as shown in the middle figure, in which both user A and user B rated one liked item and one disliked item, but user E rated all items he disliked, and user F rated all items she liked. From above, we can conclude that user A and use B rated the items more like a 'random' selection, whereas user E and user F rated items with more self-selection behavior. If we simply use the ERM to train the prediction model with the collected ratings, the rating predictions for user E and user F with more severe self-selection behavior will have a reduced accuracy, as shown in the right figure. Thus, if we aim to have an accurate prediction of the rating matrix over the en-

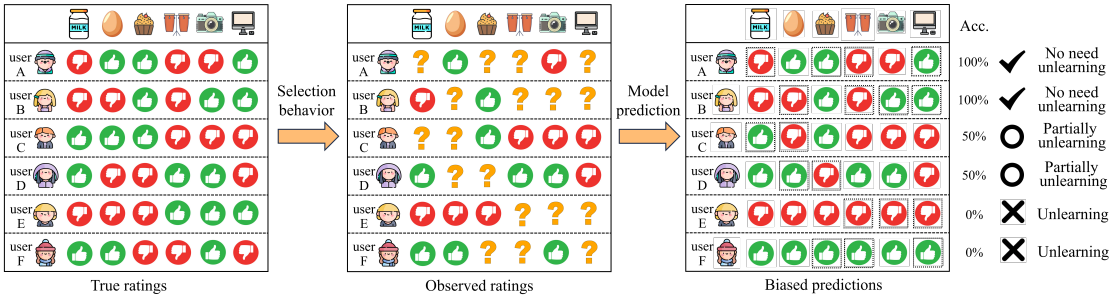

Figure 1: Motivation of unlearning for debiasing in recommendation.

tire user-item pairs, we should prefer to exploit the ratings given by user A and user B for training the prediction model, while downgrading sample weights when training with the ratings given by user E and user F.

Specifically, we propose an unlearning debiasing framework to first capture the user unlearning rate using a user unlearning rate network by fusing a large amount of observational data and a small amount of unbiased rating. Then we learn a pseudo-labeling model to maximize the error between the generated pseudo-labels and predicted labels for all user-item pairs. Finally, we train the prediction model based on the learned error-maximizing pseudo-label, user unlearning rate, and observed labels to achieve unbiased learning.

The contribution of this paper can be summarized below:

- To the best of our knowledge, we are the first work to address selection bias in recommender systems from a machine unlearning perspective.
- We use a user unlearning rate network to capture users that need to be unlearned and combine this with an error-maximizing pseudo-labeling model to unlearn those users to achieve unbiased prediction.
- Experiment results on three real-world datasets verify the effectiveness of the proposed method.

Related Work

Causal Recommendation

Selection bias is one of the most common biases in the RS collected data, which is attributed to the fact that users are free to choose which item to rate (Wang et al. 2023d; Wu et al. 2022; Li et al. 2024b; Chen et al. 2023). This will result in the distribution of the observed population being different from that of the target population (Wang et al. 2022b; Zou et al. 2023; Wang et al. 2024a, 2023a). Specifically, the error-imputation-based (EIB) method (Steck 2010) first uses the imputation model to estimate the missing labels and then trains the prediction model using observed labels and pseudo-labels. However, since the pseudo-labeling model is trained on the observed data with selection bias, the EIB method usually has a large bias due to incorrect imputations (Dudík, Langford, and Li 2011). Another type of method is based on inverse propensity scoring (IPS), which reweights the observed data to adjust the observed data distribution to the target distribution (Schnabel et al. 2016; Saito et al. 2020; Zhang et al. 2024). However, it is difficult to accurately learn the propensities due to data sparsity, and when there are extreme propensity values, its variance can be large (Wang et al. 2022a). Li et al. (2023e) proposes a method for learning covariate balancing propensities, and Li et al. (2023d) further discusses how to learn such propensities. The doubly robust (DR) based method that combines the imputation model and the propensity model are widely-used due to the low variance and the doubly robust property, i.e., it is unbiased either the imputed values or learned propensities are accurate (Wang et al. 2019; Saito 2020), but the debiasing performance of DR-based estimators heavily relies on the proper imputation or propensity learning strategy. DR-based methods have gained increasing attention in recent years (Guo et al. 2021; Li et al. 2023b; Song et al. 2023; Li, Zheng, and Wu 2022; Li et al. 2023a). In addition, Wang et al. (Wang et al. 2020b) and Liu et al. (Liu et al. 2021) use information bottleneck-based method and Yang et al. (Yang et al. 2021) and Wang et al. (Wang et al. 2023b) uses adversarial learning for debiasing.

Moreover, many studies focus on better debiasing with the help of a small amount of unbiased data (Li et al. 2023c; Xiao et al. 2024). For example, Liu et al. (2022a) and Ding et al. (2022) adopt the knowledge distillation approach to mitigate selection bias. Chen et al. (2021) uses meta-learning to leverage unbiased data for debiasing, and Wang et al. (2021b) uses bi-level optimization to learn propensities. Liu et al. (2022b) proposes a self-supervised learning approach to calibrate the rating distribution. Unlike the previous studies, our method selects the samples in the observed data that need to be forgotten, making the distribution of the remaining data the same as the unbiased data.

Recommendation Unlearning

Machine unlearning is the process of removing the influence of specific training data (also known as unlearning target) from a learned model, which stems from the privacy and security concerns of the data provider (Nguyen et al. 2022; Xu et al. 2023). Based on the design details, existing unlearning methods can be broadly categorized into three groups: data reorganization approaches (Wang et al. 2023c), model optimization approaches (Sekhari et al. 2021; Graves, Nagisetty, and Ganesh 2021), and training mechanism approaches (Chun-

dawat et al. 2023; Liu et al. 2022c), which have been widely applied to image data (Tarun et al. 2023), text data (Wu, Dobriban, and Davidson 2020), tabular data (Brophy and Lowd 2021), streaming data (Du et al. 2019), and graph-structured data (Wu et al. 2023).

However, existing machine unlearning methods cannot be directly used on recommender systems. Since recommender systems rely on collaborative information across user-item interaction, arbitrarily dividing the training data into shards could lead to poor performance. To overcome these challenges, RecEraser (Chen et al. 2022) extends SISA (Bourtoule et al. 2021) framework to recommender systems and designs novel data partition algorithms to group similar data into one shard. To enhance model utility, LASER (Li et al. 2023f) also groups similar data together but sequentially trains the recommender model on sub-components instead of training a sub-model for each shard. UltraRE (Li et al. 2024d) refines the design of each stage of RecEraser to reduce complexity without comprimising its efficacy. This type of method, which requires the distributions of a naively retrained model and an unlearned model exactly the same, is known as exact recommendation unlearning. Different from these unlearning methods focusing on model optimization or training mechanism, our approach uses the idea of data reorganization in machine unlearning for debiased learning. To the best of our knowledge, this is the first work that uses unlearning methods for debiased recommendation.

Problem Setup

Let $\mathcal{U} = \{u_1, \ldots, u_m\}$ be a set of users, $\mathcal{I} = \{i_1, \ldots, i_n\}$ a set of items, and $\mathcal{D} = \mathcal{U} \times \mathcal{I}$ the set of all user-item pairs. The rating matrix between users and items is denoted as $\mathbf{R} \in \mathbb{R}^{m \times n}$. The observation matrix is denoted as $\mathbf{O} \in \{0,1\}^{m \times n}$, where $o_{u,i} = 1$ means $r_{u,i}$ is observed and $o_{u,i} = 0$ means $r_{u,i}$ is missing. Denote the features of user and item as $x_{u,i}$, and the predicted rating as $\hat{r}_{u,i} = f_\theta(x_{u,i})$, where $f(\cdot)$ is the prediction model parameterized by θ. If the rating matrix is fully observed, then the prediction model can be unbiasedly trained by minimizing the ideal loss

$$\mathcal{L}_{\text{ideal}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \mathcal{L}(f_\theta(x_{u,i}), r_{u,i}) := \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} e_{u,i},$$

where $\mathcal{L}(\cdot, \cdot)$ is the training loss between the predicted rating $\hat{r}_{u,i}$ and the ground truth rating $r_{u,i}$, with $\mathcal{L}(\cdot, \cdot)$ as an arbitrary pre-defined loss function, e.g., square prediction error $\mathcal{L}(f_\theta(x_{u,i}), r_{u,i}) = (f_\theta(x_{u,i}) - r_{u,i})^2$. In the presence of missing ground truth ratings, the training loss for the corresponding samples cannot be computed, and a naive way is to minimize the average training loss over the observed samples, which is shown below

$$\mathcal{E}_{\text{N}}(\theta) = \frac{1}{|\mathcal{O}|} \sum_{(u,i) \in \mathcal{O}} e_{u,i},$$

where $\mathcal{O} = \{(u,i) \mid (u,i) \in \mathcal{D}, o_{u,i} = 1\}$ is the set of user-item pairs with the observed ratings. However, as users are free to choose which items to rate, so that the observed ratings are not a representative sample of all ratings, resulting in a biased estimate of the ideal loss, *i.e.*, $\mathbb{E}[\mathcal{E}_{\text{N}}(\theta)] \neq \mathcal{L}_{\text{ideal}}(\theta)$. To address this problem,

The IPS estimator uses inverse propensity to reweight the observed data, which is shown below:

$$\mathcal{E}_{IPS}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \frac{o_{u,i} e_{u,i}}{\hat{p}_{u,i}},$$

where $\hat{p}_{u,i} = \pi(x_{u,i}; \psi)$ is the propensity model for estimating the observation probability $p_{u,i} := \mathbb{P}(o_{u,i} = 1 \mid x_{u,i})$.

the DR estimator combines the propensity model $\hat{p}_{u,i}$ and the imputation model $\hat{e}_{u,i}$ as below:

$$\mathcal{E}_{DR}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(u,i) \in \mathcal{D}} \Big[\hat{e}_{u,i} + \frac{o_{u,i}(e_{u,i} - \hat{e}_{u,i})}{\hat{p}_{u,i}} \Big],$$

where $\hat{p}_{u,i} = \pi(x_{u,i}; \psi)$ is for estimating the observation probability $p_{u,i} := \mathbb{P}(o_{u,i} = 1 \mid x_{u,i})$ and $\hat{e}_{u,i} = m(x_{u,i}; \phi)$ is the error imputation model for estimating $e_{u,i}$ using $x_{u,i}$.

However, the unbiasedness condition of IPS and DR estimators is difficult to satisfy in real-world scenarios. This paper focuses on circumventing these assumptions by proposing prediction models unbiased learning method from the user self-selection perspective.

Methodology

Overview of the Methodology

In the real-world scenario, different users will have different rating preferences, *e.g.*, some users tend to randomly pick items to rate, and some users only rate items they like (or dislike). In order to identify these users, we propose to first capture the user unlearning rate using a user unlearning rate network. Then we learn a model designed to maximize the error of pseudo-label learning, and finally based on the learned error-maximizing pseudo-label, user unlearning rate, and observed labels for unbiased learning of the prediction model. We will look into the details of each module.

User Unlearning Rate Network

In this module, we aim to find users that need to be unlearned. Specifically, such users should have the following properties: there is a large discrepancy between the collected user preferences on the observational data and the user's true preferences, *i.e.*, the user rated the items based on his/her own preferences more than randomly. However, true user preferences cannot be obtained from observational data due to the presence of selection bias, so a natural idea is to utilize unbiased data. We define the following loss

$$\mathcal{L}_{\mathcal{A}}(u) = \frac{\sum_{i \in \mathcal{I}} \mathbb{I}[(u,i) \in \mathcal{A}] \cdot (f_\theta(x_{u,i}) - r_{u,i})^2}{\sum_{i \in \mathcal{I}} \mathbb{I}[(u,i) \in \mathcal{A}]},$$

where $\mathcal{A}$ is the unbiased rating set. The core idea of this loss lies in measuring the gap between the prediction performance of a prediction model on observed data and unbiased data. The larger the performance gap, the more likely the user should be unlearned.

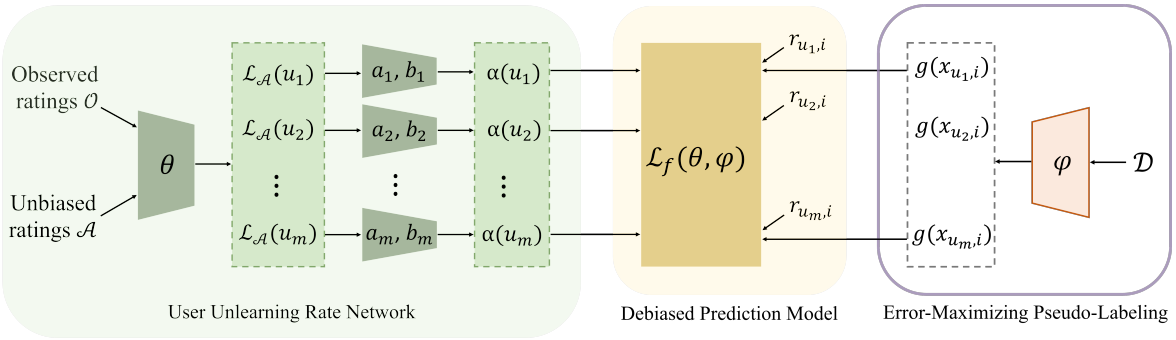

Figure 2: Overview of the proposed model structure.

Algorithm 1: Unlearning to Debias (ULTD)

Input: observed ratings $\mathcal{O}$, unbiased ratings $\mathcal{A}$, and λ

while *stopping criteria is not satisfied* **do**
  for *number of steps for training the unlearning rate network and debiased prediction model* **do**
    Sample a batch $\{(u_j, i_j)\}_{j=1}^{J}$ from $\mathcal{O}$ and a batch $\{(u_k, i_k)\}_{k=1}^{K}$ from $\mathcal{A}$;
    Update the unlearning rate network and debiased prediction model with $\mathcal{L}_f(\theta, \psi)$;
  end
  for *number of steps for training the error-maximizing pseudo-labeling model* **do**
    Sample a batch $\{(u_l, i_l)\}_{l=1}^{L}$ from $\mathcal{D}$;
    Update the error-maximizing pseudo-labeling model with $\mathcal{L}_g(\phi)$;
  end
end

However, the scalability of loss varies greatly for different users, and using loss directly as a measure of the unlearning degree would be unstable. Therefore, we define a post-processing function to learn the unlearning rate and map the loss to the interval $[0, 1]$ with a preserving order. Specifically, we adopt the Platt scaling α_u as shown below

$$\alpha_u(\mathcal{L}_{\mathcal{A}}(u)) = \sigma(s_u \mathcal{L}_{\mathcal{A}}(u) + b_u),$$

where $\sigma(\cdot)$ is the sigmoid function and $\psi = \{s_u, b_u \mid u \in \mathcal{U}\}$ are learnable parameters. To maintain the order, we constrain the parameter $s_u > 0$. Platt scaling has a wide application across diverse domains, including computer vision, natural language processing, and recommender systems.

Error-Maximizing Pseudo-Label

After obtaining the user unlearning rate, we generate adversarial data to forget the user. Specifically, we learn the parameter ϕ of the error-maximizing pseudo-label model $g_\phi(\cdot)$ by the following loss

$$\phi^* = \arg\min_{\phi} \mathbb{E}[-\mathcal{L}(f_\theta(x_{u,i}), g_\phi(x_{u,i}))] + \lambda \cdot \Omega(g_\phi)$$

The model $g_\phi(\cdot)$ is trained using the following loss

$$\mathcal{L}_g(\phi) = -\frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} (f_\theta(x_{u,i}) - g_\phi(x_{u,i}))^2 + \lambda \cdot \Omega(g_\phi)$$

Note that this loss is defined on the whole space because it does not include the observed label. In addition, this loss causes the model $g_\phi(x_{u,i})$ to generate labels that are opposite to the prediction model $f_\theta(x_{u,i})$. So we can learn the unbiased prediction model by balancing the prediction results of model $g_\phi(\cdot)$ and model $f_\theta(\cdot)$ with the appropriate weighting method, which will be introduced in the next part.

Debiased Prediction Model Learning

After obtaining the user unlearning rate and error-maximizing pseudo-label, we combine the observed labels and propose the loss for unbiasedly training the prediction model $f_\theta(\cdot)$:

$$\mathcal{L}_f(\theta, \psi) = \frac{1}{|\mathcal{U}|} \sum_{u\in\mathcal{U}} \alpha_u \left(\frac{\sum_{i\in\mathcal{I}} o_{u,i} (f_\theta(x_{u,i}) - g_\phi(x_{u,i}))^2}{\sum_{i\in\mathcal{I}} o_{u,i}} \right) + \frac{1}{|\mathcal{U}|} \sum_{u\in\mathcal{U}} (1 - \alpha_u) \left(\frac{\sum_{i\in\mathcal{I}} o_{u,i} (f_\theta(x_{u,i}) - r_{u,i})^2}{\sum_{i\in\mathcal{I}} o_{u,i}} \right),$$

In addition, we use joint learning to update the user unlearning rate network, the error-maximizing pseudo-labeling models, and the prediction model alternately. We summarize the training algorithm in Algorithm 1.

Experiment

Dataset and Preprocessing

We have selected three widely used real-world datasets for our experiments: **Coat**[1]: This dataset contains ratings 6,960 biased ratings and 4,640 unbiased ratings from 290 users for 300 items. **Yahoo! R3**[2]: This dataset includes ratings 311,704 biased ratings and 54,000 unbiased ratings from 15,400 users for 1,000 items. We binarize the ratings, assigning 0 to ratings below three and 1 to ratings of three or above. Additionally, we use the fully exposed industrial dataset **KuaiRec** (Gao

[1]https://www.cs.cornell.edu/~schnabts/mnar/
[2]http://webscope.sandbox.Music.com/

Table 1: The debiasing on three datasets. The best two results are bolded, and the best baseline is underlined. Ours-TP and Ours-JL mean that the two-phase (TP) learning and joint learning (JL) are adopted with our methods, respectively.

	Coat			Yahoo! R3			KuaiRec		
Method	AUC ↑	N@5 ↑	R@5 ↑	AUC ↑	N@5 ↑	R@5 ↑	AUC ↑	N@50 ↑	R@50 ↑
MF	0.747	0.513	0.550	0.714	0.556	0.720	0.827	0.565	0.828
IPS	0.751	0.518	0.554	0.726	0.560	0.725	0.815	0.564	0.836
DR	0.756	0.525	0.542	0.721	0.595	0.741	0.830	0.561	0.843
DR-JL	0.759	0.542	0.556	0.721	0.595	0.741	0.823	0.572	0.860
ESCM^2-DR	0.760	0.547	0.563	0.717	0.566	0.727	0.831	0.571	0.852
CausE	0.746	0.512	0.552	0.728	0.552	0.736	0.811	0.561	0.830
KDCRec	0.760	0.524	0.559	0.731	0.575	0.736	0.826	0.571	0.845
LTD	0.754	0.536	0.567	0.724	0.617	0.756	0.791	0.548	0.855
AutoDebias	0.762	0.528	0.568	0.737	0.634	0.780	0.824	0.574	0.859
Res-IPS	0.769	**0.563**	0.584	0.756	0.633	0.781	0.830	0.578	0.876
Res-DR	0.777	0.561	**0.591**	0.746	0.651	0.799	**0.841**	0.581	0.870
Ours-TP	**0.782**	0.556	0.586	**0.761**	**0.654**	**0.803**	0.839	**0.590**	**0.878**
Ours-JL	**0.792**	**0.584**	**0.616**	**0.766**	**0.656**	**0.811**	**0.844**	**0.593**	**0.886**

et al. 2022), which contains 4,676,570 video watching ratio records from 1,411 users for 3,327 videos. For this dataset, we binarize the records by assigning 0 to records with a value less than two, and 1 otherwise.

Baselines

We compared our method with a series of baseline methods widely utilized in debiasing (RS), including matrix factorization (MF) (Koren, Bell, and Volinsky 2009), IPS (Schnabel et al. 2016), DR (Saito 2020), DR-JL (Wang et al. 2019), and ESCM^2-DR (Wang et al. 2022a). In addition, we include the data fusion baselines such as CausE (Bonner and Vasile 2018), KDCRec (Liu et al. 2022a), LTD (Wang et al. 2021b), AutoDebias (Chen et al. 2021), Res-IPS (Li et al. 2024a), and Res-DR (Li et al. 2024a).

Evaluation Metrics and Details

We utilize three widely adopted evaluation metrics: AUC, NDCG@K (N@K), and Recall@K (R@K). For the datasets Coat and Music, we set K = 5, while for KuaiRec, we set K = 50. Throughout the parameter-tuning process, all the methods are implemented on PyTorch with Adam as the optimizer. We tune learning rate in $\{0.005, 0.01, 0.05, 0.1\}$, batch size in $\{32, 64, 128, 256\}$ for **Coat** and $\{1024, 2048, 4096, 8192\}$ for **Yahoo! R3** and **KuaiRec**, and embedding dimension in $\{4, 8, 16, 32, 64\}$ for **Coat** and $\{16, 32, 64, 128, 256\}$ for **Yahoo! R3** and **KuaiRec**. For the user unlearning rate network, we tune the layer number in $\{1, 2, 3\}$ for all three datasets and regularization hyper parameter λ in $\{1e-4, 1e-3, 1e-2, 1e-1, 1\}$. For all experiments, unless explicitly stated, we split 5% unbiased data from the test set to the training set. In our model, we tune the weight β and γ in $\{1e-6, 5e-6, 1e-5, ..., 5e-2, 1e-1\}$.

Performance Comparison

Table 1 shows the prediction performance with varying baselines and our methods. First, the baseline methods outperform the naive method, demonstrating the necessity of addressing bias in recommendation systems. In addition, methods like Res-DR and AutoDebias show strong performance, which is due to the usage of unbiased data. Note that across all three datasets, the proposed methods consistently outperform the baseline methods in AUC, NDCG@K, and Recall@K metrics. These results highlight the robustness and superiority of the proposed methods in both ranking and retrieval tasks, making them highly effective for real-world applications in recommendation systems. Furthermore, we find that joint learning of all models works better than two-phase learning (*i.e.*, learning and fixing the error-maximizing pseudo-labeling model first, and then learning the user unlearning rate network and prediction models).

Conclusion and Future Work

To the best of our knowledge, we are the first work to address selection bias in recommender systems from a machine unlearning perspective. Our proposed method addresses this challenge by introducing a user unlearning rate network, which identifies users whose data should be unlearned to reduce bias. Meanwhile, we generate the error-maximizing pseudo-labels and use these labels for those users who need to be unlearned against the predicted label. we effectively mitigate the selection bias in the data by fusing these pseudo-labels with observed labels. Additionally, our proposed unlearning to debias training algorithm enhances the unbiased learning of the prediction model. Extensive experiments on three real-world datasets demonstrate the effectiveness of our method. There are several avenues for future research to further enhance its capabilities and broaden its applicability. For example, recommender systems often should be applied in dynamic environments where user preferences and item characteristics change over time. It is interesting to integrate our debiasing method with online learning frameworks, which allows the system to continuously use the new data (including both biased and unbiased data) to learn an unbiased model.

References

Bonner, S.; and Vasile, F. 2018. Causal embeddings for recommendation. In *RecSys*.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *IEEE S&P*.

Brophy, J.; and Lowd, D. 2021. Machine unlearning for random forests. In *ICML*.

Chen, C.; Sun, F.; Zhang, M.; and Ding, B. 2022. Recommendation unlearning. In *WWW*.

Chen, J.; Dong, H.; Qiu, Y.; He, X.; Xin, X.; Chen, L.; Lin, G.; and Yang, K. 2021. AutoDebias: Learning to debias for recommendation. In *SIGIR*.

Chen, J.; Dong, H.; Wang, X.; Feng, F.; Wang, M.; and He, X. 2023. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Transactions on Information Systems*, 41: 1–39.

Chen, J.; Wang, C.; Ester, M.; Shi, Q.; Feng, Y.; and Chen, C. 2018. Social recommendation with missing not at random data. In *ICDM*.

Chundawat, V. S.; Tarun, A. K.; Mandal, M.; and Kankanhalli, M. 2023. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher. In *AAAI*.

Dai, Q.; Li, H.; Wu, P.; Dong, Z.; Zhou, X.-H.; Zhang, R.; Zhang, R.; and Sun, J. 2022. A generalized doubly robust learning framework for debiasing post-click conversion rate prediction. In *SIGKDD*.

Ding, S.; Feng, F.; He, X.; Jin, J.; Wang, W.; Liao, Y.; and Zhang, Y. 2022. Interpolative distillation for unifying biased and debiased recommendation. In *SIGIR*.

Du, M.; Chen, Z.; Liu, C.; Oak, R.; and Song, D. 2019. Lifelong anomaly detection through unlearning. In *CCS*.

Dudík, M.; Langford, J.; and Li, L. 2011. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*.

Gao, C.; Li, S.; Lei, W.; Chen, J.; Li, B.; Jiang, P.; He, X.; Mao, J.; and Chua, T.-S. 2022. KuaiRec: A fully-observed dataset and insights for evaluating recommender systems. In *CIKM*.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. Amnesiac machine learning. In *AAAI*.

Guo, S.; Zou, L.; Liu, Y.; Ye, W.; Cheng, S.; Wang, S.; Chen, H.; Yin, D.; and Chang, Y. 2021. Enhanced doubly robust learning for debiasing post-click conversion rate estimation. In *SIGIR*.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural Collaborative Filtering. In *WWW*.

Hernández-Lobato, J. M.; Houlsby, N.; and Ghahramani, Z. 2014. Probabilistic matrix factorization with non-random missing data. In *ICML*.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Kweon, W.; and Yu, H. 2024. Doubly Calibrated Estimator for Recommendation on Data Missing Not At Random. In *WWW*.

Li, H.; Dai, Q.; Li, Y.; Lyu, Y.; Dong, Z.; Zhou, X.-H.; and Wu, P. 2023a. Multiple robust learning for recommendation. In *AAAI*.

Li, H.; Lyu, Y.; Zheng, C.; and Wu, P. 2023b. TDR-CL: Targeted Doubly Robust Collaborative Learning for Debiased Recommendations. In *ICLR*.

Li, H.; Wu, K.; Zheng, C.; Xiao, Y.; Wang, H.; Geng, Z.; Feng, F.; He, X.; and Wu, P. 2024a. Removing hidden confounding in recommendation: a unified multi-task learning approach. In *NeurIPS*.

Li, H.; Xiao, Y.; Zheng, C.; and Wu, P. 2023c. Balancing unobserved confounding with a few unbiased ratings in debiased recommendations. In *WWW*, 1305–1313.

Li, H.; Xiao, Y.; Zheng, C.; Wu, P.; Chen, X.; Geng, Z.; and Cui, P. 2023d. Adaptive Causal Balancing for Collaborative Filtering. In *ICLR*.

Li, H.; Xiao, Y.; Zheng, C.; Wu, P.; and Cui, P. 2023e. Propensity Matters: Measuring and Enhancing Balancing for Recommendation. In *ICML*.

Li, H.; Zheng, C.; Ding, S.; Feng, F.; He, X.; Geng, Z.; and Wu, P. 2024b. Be Aware of the Neighborhood Effect: Modeling Selection Bias under Interference for Recommendation. In *ICLR*.

Li, H.; Zheng, C.; Wang, S.; Wu, K.; Wang, E.; Wu, P.; Geng, Z.; Chen, X.; and Zhou, X.-H. 2024c. Relaxing the Accurate Imputation Assumption in Doubly Robust Learning for Debiased Collaborative Filtering. In *ICML*.

Li, H.; Zheng, C.; and Wu, P. 2022. StableDR: Stabilized Doubly Robust Learning for Recommendation on Data Missing Not at Random. In *ICLR*.

Li, Y.; Chen, C.; Zhang, Y.; Liu, W.; Lyu, L.; Zheng, X.; Meng, D.; and Wang, J. 2024d. Ultrare: Enhancing receraser for recommendation unlearning via error decomposition. In *NeurIPS*.

Li, Y.; Chen, C.; Zheng, X.; Liu, J.; and Wang, J. 2023f. Making recommender systems forget: Learning and unlearning for erasable recommendation. *Knowledge-Based Systems*, 283: 111124.

Liu, D.; Cheng, P.; Lin, Z.; Luo, J.; Dong, Z.; He, X.; Pan, W.; and Ming, Z. 2022a. KDCRec: Knowledge distillation for counterfactual recommendation via uniform data. *IEEE Transactions on Knowledge and Data Engineering*, 35(8): 8143–8156.

Liu, D.; Cheng, P.; Zhu, H.; Dong, Z.; He, X.; Pan, W.; and Ming, Z. 2021. Mitigating confounding bias in recommendation via information bottleneck. In *RecSys*.

Liu, H.; Tang, D.; Yang, J.; Zhao, X.; Liu, H.; Tang, J.; and Cheng, Y. 2022b. Rating distribution calibration for selection bias mitigation in recommendations. In *WWW*.

Liu, Y.; Xu, L.; Yuan, X.; Wang, C.; and Li, B. 2022c. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *INFOCOM*.

Marlin, B.; Zemel, R. S.; Roweis, S.; and Slaney, M. 2007. Collaborative filtering and the missing at random assumption. In *UAI*.

Marlin, B. M.; and Zemel, R. S. 2009. Collaborative prediction and ranking with non-random missing data. In *RecSys*.

Nguyen, T. T.; Huynh, T. T.; Nguyen, P. L.; Liew, A. W.-C.; Yin, H.; and Nguyen, Q. V. H. 2022. A survey of machine unlearning. *arXiv preprint arXiv:2209.02299*.

Saito, Y. 2020. Doubly robust estimator for ranking metrics with post-click conversions. In *RecSys*.

Saito, Y.; Yaginuma, S.; Nishino, Y.; Sakata, H.; and Nakata, K. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *WSDM*.

Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember what you want to forget: Algorithms for machine unlearning. In *NeurIPS*.

Shi, Y.; Larson, M.; and Hanjalic, A. 2014. Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. *ACM Computing Surveys*, 47(1): 1–45.

Song, Z.; Chen, J.; Zhou, S.; Shi, Q.; Feng, Y.; Chen, C.; and Wang, C. 2023. CDR: Conservative doubly robust learning for debiased recommendation. In *CIKM*.

Steck, H. 2010. Training and testing of recommender systems on data missing not at random. In *SIGKDD*.

Tarun, A. K.; Chundawat, V. S.; Mandal, M.; and Kankanhalli, M. 2023. Fast yet effective machine unlearning. *IEEE Transactions on Neural Networks and Learning Systems*, 35(9): 13046–13055.

Wang, F.; Zhong, W.; Xu, X.; Rafique, W.; Zhou, Z.; and Qi, L. 2020a. Privacy-aware cold-start recommendation based on collaborative filtering and enhanced trust. In *DSAA*.

Wang, F.; Zhu, H.; Srivastava, G.; Li, S.; Khosravi, M. R.; and Qi, L. 2021a. Robust collaborative filtering recommendation with user-item-trust records. *TCSS*.

Wang, H.; Chang, T.-W.; Liu, T.; Huang, J.; Chen, Z.; Yu, C.; Li, R.; and Chu, W. 2022a. Escm2: Entire space counterfactual multi-task model for post-click conversion rate estimation. In *SIGIR*.

Wang, H.; Kuang, K.; Chi, H.; Yang, L.; Geng, M.; Huang, W.; and Yang, W. 2023a. Treatment effect estimation with adjustment feature selection. In *KDD*.

Wang, H.; Kuang, K.; Lan, L.; Wang, Z.; Huang, W.; Wu, F.; and Yang, W. 2024a. Out-of-distribution generalization with causal feature separation. *IEEE Transactions on Knowledge and Data Engineering*, 36(4): 1758–1772.

Wang, H.; Yang, W.; Yang, L.; Wu, A.; Xu, L.; Ren, J.; Wu, F.; and Kuang, K. 2022b. Estimating Individualized Causal Effect with Confounded Instruments. In *KDD*.

Wang, J.; Li, H.; Zhang, C.; Liang, D.; Yu, E.; Ou, W.; and Wang, W. 2023b. Counterclr: Counterfactual contrastive learning with non-random missing data in recommendation. In *ICDM*.

Wang, L.; Chen, T.; Yuan, W.; Zeng, X.; Wong, K.-F.; and Yin, H. 2023c. Kga: A general machine unlearning framework based on knowledge gap alignment. *arXiv preprint arXiv:2305.06535*.

Wang, L.; Ma, C.; Wu, X.; Qiu, Z.; Zheng, Y.; and Chen, X. 2024b. Causally Debiased Time-aware Recommendation. In *WWW*.

Wang, W.; Zhang, Y.; Li, H.; Wu, P.; Feng, F.; and He, X. 2023d. Causal Recommendation: Progresses and Future Directions. In *SIGIR*.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *ICML*.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2021b. Combating selection biases in recommender systems with a few unbiased ratings. In *WSDM*.

Wang, Z.; Chen, X.; Wen, R.; Huang, S.-L.; Kuruoglu, E.; and Zheng, Y. 2020b. Information theoretic counterfactual learning from missing-not-at-random feedback. In *NeurIPS*.

Wu, J.; Yang, Y.; Qian, Y.; Sui, Y.; Wang, X.; and He, X. 2023. GIF: A General Graph Unlearning Strategy via Influence Function. In *WWW*.

Wu, P.; Li, H.; Deng, Y.; Hu, W.; Dai, Q.; Dong, Z.; Sun, J.; Zhang, R.; and Zhou, X.-H. 2022. On the Opportunity of Causal Learning in Recommendation Systems: Foundation, Estimation, Prediction and Challenges. In *IJCAI*.

Wu, Y.; Dobriban, E.; and Davidson, S. 2020. Deltagrad: Rapid retraining of machine learning models. In *ICML*.

Xiao, Y.; Li, H.; Tang, Y.; and Zhang, W. 2024. Addressing Hidden Confounding with Heterogeneous Observational Datasets for Recommendation. In *NeurIPS*.

Xu, H.; Zhu, T.; Zhang, L.; Zhou, W.; and Yu, P. S. 2023. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1): 1–36.

Yang, M.; Cai, G.; Liu, F.; Jin, J.; Dong, Z.; He, X.; Hao, J.; Shao, W.; Wang, J.; and Chen, X. 2023. Debiased recommendation with user feature balancing. *ACM Transactions on Information Systems*, 41(4): 1–25.

Yang, M.; Dai, Q.; Dong, Z.; Chen, X.; He, X.; and Wang, J. 2021. Top-n recommendation with counterfactual user preference simulation. In *CIKM*.

Zhang, H.; Liu, G.; and Wu, J. 2018. Social collaborative filtering ensemble. In *PRICAI*.

Zhang, H.; Luo, F.; Wu, J.; He, X.; and Li, Y. 2023. LightFR: Lightweight federated recommendation with privacy-preserving matrix factorization. *ACM Transactions on Information Systems*, 41(4): 1–28.

Zhang, H.; Wang, S.; Li, H.; Zheng, C.; Chen, X.; Liu, L.; Luo, S.; and Wu, P. 2024. Uncovering the Propensity Identification Problem in Debiased Recommendations. In *ICDE*.

Zou, H.; Wang, H.; Xu, R.; Li, B.; Pei, J.; Jian, Y. J.; and Cui, P. 2023. Factual Observation Based Heterogeneity Learning for Counterfactual Prediction. In *CCLR*.

Reproducibility Checklist

Unless specified otherwise, please answer "yes" to each question if the relevant information is described either in the paper itself or in a technical appendix with an explicit reference from the main paper. If you wish to explain an answer further, please do so in a section titled "Reproducibility Checklist" at the end of the technical appendix.

This paper:

Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes)

Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes)

Provides well marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes)

Does this paper make theoretical contributions? (yes)

- All assumptions and restrictions are stated clearly and formally. (yes)
- All novel claims are stated formally (e.g., in theorem statements). (yes)
- Proofs of all novel claims are included. (yes)
- Proof sketches or intuitions are given for complex and/or novel results. (yes)
- Appropriate citations to theoretical tools used are given. (yes)
- All theoretical claims are demonstrated empirically to hold. (yes)
- All experimental code used to eliminate or disprove claims is included. (partial)

Does this paper rely on one or more datasets? (yes)

- A motivation is given for why the experiments are conducted on the selected datasets (yes)
- All novel datasets introduced in this paper are included in a data appendix. (yes)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying. (yes)

Does this paper include computational experiments? (yes)

- Any code required for pre-processing data is included in the appendix. (yes)
- All source code required for conducting and analyzing the experiments is included in a code appendix. (partial)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from. (yes)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (yes)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes)
- This paper states the number of algorithm runs used to compute each reported result. (yes)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (yes)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (partial)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes)