Correcting for Selection Bias in Graph Collaborative Filtering

Md Aminul Islam
University of Illinois at Chicago
mislam34@uic.edu

Ahmed Sayeed Faruk
University of Illinois at Chicago
afaruk2@uic.edu

1 Abstract

Graph-based collaborative filtering (CF) methods have emerged as powerful tools for personalized recommendations by leveraging the relational structure of user-item interactions. However, selection bias—a systematic discrepancy arising from the fact that user-item interactions are typically influenced by exposure mechanisms rather than purely user preferences—remains a critical challenge in ensuring the fairness and accuracy of these models. Without addressing this bias, GNN-based CF models may produce skewed predictions, adversely affecting recommendation quality and fairness. In this work, we propose a novel correction framework to effectively mitigate the impact of selection bias in GNN-based collaborative filtering. Our approach leverages a debiasing mechanism that explicitly models exposure probability, integrating it into GNN architecture. We evaluate our framework on two benchmark datasets, Epinions and Gowalla. Experiment results demonstrate the effectiveness of our approach compared to some state-of-the-art baselines in addressing selection bias and improving recommendation quality.

2 Introduction

Recommender systems have become a fundamental component of online platforms, enhancing online user experiences by inferring user preferences and personalizing content. These systems frequently rely on user interaction data, such as clicks or purchases, to infer user preferences. This reliance is mainly due to the ease of collecting such interaction data. Collaborative filtering (CF) methods model users' preferences based on their historical interactions with items [4]. Recently, graph neural networks (GNNs) have gained attention for their ability to represent recommendation tasks as graphs. In recommender systems, user-item interactions can be modeled as a bipartite graph, where users and items are represented as nodes and edges represent interactions between users and items. Moreover, GNN-based CF techniques can leverage higher-order neighbor information by aggregating user and item embeddings [20]. This aggregation allows GNNs to learn high-quality representations of users and items, resulting in state-of-the-art performance in recommendation tasks [2].

Although the interaction data provides valuable insights, it is often subject to various biases [1], with selection bias being one of the most prevalent. Selection bias occurs when users are not exposed to all relevant items [8], either because the recommender system displays only the top-k items to a particular user, or because the user does not view all the listed results. This prevents the user from observing and interacting with the non-exposed items, even if they are relevant. This results in truncated interaction data for only the items that the user has been exposed to. Training a model to infer user preferences using such biased data can degrade ranking performance, as the system continually reinforces its biased outcomes, creating a feedback loop that negatively impacts user experiences.

Selection bias mitigation has been studied in the context of recommender systems [7, 8, 9, 12, 21, 22]. However, these methods primarily focus on addressing selection bias in content-based recommender systems and overlook the preferences of similar users. Moreover, they do not leverage the capabilities

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

of GNNs, which can capture information from high-order neighbors. These methods typically treat user-item interactions as independent, neglecting the important relational information from their neighbors. There has been a focus on mitigating popularity bias in GNN-based CF methods, a common form of bias in recommender systems where a popular item disproportionately influences recommendations. In GNN-based CF methods, for instance, a popular item can propagate its influence through neighboring nodes during aggregation, further reinforcing biased recommendations for popular items. A recent method [24] has been developed to counteract this effect by applying an inverse popularity score during the aggregation process of GNNs, thereby diminishing the dominance of the popular items. While this method addresses popularity bias in GNN-based CF methods, to the best of our knowledge, no prior work has been proposed to address selection bias in GNN-based CF methods. Reducing selection bias in GNN-based CF models is essential for learning embeddings that represent true users' preferences, rather than biases introduced by the exposure process, while also fully utilizing the potential of GNNs to capture high-order neighborhood information.

In this paper, we propose a new method to mitigate selection bias in GNN-based CF methods. Our approach leverages a debiasing mechanism for selection bias correction that explicitly models exposure probability and integrates it into the GNN architecture to counteract selection bias during the aggregation process.

3 Related Work

Several prior studies [7, 8, 9, 12, 21, 22] have focused on addressing selection bias in content-based and web-search-based recommender systems that rely on implicit interaction data (e.g., clicks). Ovaisi et al. [8, 9] propose selection bias correction approaches based on Heckman's two-stage econometric model [11] to handle selection bias in click data. However, the Heckman model imposes the requirement of linear models in both stages to adhere to the statistical assumptions of the model. Linear models may not be capable of capturing the complex relationships between features and relevance, particularly when working with the high-dimensional feature spaces often seen in recommender systems. To address selection bias, Oosterhuis et al. [7] propose a policy-aware propensity model, which assumes that each relevant item has a non-zero probability of appearing in the top-k ranking positions. This requires multiple logging policies to generate sufficiently distinct ranking orders. Saito [12] and Yuan et al. [22] propose combining propensity-based models with imputation techniques to correct for both selection and position biases. Yuan et al. [21] present a propensity-independent method for selection bias correction using online result randomization, which incorporates both shown and unshown items to estimate an unbiased click-through rate. The use of online result randomization, where an item is displayed in multiple positions for the same query, can negatively affect user experience [1, 18]. Selection bias is also a concern in explicit interaction data (e.g., user ratings) due to the item selection process or users' self-selection behavior [16]. Several bias correction approaches have been proposed for content-based and web-search-based recommender systems to tackle selection bias in explicit interaction data [5, 16, 14, 13, 19]. However, our current work focuses on addressing selection bias in implicit interaction data within GNN-based CF methods.

4 Problem Description

In this paper, we address the issue of selection bias in GNN-based CF methods. Unlike the previous work [24] that primarily addresses popularity bias, our approach aims to correct for selection bias. Selection bias occurs because users are exposed to a subset of relevant items, which creates biased interaction data. Since users cannot interact with unexposed items, certain relevant items are unfairly ignored, leading to biased recommendations. Selection bias in GNNs can be amplified during aggregation as nodes only receive information from interacted neighbors, which are often biased by item exposures. With each GNN layer, this biased neighborhood influence spreads across layers, resulting in embeddings that over-represent commonly exposed items. As a result, nodes with fewer interactions contribute less to the model, reinforcing the skewed representations and diminishing generalizability to less exposed items.

The key objective is to obtain unbiased embeddings for both users and items, ensuring that biased user-item interactions do not distort node representations. After obtaining the unbiased embeddings for users and items, the interaction probability for each user-item pair can be calculated by taking the inner product of their respective node representations. Next, we can recommend the top-N items to a

target user based on the predicted interaction probabilities of items the user has not yet interacted with.

Because the selection bias can be amplified during the aggregation process of GNNs, we aim to adjust for the bias during the aggregation process of GNNs. One possible way is to leverage the inverse propensity weighting (IPW) [6] during the aggregation process of GNNs. Propensity is the probability of exposure for a particular item for a target user. The key idea involves reweighting the neighboring nodes during the aggregation process to mitigate selection bias. Specifically, during the aggregation, we compute the IPW for a user-item pair and apply the IPW as a weight to that pair. As a result, user-item pairs with a higher probability of exposure will receive lower weights, while those with a lower probability of exposure will be assigned higher weights. This approach may enable us to counteract the effects of selection bias during the aggregation process of GNNs, helping to achieve unbiased node representations.

The main challenge of this project is to design an appropriate propensity model that can be used to calculate the IPW for each user-item pair. Therefore, the objective is to design a well-tuned propensity model for use during the aggregation process of GNNs to generate unbiased embedding of nodes. This embedding prevents biased user-item interactions from dominating in node representations.

5 Preliminaries

We begin with the necessary notations used in this paper and then give a more in-depth introduction to the LightGGN [3] backbone, to which we initially apply our method.

5.1 Notations

The historical interactions between users and items of a recommender system are represented with a bipartite graph $\boldsymbol{\mathcal{G}} = (\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}})$. The set of m user nodes $\{v_1, v_2, \ldots, v_m\}$ is denoted with $\boldsymbol{\mathcal{V}}$ and the set of n item nodes $\{v_{m+1}, v_{m+2}, \ldots, v_{m+n}\}$ is denoted with $\boldsymbol{\mathcal{I}}$. The observed interactions between users and items are denoted with $\boldsymbol{\mathcal{V}}$. $\mathbf{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ is a binary adjacency matrix of graph $\boldsymbol{\mathcal{G}}$, where $\mathbf{A}_{u,i} = 1$ indicates an interaction between user node u and item node i, and $\mathbf{A}_{u,i} = 0$ otherwise. The degree of a user node u is denoted with d_u, which is the summation of all the elements in row $\mathbf{A}_u$. A trainable embedding lookup table for k-th layer is denoted with $\mathbf{E}^{(k)} \in \mathbb{R}^{(n+m)\times t}$, where t is embedding dimension, maps user u and item i from one-hot encoding to dense vectors $\mathbf{e}_u^{(k)}$ and $\mathbf{e}_i^{(k)}$, respectively, in k-th layer embedding. Given a target user, the goal is to recommend top-N unconnected items that are likely to be clicked by the user.

5.2 LightGCN

To make things easier at the beginning, we focus on explaining our the proposed method with LightGCN. In the future, we aim to adapt our method to different backbone architectures of GNNs.

LightGCN [3] is a simplified version of GCNs for CF-based methods in recommender systems. LightGCN learns the representations of users and items through the regular message-passing mechanism of GNNs. It keeps the neighborhood aggregation of GNNS and eliminates additional steps, such as feature transformation and non-linear activation, to mitigate the issue of overfitting. The model updates user and item embeddings by aggregating information from their neighbors over several layers. If H is an aggregation function, then the representations at the k-th propagation layer can be written as:

$$\mathbf{E}^{(k)} = H(\mathbf{E}^{(k-1)}, \boldsymbol{\mathcal{G}}). \tag{1}$$

LightGCN does not use non-linear activation and relies on simple linear embedding propagation, specifically a weighted sum, as the aggregation function. Therefore, we can rewrite propagation at k-th layer as:

$$\mathbf{E}^{(k)} = \tilde{\mathbf{A}}\mathbf{E}^{(k-1)}, \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$. Here, $\tilde{\mathbf{A}}$ is the normalized adjacency matrix and $\mathbf{D}$ is a diagonal matrix where the diagonal entries represent the degree of each node in the graph. $\mathbf{D}$ is used to normalize the adjacency matrix $\mathbf{A}$. Normalization helps in ensuring that nodes with extremely high degrees do not disproportionately impact the learned embeddings.

In LightGCN, the pairwise Bayesian Personalized Ranking (BPR) loss [10] is used to maximize the difference in rating scores between positive and negative samples. An L_2 regularization term is also used for the loss function to prevent overfitting and ensure model generalization.

$$\boldsymbol{\mathcal{L}}_{BPR} = \sum_{(u,i)\in\boldsymbol{\mathcal{N}}(u), j\notin\boldsymbol{\mathcal{N}}(u)} -\ln\sigma(y_{ui} - y_{uj}) + \gamma\|\Theta\|_2^2. \tag{3}$$

$\boldsymbol{\mathcal{N}}(u)$ denotes the set of items that user u has previously interacted with. The term y_{ui} denotes the inner product of the final embeddings of user u and item i, which indicates their relevance score. Each j represents an item that user u has not interacted with. The function σ represents the sigmoid activation function, and Θ denotes the embeddings of users and items for the current batch. The parameter γ is a hyperparameter that regulates the impact of the L_2 penalty in the loss function.

6 Methodology

We introduce a novel framework for correcting selection bias in GNN-based collaborative filtering methods. At the core of this framework is a propensity model designed to estimate the exposure probability for each user-item pair. Based on these probabilities, we compute the IPW, which is applied as a per-edge weight during the aggregation process of GNNs to mitigate selection bias. Section 6.1 details the use of IPW in the aggregation process, while Section 6.2 describes various methods for calculating the IPW.

6.1 Selection Bias Debiasing (SBD) Aggregator

Let w_{ui} be the IPW for user u with respect to item i. The IPW for each user-item pair acts as a weight during the aggregation process. Items with a higher probability of being exposed receive a lower IPW, which reduces the aggregation weight with their neighbors. Conversely, underrepresented items are assigned higher IPW, increasing their aggregation weight with neighbors. This gives underrepresented items more opportunities for recommendation. By incorporating these two principles, the aggregation process balances weights between over- and under-represented items, helping to mitigate the effects of selection bias. The user embeddings and item embeddings can be expressed by incorporating the IPW as follows:

$$\mathbf{e}_u^{(k+1)} = \sum_{i\in\boldsymbol{\mathcal{N}}(u)} w_{ui}^{(k)}\mathbf{e}_i^{(k)}; \quad \mathbf{e}_i^{(k+1)} = \sum_{u\in\boldsymbol{\mathcal{N}}(i)} w_{iu}^{(k)}\mathbf{e}_u^{(k)}. \tag{4}$$

GNNs have oversmoothing problem where all node embeddings may become too similar because of aggregations over multiple layers. Therefore, we use residual connections to prevent embeddings from becoming too similar. The embeddings using the residual connections can be written as:

$$\hat{\boldsymbol{e}}_u^{(k)} = \boldsymbol{e}_u^{(k)} + \lambda\boldsymbol{e}_u^{(0)}; \quad \hat{\boldsymbol{e}}_i^{(k)} = \boldsymbol{e}_i^{(k)} + \lambda\boldsymbol{e}_i^{(0)}. \tag{5}$$

Embeddings at $(k+1)^{th}$ layer can be written using the weight and residual connections as follows:

$$\mathbf{e}_u^{(k+1)} = \sum_{i\in\boldsymbol{\mathcal{N}}(u)} w_{ui}^{(k)}\hat{\mathbf{e}}_i^{(k)}; \quad \mathbf{e}_i^{(k+1)} = \sum_{u\in\boldsymbol{\mathcal{N}}(i)} w_{iu}^{(k)}\hat{\mathbf{e}}_u^{(k)}. \tag{6}$$

Following LightGCN, we apply a mean pooling function to the embeddings from all layers to obtain the final representations, which can be written as

$$\boldsymbol{e}_u = \frac{1}{k+1}\sum_{q=0}^{k}\boldsymbol{e}_u^{(q)}; \quad \boldsymbol{e}_i = \frac{1}{k+1}\sum_{q=0}^{k}\boldsymbol{e}_i^{(q)}. \tag{7}$$

Finally, the inner products are computed to obtain the final interaction probability for each user-item pair, represented as $y_{ui} = \boldsymbol{e}_u^\top\boldsymbol{e}_i$. Based on these probabilities, top-N unconnected items to a target user u can be recommended, which are most likely to be clicked by the user u.

6.2 Propensity Model

IPW is a commonly used method to counteract selection and position bias in recommendation systems [7, 17], but has not been used in GNN-based methods to counteract bias. IPW can help mitigate the effect of selection bias during the aggregation process, preventing further propagation of this bias. The propensity score is the probability of exposing an item i to a user u. This propensity score can be estimated based on prior exposure of the item to the user. IPW for a user-item exposure is the reciprocal of the propensity score. To debias the selection bias, we consider the following methods to design a propensity model to estimate the aggregation weights ($w_{ui}^{(k)}$, $w_{iu}^{(k)}$) for user-item pairs in a biased interaction dataset.

6.2.1 Layer-based propensity score estimation

Selection bias can be severe if nodes have a dominance of local neighborhood. To boost distant neighbors and reduce the influence of local neighbors, we assign higher weights to higher layers. This technique incorporates long-range interactions while also keeping local information. In this approach, we apply current layer number as a weight during aggregation as follows:

$$w_{ui}^{k} = \frac{k+1}{\sqrt{d_u d_i}}; \quad w_{iu}^{k} = \frac{k+1}{\sqrt{d_i d_u}},$$

where d_u and d_i denote the degree of user and item, respectively. We denote this method as SBD-LightGCN (M1).

6.2.2 CF-based propensity score estimation

We leverage a matrix factorization model to estimate propensity scores, with observed interactions as positive samples and unobserved ones as negative samples of exposure. Here, we assume that the items interacted with by a user have been exposed to that user, and the items that were not interacted with have not been exposed. We can then predict the probability of exposure for each user-item pair based on the learned user and item embeddings from which we can calculate IPW. In this approach, we train a basic matrix factorization model and estimate inverse propensity scores using pre-trained MF as follows:

$$r_{ui} = 1 - \frac{\mathbf{e}_u^\top \mathbf{e}_i}{||\mathbf{e}_u||_2^2 ||\mathbf{e}_i||_2^2}; \quad r_{iu} = 1 - \frac{\mathbf{e}_i^\top \mathbf{e}_u}{||\mathbf{e}_i||_2^2 ||\mathbf{e}_u||_2^2}.$$

The user and item aggregation weights are estimated as follows:

$$w_{ui}^{k} = \frac{r_{ui}}{\sqrt{d_u d_i}}; \quad w_{iu}^{k} = \frac{r_{iu}}{\sqrt{d_i d_u}}.$$

We denote this method as SBD-LightGCN (M2).

6.2.3 Naive propensity score estimation

We explore a static approach for estimating user propensity scores, where we assume that the likelihood of user interaction increases with the frequency of user exposure, regardless of an item's specific preferences. Therefore, we treat the user interactions as a proxy of user exposures to calculate the user propensity scores. We estimate the user propensity score of a user-item pair as the ratio of observed interactions with the user to the total interactions in the dataset. This user propensity score ignores item-specific factors.

Similarly, we assume that the likelihood of interaction increases with the frequency of item exposure, regardless of a user's specific preferences. Therefore, we treat the item interactions as a proxy of exposures to calculate the item propensity scores. We estimate the item propensity score of a user-item pair as the ratio of observed interactions with the item to the total interactions in the dataset. This item propensity score ignores user-specific factors.

We estimate the user and item aggregation weights using both the user-based and item-based propensity scores as follows:

$$w_{ui}^{k} = \frac{1}{\frac{d_u}{|E|}\frac{d_i}{|E|}}; \quad w_{iu}^{k} = \frac{1}{\frac{d_i}{|E|}\frac{d_u}{|E|}},$$

Dataset	Methods	Recall@20	NDCG@20
Epinions	LightGCN	0.1085	0.0790
	IPS_LightGCN	0.0990	0.0725
	SimGCL	0.1189	0.0867
	GTN	0.1239	0.0914
	SBD-LightGCN (M1)	**0.1281**	**0.0943**
	SBD-LightGCN (M2)	0.1277	0.0940
	SBD-LightGCN (M3)	0.1120	0.0825
	Improvement (%)	+3.39%	+3.17%
Gowalla	LightGCN	0.1677	0.1428
	IPS_LightGCN	0.1305	0.1375
	SimGCL	0.1528	0.1449
	GTN	0.1693	0.1495
	SBD-LightGCN (M1)	**0.1795**	**0.1529**
	SBD-LightGCN (M2)	0.1782	0.1508
	SBD-LightGCN (M3)	0.1715	0.1471
	Improvement (%)	+7.04%	+7.07%

Table 1: Performance of different methods across various datasets.

, where E denotes the total interactions. This approach assigns higher weights to under-represented users/items and thus reduces the dominance of over-represented users/items. We denote this method as SBD-LightGCN (M3).

6.2.4 Additional method

We also explore debiasing techniques for training data to mitigate selection bias before feeding it into GNNs. Specifically, we add new edges with exposure probabilities exceeding a predefined threshold, as estimated by a pre-trained matrix factorization model, and remove edges associated with over-represented nodes to balance the graph structure. While we experiment with various approaches to debias the training data, a finalized method has not yet been determined. Therefore, we exclude the experimental results related to this direction.

7 Evaluations

We evaluate the performance of our proposed framework using two real-world datasets: Gowalla[1] and Epinions[2]. The Gowalla dataset, collected from a location-based social networking platform, contains contains $29,858$ users, $40,981$ items, and $1,027,370$ interactions. The Epinions dataset, derived from a product review website, includes $11,496$ users, $11,656$ items, and $327,942$ interactions.

To ensure consistency with prior studies [15, 23], we filter the datasets to include only users and items with at least ten interactions. For training, 70% interacted items are randomly sampled for each user. To tune hyperparameters, another 10% is used as validation, and the remaining 20% of interactions are used for testing. We evaluate model performance using two widely adopted metrics, Recall@N and NDCG@N to measure the quality of top-N recommendations.

We evaluate our framework over several baseline methods, e.g., LightGCN, IPS-LightGCN, SimGCL, and GTN, using the user-item interaction datasets. We evaluate the performance in terms of Recall@20 and NDCG@20. We follow the experimental setup and parameters for the baseline methods. For our method, SBD-LightGCN (M1), We set $k = 4$. The experimental results are summarized in Table 1.

8 Experimental results

The experimental results, summarized in Table 1, compare the performance of various recommendation models across two datasets, Epinions and Gowalla, in terms of Recall@20 and NDCG@20.

[1] https://snap.stanford.edu/data/loc-gowalla.html

[2] http://www.trustlet.org/downloaded_epinions.html

GTN showed better performance among the baselines in both datasets. The proposed SBD-LightGCN variants outperformed the baselines where SBD-LightGCN (M1) achieved the best results. In Epinions dataset, SBD-LightGCN (M1) showed an improvement of 3.39% in Recall@20 and 3.17% in NDCG@20 compared to the best baseline. These improvements in Gowalla datasets are 7.04% and 7.07%, respectively.

The proposed SBD-LightGCN framework consistently outperformed the baseline methods across both datasets. The largest gains were observed on the Gowalla dataset, showcasing the method's effectiveness in improving recommendation quality.

9 Individual Contributions

In terms of contributions to the writing, Aminul was primarily responsible for the Introduction, Related Work, and Preliminaries sections, while Ahmed focused on the Abstract, Experiments, and Conclusions. Both authors collaborated on the planning of the problem setup, methodology, and experimental design. For implementation, Aminul developed the first two methods presented in Section 6, whereas Ahmed worked on the latter two methods.

10 Conclusion and Future Work

We propose SBD-LightGCN, a novel framework that integrates selection bias correction techniques into the LightGCN architecture. By explicitly modeling the exposure bias present in observed user-item interactions, our method enhances recommendation quality and provides a principled solution to mitigating selection bias in collaborative filtering systems. Experimental results demonstrate the effectiveness of our approach, highlighting the importance of addressing selection bias in graph-based models. For future work, we aim to dynamically update the IPW values across different layers, enabling more flexible and accurate bias adjustments as information propagates through the network. Additionally, we plan to tune the hyperparameters and extend our bias correction framework to other GNN architectures. These advancements will further contribute to the development of fair, robust, and high-performance recommendation systems.

References

[1] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. Unbiased learning to rank with unbiased propensity estimation. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 385–394, 2018.

[2] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Graph neural networks for recommender system. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 1623–1625, 2022.

[3] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[5] Jin Huang, Harrie Oosterhuis, and Maarten de Rijke. It is different when items are older: Debiasing recommendations when selection bias and user preferences are dynamic. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, page 381–389, 2022.

[6] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. Unbiased learning-to-rank with biased feedback. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 781–789, 2017.

[7] Harrie Oosterhuis and Maarten de Rijke. Policy-aware unbiased learning to rank for top-k rankings. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 489–498, 2020.

[8] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. Correcting for selection bias in learning-to-rank systems. In *Proceedings of The Web Conference 2020*, pages 1863–1873, 2020.

[9] Zohreh Ovaisi, Kathryn Vasilaky, and Elena Zheleva. Propensity-independent bias recovery in offline learning-to-rank systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1763–1767, 2021.

[10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[11] Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

[12] Yuta Saito. Doubly robust estimator for ranking metrics with post-click conversions. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 92–100, 2020.

[13] Tobias Schnabel and Paul N Bennett. Debiasing item-to-item recommendations with small annotated datasets. In *Fourteenth ACM Conference on Recommender Systems*, pages 73–81, 2020.

[14] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, page 1670–1679, 2016.

[15] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[16] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 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*, page 427–435, 2021.

[17] Xuanhui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. Learning to rank with selection bias in personal search. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 115–124, New York, NY, USA, 2016. Association for Computing Machinery.

[18] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM)*, pages 610–618, 2018.

[19] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommender systems. In *Fourteenth ACM Conference on Recommender Systems*, pages 426–431, 2020.

[20] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Enhanced graph learning for collaborative filtering via mutual information maximization. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 71–80, 2021.

[21] Bowen Yuan, Jui-Yang Hsia, Meng-Yuan Yang, Hong Zhu, Chih-Yao Chang, Zhenhua Dong, and Chih-Jen Lin. Improving ad click prediction by considering non-displayed events. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 329–338, 2019.

[22] Bowen Yuan, Yaxu Liu, Jui-Yang Hsia, Zhenhua Dong, and Chih-Jen Lin. Unbiased ad click prediction for position-aware advertising systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 368–377, 2020.

[23] Minghao Zhao, Le Wu, Yile Liang, Lei Chen, Jian Zhang, Qilin Deng, Kai Wang, Xudong Shen, Tangjie Lv, and Runze Wu. Investigating accuracy-novelty performance for graph-based collaborative filtering. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 50–59, 2022.

[24] Huachi Zhou, Hao Chen, Junnan Dong, Daochen Zha, Chuang Zhou, and Xiao Huang. Adaptive popularity debiasing aggregator for graph collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 7–17, 2023.