A First Look at Selection Bias in Preference Elicitation for Recommendation

SHASHANK GUPTA, University of Amsterdam, The Netherlands

HARRIE OOSTERHUIS, Radboud Universiteit, The Netherlands

MAARTEN DE RIJKE, University of Amsterdam, The Netherlands

Preference elicitation explicitly asks users what kind of recommendations they would like to receive. It is a popular technique for conversational recommender systems to deal with cold-starts. Previous work has studied selection bias in implicit feedback, e.g., clicks, and in some forms of explicit feedback, i.e., ratings on items. Despite the fact that the extreme sparsity of preference elicitation interactions make them severely more prone to selection bias than natural interactions, the effect of selection bias in preference elicitation on the resulting recommendations has not been studied yet. To address this gap, we take a first look at the effects of selection bias in preference elicitation and how they may be further investigated in the future. We find that a big hurdle is the current lack of any publicly available dataset that has preference elicitation interactions. As a solution, we propose a simulation of a topic-based preference elicitation process. The results from our simulation-based experiments indicate (i) that ignoring the effect of selection bias early in preference elicitation can lead to an exacerbation of overrepresentation in subsequent item recommendations, and (ii) that debiasing methods can alleviate this effect, which leads to significant improvements in subsequent item recommendation performance. Our aim is for the proposed simulator and initial results to provide a starting point and motivation for future research into this important but overlooked problem setting.

CCS Concepts: • **Information systems** → *Recommender systems.*

ACM Reference Format:
Shashank Gupta, Harrie Oosterhuis, and Maarten de Rijke. 2023. A First Look at Selection Bias in Preference Elicitation for Recommendation. In *CONSEQUENCES Workshop at RecSys '23, September 18-22, 2023, Singapore.* ACM, New York, NY, USA, 6 pages.

1 INTRODUCTION

Traditional recommender systems provide a single-shot human-system interface that is static in nature. They often rely on the user's past interactions to infer their preferences and generate a recommendation based on that. Traditional collaborative filtering (CF)-based methods fall into this category [4, 6, 7]. However, these methods have trouble handling settings where user preferences are dynamic – in practice, preferences often drift over time due to external covariates [7] – or single-shot recommendation settings where user intent has to be inferred from contextual information, instead of past interactions [14]. Additionally, these methods struggle to generate good recommendations for cold-start users and items. These issues, coupled with the sparse nature of user-item interaction data, make learning a good recommendation model difficult. A solution to these issues could be asking for a user's preferences directly at a coarser granularity in a *preference elicitation* (PE) stage. Users are generally very willing to indicate or clarify their preferences, when prompted [15].

PE can be used in a variety of settings, including so-called question-based conversational recommender systems (CRSs) [2, 10, 23], which consist of the following main components: (i) *preference elicitation* (PE), where the user's

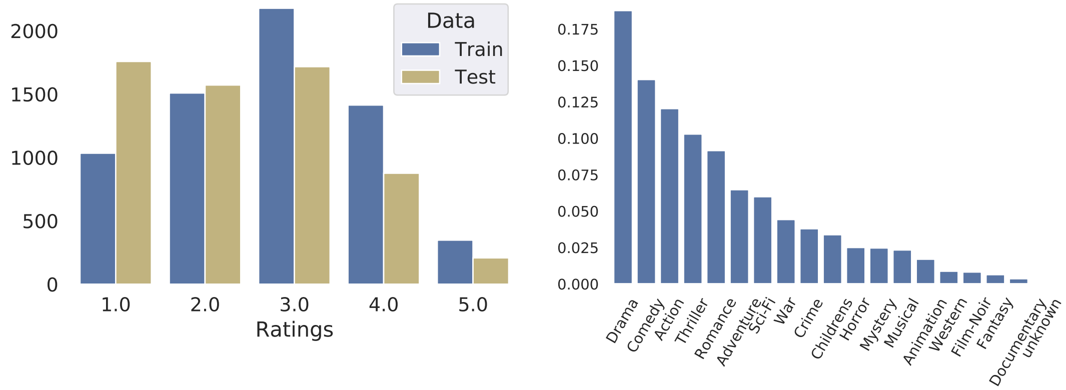

Fig. 1. Rating distribution over item topics on the Coat Music dataset (Left), and Genre popularity in the MovieLens dataset (Right).

preferences on items or item topics are collected or elicited, and, subsequently, (ii) *item recommendation*, where the system generates recommendations for users, conditioned on their response during the PE stage. The interactive aspect of CRSs can help in dealing with dynamic user preferences and the lack of intent information. It can also help with the cold-start problem, by collecting user's preferences on a group of items, instead of on an item directly [1].

Recommender systems are commonly optimized based on logged user interactions. However, such interactions provide a biased view of the actual user preferences [12, 13, 18, 22]. In particular, ratings are generally not evenly spread over all items but are heavily affected by popularity bias, resulting in a small number of items receiving most ratings. Figure 1 (left) demonstrates this effect on the rating distribution of item topics in Coat, a popular recommendation dataset with an unbiased test set [19]. Popularity bias can be seen as a specific form of selection bias, due to which only part of the user preferences are observed in ratings [13]. Importantly, selection bias on the item level propagates to the topic level; for example, Figure 1 demonstrates the popularity distribution over movie genres in the MovieLens dataset. Similar to how selection bias in item ratings results in a biased view over topic preferences, it seems likely that selection bias in a PE stage could negatively affect the subsequent recommendation stage. While selection bias in user interaction data is widely studied [12, 13, 13, 18, 19, 22], to the best of our knowledge, previous work has not considered the effects of selection bias in PE. To address this gap, this work takes a first look at the problem of selection bias in PE for recommendation. We focus on elicitation on the topic-level followed by subsequent item recommendation. Because there is currently no publicly available recommendation dataset that represents PE, we introduce a method for simulating a PE stage from static recommendation datasets. Our experimental results in the simulator reveal that selection bias in the PE stage does, indeed, have negative effects on subsequent item recommendation. We find that existing debiasing methods can be adapted to reduce these effects, leading to significantly better recommendations.

2 CORRECTING FOR SELECTION BIAS IN PREFERENCE ELICITATION

In this section, we discuss how common debiasing methods for item recommendation can be applied to topic-level PE [19]. Let U be the set of all users, I the set of all items, and T the set of all item-topics (referred to as topics hereafter) in the dataset, and $Y \in \{0,1\}^{|U|\cdot|T|}$ the user-topic *complete* rating matrix; $Y_{u,t}$ is the true rating for the pair (u,t). $T \in \{0,1\}^{|I|\cdot|T|}$ is the indicator matrix where $T_{i,t} = 1$ if item i belongs to the topic t. $R \in \{0,1\}^{|U|\cdot|I|}$ is the rating matrix, with entry $R_{u,i}$ indicating user u's rating for item i. In reality, not all entries in the Y matrix are observed; let $O \in \{0,1\}^{|U|\cdot|T|}$ be the observation matrix, with $O_{u,t}$ indicating whether the rating $Y_{u,t}$ is observed or not. The entries in the $Y_{u,t}$ matrix are affected by selection bias. O controls the selection bias, where certain ratings are overrepresented or underrepresented in the dataset; we use $\rho_{u,t} = P(O_{u,t} = 1)$ to denote the probability of observing a rating $Y_{u,t}$ in the dataset.

Ideal rating estimator. An ideal rating prediction loss can be defined as follows:

$$\mathcal{L}_{\text{ideal}} = \frac{1}{|U||T|} \sum_{u,t} L(\hat{y}_{u,t}, y_{u,t}). \tag{1}$$

The loss function $L(\hat{y}_{u,t}, y_{u,t})$ used for rating prediction could be mean squared error (MSE).

Naive rating estimator. One could naively ignore selection bias in the observed rating data and estimate the prediction loss by simple averaging, resulting in the naive training loss estimator:

$$\mathcal{L}_{\text{naive}} = \frac{1}{|\{u, t : O_{u,t} = 1\}|} \sum_{u,t:O_{u,t}=1} L(\hat{y}_{u,t}, y_{u,t}), \tag{2}$$

where $|\{u, t : O_{u,t} = 1\}|$ is the number of observed ratings in the dataset. It is clearly a biased estimator of the ideal-loss (Eq. 1) [19].

Unbiased preference elicitation. To debias the loss function in Eq. 2, we apply inverse propensity scoring (IPS) [8, 18, 19], where the propensity value $\rho_{u,t} = p(O_{u,t} = 1)$ is used as a weight in the loss function. The modified loss function is defined as follows:

$$\mathcal{L}_{\text{ips}} = \frac{1}{|U||T|} \sum_{u,t:O_{u,t}=1} \frac{L(\hat{y}_{u,t}, y_{u,t})}{\rho_{u,t}}. \tag{3}$$

The modified $\mathcal{L}_{\text{ips}}$ is an unbiased estimate of the ideal-loss defined in Eq. 1 [18, 19], i.e., $\mathbb{E}_O[\mathcal{L}_{\text{ips}}] = \mathcal{L}_{\text{ideal}}$.

3 EXPERIMENTS

Below, we discuss the semi-synthetic experimental setup, fully-synthetic setup, followed by empirical results. For details on simulating preference elicitation data, and the synthetic topic generation, we defer to Appendix A.

Yahoo! R3 dataset. This dataset is collected as part of a music-recommendation service; it includes rating information from 15,400 users on 1,000 items, which are self-selected by users, i.e., these are MNAR ratings [21]. A separate test-set comprises of ratings from a uniformly-random policy, ensuring the ratings are free from selection bias. Topic information is not present in the dataset, hence we use the synthetic topic generation method discussed in Appendix A. We use 20% of the unbiased test data to generate the bipartite user-item graph and generate item embeddings, followed by synthetic topic generation, and finally the unbiased PE data (Appendix A). For clustering, we experiment with different numbers of clusters to evaluate the robustness of the method under different setups.

Fully-synthetic dataset. Along with simulating conversations from user-item interactions, we also experiment with a fully-synthetic dataset setting, where we simulate user-topic interactions directly. Following [5], the following two stage process is applied: (i) Given N users and T topics, their corresponding latent-factors for users ($\mathbf{P} \in \mathbf{R}^{N*d}$) and topics ($\mathbf{Q} \in \mathbf{R}^{T*d}$) are generated via Gaussian distribution $\mathcal{N}(0, 1)$. The rating scores are generated via a dot-produce of user and topic latent factors. And (ii) the MNAR logged data is generated via the following mechanism:

$$P(o_{u,t} \mid y_{u,t}) = \alpha P(o_{u,t} \mid y_{u,t}, \text{pos-bias}) + (1 - \alpha) P(o_{u,i} \mid \text{uniform}) \tag{4}$$

The simulator is available at: https://github.com/shashankg7/Bias-Preference-Elicitation.

Table 1. Performance of the debiasing method on the unbiased rating prediction task on the Yahoo! R3 dataset. Significant improvements over the baseline (MF) are marked with † ($p < 0.01$). Average values over 10 different runs are reported.

Exp. setting	**Method**	**MAE↓**	**MSE↓**	**NDCG@3↑**
#clusters = 25	MF	1.3041	2.5634	0.7461
	ExpoMF	1.3075	2.8213	0.7503
	MF-IPS	**0.8327**†	**1.0832**†	**0.7511**†
#clusters = 50	MF	1.3094	2.5857	0.7476
	ExpoMF	1.3050	2.8138	0.7511
	MF-IPS	**0.8268**†	**1.0777**†	**0.7553**†
#clusters = 75	MF	1.3112	2.5887	0.7460
	ExpoMF	0.8451	1.1530	0.7505
	MF-IPS	**0.8451**†	**1.1530**†	**0.7521**†
#clusters = 100	MF	1.3057	2.5403	0.7460
	ExpoMF	1.3109	2.8316	0.7499
	MF-IPS	**0.8464**†	**1.1553**†	**0.7518**†

Table 2. Performance of the debiasing method on the unbiased rating prediction task on the fully-synthetic dataset. Significant improvements over the baseline (MF) are marked with † ($p < 0.01$). Average values over 10 different runs are reported.

Exp. setting	**Method**	**MAE↓**	**MSE↓**	**NDCG@3↑**
$\alpha = 0.25$	MF	0.8449	1.0847	**0.7611**
	ExpoMF	1.6643	3.9344	0.6638
	MF-IPS	**0.7894**†	**0.9874**†	0.7511
$\alpha = 0.5$	MF	0.8666	1.1461	**0.7852**
	ExpoMF	1.6506	3.9178	0.6838
	MF-IPS	**0.7670**†	**0.9185**†	0.7836
$\alpha = 0.75$	MF	0.9012	1.2383	0.8053
	ExpoMF	1.6469	3.9708	0.6984
	MF-IPS	**0.7330**†	**0.8322**†	**0.8230**†
$\alpha = 1.0$	MF	0.9622	1.3974	0.8179
	ExpoMF	1.6473	4.0386	0.7121
	MF-IPS	**0.7254**†	**0.8078**†	**0.8362**†

4 RESULTS

We evaluate the effect of debiasing PE on the unbiased test set. We use mean average error (MAE) and mean squared error (MSE) as evaluation metrics [19] for measuring accuracy in rating prediction. To evaluate the quality of rankings, we use NDCG@3, following Saito [17]. We use ExpoMF [11] as a baseline for debiasing, which uses a generative model to correct for the bias.

Results for the semi-synthetic dataset are presented in Table 1. Results are reported for different numbers of item clusters in the synthetic topic generation (see Section A). Different numbers of clusters represent a different PE setting where the number of item topics varies. Metric values suggest that a naive method for learning rating prediction (using the objective in Eq. 2) results in sub-optimal performance across all settings of clusters. The results suggest that, even for a small-scale PE system (with 35 item-topics), a selection-bias exists, and using IPS for debiasing helps.

For the fully-synthetic setup, results are presented in Table 2. Results are reported for different values of α (see Eq. 4), which represent different levels of selection bias. A lower value of α represents a setting where the second term (with uniform observation probability) dominates, simulating a setting where data is sampled from a uniformly-random policy. Similarly, a higher α value represents a setting with higher positivity-bias. The value of α controls the degree of positivity bias in the simulated logged data. The results from a debiasing rating-prediction method (MF-IPS) are consistent with the results in the semi-synthetic setting for the rating prediction task, for the MAE and MSE metrics. However, for lower values of α (0.25, 0.5), the baseline matrix factorization (MF) outperforms other methods in terms of NDCG. We suspect this is caused by the uniform data generation part dominating the biased counterpart, hence there is less signal for learning user preferences. For higher α values, the results are consistently better for the IPS method. It is also interesting to note that even for the case where the uniformly-random policy dominates ($\alpha = 0.25$), debiasing improves the performance in terms of MAE and MSE.

The results in this section show that a naive method for rating prediction in the PE stage results in a sub-optimal system, which we consistently observe across all experimental setups.

5 CONCLUSION

We have explored the effect of selection bias in PE for recommender systems. We have shown that user-item interactions (ratings) in the preference elicitation stage suffer from the issue of selection bias, which is a common issue when dealing with ratings at the item-level [19]. We have also explored how training a PE system on biased data can lead to error propagation in downstream tasks. To the best of our knowledge, we are the first to explore and identify the issue of bias in the PE stage. We have shown that, similar to the case of static item recommendations, selection bias exists in a PE setting as well.

We have also investigated the application of existing debiasing methods used in item-based recommendation methods, and have shown that these methods can be successfully applied in our setting. Importantly, given a lack of unbiased test collections for evaluating bias in a PE, we have proposed, and are sharing, a simulation method to generate an unbiased test collection for evaluating debiasing methods. Finally, with the release of our simulator and experimental source code, in addition to our comparison of existing methods, we wish to provide a starting point and motivation for future research to further investigate the problem of bias in similar areas. As part of future work, we propose a joint debiasing method for the PE stage and the corresponding downstream tasks.

REFERENCES

[1] Shuo Chang, F Maxwell Harper, and Loren Terveen. 2015. Using Groups of Items for Preference Elicitation in Recommender Systems. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. 1258–1269.

[2] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. Q&R: A Two-Stage Approach toward Interactive Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 139–148.

[3] Ming Gao, Leihui Chen, Xiangnan He, and Aoying Zhou. 2018. BiNE: Bipartite Network Embedding. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 715–724.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web*. 173–182.

[5] Jin Huang, Harrie Oosterhuis, Maarten de Rijke, and Herke van Hoof. 2020. Keeping Dataset Biases out of the Simulation: A Debiased Simulator for Reinforcement Learning based Recommender Systems. In *Fourteenth ACM Conference on Recommender Systems*. 190–199.

[6] Ilija Ilievski and Sujoy Roy. 2013. Personalized News Recommendation based on Implicit Feedback. In *Proceedings of the 2013 International News Recommender Systems Workshop and Challenge*. 10–15.

[7] Dietmar Jannach, Lukas Lerche, and Markus Zanker. 2018. Recommending Based on Implicit Feedback. In *Social Information Access*. Springer, 510–569.

[8] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 781–789.

[9] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations, ICLR*.

[10] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-Action-Reflection: Towards Deep Interaction between Conversational and Recommender Systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 304–312.

[11] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling User Exposure in Recommendation. In *Proceedings of the 25th international conference on World Wide Web*. 951–961.

[12] 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.

[13] 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.

[14] Rishabh Mehrotra, Mounia Lalmas, Doug Kenney, Thomas Lim-Meng, and Golli Hashemian. 2019. Jointly Leveraging Intent and Interaction Signals to Predict User Satisfaction with Slate Recommendations. In *The World Wide Web Conference*. 1256–1267.

[15] Bilih Priyogi. 2019. Preference Elicitation Strategy for Conversational Recommender System. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 824–825.

[16] Douglas A. Reynolds. 2009. Gaussian Mixture Models. *Encyclopedia of Biometrics* 741, 659-663 (2009).

[17] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-not-at-random Explicit Feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 309–318.

[18] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[19] 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.

[20] Adith Swaminathan and Thorsten Joachims. 2015. The Self-normalized Estimator for Counterfactual Learning. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*. 3231–3239.

[21] Yahoo! R3. 2022. R3 - Yahoo! Music Ratings for User Selected and Randomly Selected Songs, version 1.0. URL: https://webscope.sandbox.yahoo.com/catalog.php?datatype=r.

[22] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased Offline Recommender Evaluation for Missing-Not-At-Random Implicit Feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 279–287.

[23] Yichi Zhang, Zhijian Ou, and Zhou Yu. 2020. Task-oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9604–9611.

A EXPERIMENTS

Simulating preference elicitation. To evaluate the effects of an unbiased recommendation method, ideally, we need an unbiased held-out dataset collected with a randomized logging policy at the item-topic level, free from the effects of selection bias [5, 19]. Unfortunately and to the best of our knowledge, no such dataset exists for PE. As a solution, we propose a simple method to simulate a benchmark dataset to evaluate the effects of selection bias in PE. For each topic t, we aggregate the ratings from each item i which belongs to the topic, for both the biased training set and the unbiased test set. As a result, we get a biased training set with user-topic interactions and an unbiased test set without the effects to selection bias, to evaluate the performance of various debiasing methods.

Synthetic topic generation. An item's topic category information is not always guaranteed to be present, for reasons such as privacy constraints from external vendors, noisy or unreliable topic labelling, etc. To deal with this issue, we propose a synthetic topic generation method that only relies on user-item interaction information. Given user-item interactions, we create a bipartite graph $G = \langle V, E \rangle$, where the set of vertices V is divided into two groups, one of which consists of nodes representing users, and the other has nodes representing items. The set E consists of edges between the two groups. Each interaction pair (u, i) results in an edge between the node corresponding to i and u. Given this bipartite-graph, we learn node embeddings via graph representation learning bipartite network embedding (BINE) [3]. We make use of a small unbiased test set to generate the bipartite graph, in an attempt to learn unbiased network embeddings. Given the vector representation of all items from the graph embedding method, we use clustering to group the items in the embedding space. We use Gaussian mixture models [16] to cluster the embeddings. The cluster centers are considered as the topics.

Coat dataset. This dataset consists of user interactions for a coat-recommendation service, which includes ratings from 290 users on 300 items which are self-selected by users, i.e., these are MNAR ratings [19]. For the unbiased test, a uniformly-random policy is deployed to collect unbiased ratings on 10 items. Items are labelled with topics in the dataset, where each item can belong to multiple categories. Propensity scores $P(O_{u,i} = 1)$ are computed using logistic-regression with item covariates.

Hyperparameters. We use 5-fold cross-validation for hyper-parameter tuning in all our experiments. We use Adam [9] for optimizing the model-parameters for the loss-functions defined previously. For hyper-parameter tuning, we use the self normalizing importance sampling (SNIPS) estimator [20], and optimize for MAE.