




Making Alice Appear Like Bob: A Probabilistic Preference Obfuscation Method For Implicit Feedback Recommendation Models

Gustavo Escobedo¹, Marta Moscati¹, Peter Muellner⁴,
Simone Kopeinik⁴, Dominik Kowald⁴, Elisabeth Lex³,
and Markus Schedl^{1,2}

¹ Johannes Kepler University Linz, Linz, Austria
{gustavo.escobedo,marta.moscati,markus.schedl}@jku.at

² Linz Institute of Technology, Linz, Austria

³ Graz University of Technology, Graz, Austria
elisabeth.lex@tugraz.at

⁴ Know-Center GmbH, Graz, Austria
{pmuellner,skopeinik,dkowald}@know-center.at

Abstract. Users’ interaction or preference data used in recommender systems carry the risk of unintentionally revealing users’ private attributes (e.g., gender or race). This risk becomes particularly concerning when the training data contains user preferences that can be used to infer these attributes, especially if they align with common stereotypes. This major privacy issue allows malicious attackers or other third parties to infer users’ protected attributes. Previous efforts to address this issue have added or removed parts of users’ preferences prior to or during model training to improve privacy, which often leads to decreases in recommendation accuracy. In this work, we introduce **SB0**, a novel probabilistic obfuscation method for user preference data designed to improve the accuracy–privacy trade-off for such recommendation scenarios. We apply **SB0** to three state-of-the-art recommendation models (i.e., BPR, MultVAE, and LightGCN) and two popular datasets (i.e., MovieLens-1M and LFM-2B). Our experiments reveal that **SB0** outperforms comparable approaches with respect to the accuracy–privacy trade-off. Specifically, we can reduce the leakage of users’ protected attributes while maintaining on-par recommendation accuracy.

Keywords: Recommender Systems · Privacy · Obfuscation · Debiasing · Implicit Feedback

1 Introduction

Recommender systems (RSs) provide relevant content to their users, commonly based on large collections of users’ historical interaction data with items, using

collaborative filtering techniques. The historical data used for training of and inference in recommendation models consists of interactions of users with several items and hence represent the preference of each user. While such user-item interaction data is necessary to create an accurate recommendation model, it may also reflect inherent biases in user behavior, which are subsequently encoded or even amplified during model training. For instance, users of music recommender systems from different countries and of different genders tend to prefer different artists and genres [15, 17, 29], leading to a correlation between users’ sensitive attributes and behavioral patterns encoded in their interaction data.

As a consequence, this leads to two important risks: possible privacy breaches and stereotypical or even unfair recommendations. As for *privacy* issues, users’ protected information can be leaked when untrusted third parties get access to the users’ interaction data [15] or internal user representation of the model [9, 33]. For instance, for a group of users that is highly correlated with a list of stereotypical items, private attributes (e.g., gender, occupation, or country) can be unveiled through malicious attacks on the model or the data [2]. Concerning *unfairness*, recommendation models trained on interaction data that is correlated with sensitive user attributes have been shown to impact the quality of recommendations across different user groups distinguished by these attributes [23].

Both problems (privacy concerns and fairness issues) are intertwined because they originate from the correlations between users’ interaction behaviors and their sensitive attributes. To mitigate them, several privacy-enhancing methods have been introduced, targeting different stages of the recommendation model’s training process (pre-, in-, and post-processing) [27]. Among the pre-processing methods, user preference obfuscation approaches have been proposed to impede malicious attacks that aim at the leakage of private user attributes before training. These approaches primarily consist of adding or removing carefully selected items from users’ preference data and have specifically been applied to user-item matrices containing ratings [31, 32].

In the work at hand, we introduce *Stereotypicality-Based Obfuscation* (SBO), a probabilistic user preference obfuscation method to counteract inference attacks against private user attributes. Unlike existing methods, SBO selects users and items to obfuscate in a probabilistic fashion, using novel stereotypicality metrics. This limits the number of users whose items require obfuscation and adjusts the selection probability of non-stereotypical items in the sampling process. We demonstrate SBO’s performance in terms of recommendation utility and accuracy of an attacker that aims to unveil the users’ gender. Experiments with three common recommendation algorithms—BPR-MF, LIGHTGCN, and MULTVAE—on two standard recommendation datasets from the movie and music domains—ML-1M (MovieLens) [10] and LFM-2B-100K (Last.fm) [23, 30]—showed a favorable accuracy–privacy trade-off of our method.

In the remainder of the paper, we review relevant previous work (Sect. 2), detail the proposed SBO method (Sect. 3), present the setup of our evaluation experiments (Sect. 4), and discuss results (Sect. 5). Ultimately, we summarize our findings and provide an outlook (Sect. 6).

2 Related Work

Related work belongs to two strands of research: privacy-aware RSs (Sect. 2.1) and fairness in RSs through adversarial training (Sect. 2.2). Both can be addressed by altering the user's input data to the RS or the model's latent user representations.

2.1 Privacy-Aware Recommender Systems

RSs typically expose their users to several privacy risks. For example, the disclosure of information that is used to train the recommendation model (e.g., interaction data) [11, 40] to third parties, or the inference of information that is not used during model training but correlated with the training data (e.g., gender or age) [36, 43].

Various technologies have been employed to address users' privacy concerns, such as homomorphic encryption [14], federated learning [22, 24], and differential privacy [25, 26]. Homomorphic encryption aims to generate privacy-aware recommendations by utilizing encrypted user data [42]. Federated learning operates under the principle that sensitive user data should remain on the user's device [1]. Lastly, differential privacy (DP) is used to counter privacy risks by incorporating a carefully tuned level of random perturbation into the recommender system [5]. Many works apply DP to protect a user's sensitive attribute. However, malicious parties can still scrutinize the generated recommendations to infer protected attributes [8]. This is the case if non-sensitive interaction data correlates with the user's sensitive attributes and forms distinct patterns that can be uncoded.

For this reason, Weinsberg et al. [36] suggest an approach that detects rating data that is indicative of gender and adds ratings for items indicative of the opposite gender to obfuscate a user's real gender. However, the authors regard the set of items in a user profile as the source of the privacy risk (i.e., the correlation with gender), and their approach leads to a severe drop in recommendation accuracy. In contrast, in the work at hand, we regard the *conjunction* of items in the user profile as the source of the privacy risk, i.e., the correlation of the user's behavioral pattern with gender stereotypes. Additionally, we address the accuracy drop by applying our perturbation mechanism only to users whose behavioral patterns coincide with gender stereotypes.

2.2 Fairness Through Adversarial Training in Recommendation

In the context of RSs, protecting users' privacy often relates to concepts of user fairness [2, 4, 6, 39, 41]—a topic of lively interest in research and public communities [3, 7, 35]. A particular overlap of the two strands exists with so-called fairness through unawareness or fairness through blindness approaches, where "unfair" bias is mitigated by hiding the users' sensitive attributes in the model training process [34]. Thus, privacy and fairness can potentially be ensured if the users' data on protected/sensitive attributes is not encoded in the model.

In RS research, several works use adversarial learning as an in-processing technique [13] to generate feature-independent user embeddings. For instance, to achieve counterfactual fairness, Li et al. [18] apply an adversarial learning module to enforce user embeddings to be independent of the protected attributes. Ganhör et al. [9] and Vassøy et al. [33] add adversarial training to autoencoder-based RSs (e.g., [16]) to remove the implicit information of protected attributes from latent representations of users. Wu et al. [37] use adversarial learning to develop a RS based on two representations of the user: a representation that carries the biased information through sensitive attributes and a bias-free representation that only encodes user interests. Wu et al. [38] develop a graph-based adversarial learning module to increase the fairness of recommendations. More similar to our work, Weinsberg et al. [36] and Strucks et al. [32] use obfuscation to achieve privacy; Slokom et al. [31] show that obfuscation also impacts the fairness of recommendations, while Lin et al. [21] use obfuscation to debias gender from RSs. In contrast to prior works, the work at hand introduces the usage of the user’s attribute-specific stereotypicality of items for the probabilistic selection of the data to obfuscate.

3 Methodology

The core idea of the proposed **SBO** method is to reduce the *stereotypicality* of the users’ preferences by applying item obfuscation (imputation and/or removal) at the user level. For this purpose, we first define an item stereotypicality score (I_{Ster}) based on the item’s group inclination (*IGI*). The *IGI* value indicates how likely it is that a user of a given group consumes a certain item. Then, we use the I_{Ster} values to establish the user’s stereotypicality (U_{Ster}) from the interaction data, which enables us to determine each user’s degree of stereotypicality concerning the group to which the user belongs. For instance, a male user who predominantly listens to male-associated music tracks will obtain a high user stereotypicality score. U_{Ster} is then used to identify suitable candidates for obfuscation according to a given threshold. For each candidate user selected for obfuscation, we sample a number of items proportional to a fixed percentage of the number of items the user interacted with and apply obfuscation operations on the sample.

We formally present our method in the subsequent sections, focusing on obfuscating gender¹ information because it is a common target for attacks. Note that our method can be easily adapted for other protected attributes. We start by defining the different stereotypicality scores for users and items. Then, we formulate **SBO** with the supported sub-sampling and obfuscation strategies.

3.1 Item’s Group Inclination

We split the set of unique users $U = \{u_1, \dots, u_{|U|}\}$ in k groups $\{U_g\}_{g=1}^k$, where $U_g \subset U$ and $\bigcap_{g=1}^k U_g = \emptyset$, based on the $k \geq 2$ mutually exclusive values of

¹ In this work, we consider only two possible values of gender. However, we acknowledge that the assumption of binary gender is an over-simplification.

the categorical protected attribute p associated with each user. In this work, we split the original set of users by their associated gender. Therefore, we define two groups, U_m and U_f , corresponding to the male and female users, respectively.

Items present different degrees of association to different user groups. Therefore, for each element in the set of unique items $V = \{v_1, \dots, v_{|V|}\}$, we define the item inclination towards the user group U_g as the fraction between the number of users in U_g that interacted with item v , and the total number of users in U_g . Therefore, given the set of observed interactions $L_{\text{obs}} \subset U \times V$, $IGI(v, U_g)$ is given by:

$$IGI(v, U_g) = \frac{|\{u : (u, v) \in L_{\text{obs}}\}|}{|U_g|} \quad (1)$$

3.2 Item Stereotypicality

In order to determine if an item is a good candidate for obfuscation, we introduce the item stereotypicality (I_{Ster}), which relates the IGI values of the same item v (Eq. 1) across two user groups. The closer the values of inclination across groups, the closer to zero the value of I_{Ster} . This also indicates that the items closer to the extremes are the most stereotypical ones. The definition of I_{Ster} and its dependence on U_g and $U_{g'}$ is given by:

$$I_{\text{Ster}}(v, U_g, U_{g'}) = \frac{IGI(v, U_g) - IGI(v, U_{g'})}{\max\{IGI(v, U_g), IGI(v, U_{g'})\}} \quad (2)$$

Therefore, $I_{\text{Ster}}(v, U_g, U_{g'}) = -I_{\text{Ster}}(v, U_{g'}, U_g)$.

Figure 1 shows the distribution of I_{Ster} values over items for the LFM-2B-100K and ML-1M datasets, and for the users in the U_m group, i. e., setting $U_g = U_m$ and $U_{g'} = U_f$ in Eq. 1. Whenever considering a user, we gather the corresponding I_{Ster} values that match the value of the user-protected attribute. In addition, these values are calculated only for items that were consumed by at least one user in each user group.

3.3 User Group Stereotypicality

Next, we introduce a measure of the target user’s strength of preference towards group-biased or stereotypical items as defined in Subsect. 3.1. Given a user u and the items in their profile $v \in X_u$, the user’s preference towards stereotypical items is measured as the mean $U_{\text{Ster}}^{\text{mean}}$ or median $U_{\text{Ster}}^{\text{median}}$ of the distribution of values of I_{Ster}^u over the items in X_u . Throughout this paper, for simplicity, we refer to these scores as U_{Ster} for both definitions (mean and median), but separately explore the effects of both in our results.

The U_{Ster} values are used to determine whether a user is to be considered *highly stereotypical*. Therefore, we define the threshold of user stereotypicality γ as the mean value of all users’ U_{Ster} scores. Users with $U_{\text{Ster}} \geq \gamma$ are considered *highly stereotypical* and hence selected as targets for obfuscation. Figure 2 shows the values of U_{Ster} of users from LFM-2B-100K and ML-1M in order of descending stereotypicality, as well as the thresholds γ .

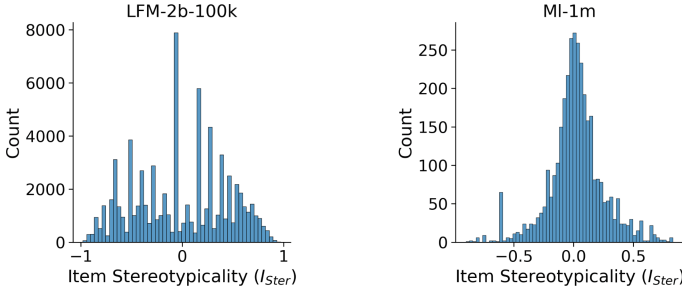


Fig. 1. Distribution of item stereotypicality $I_{\text{Ster}}(v, U_g, U_{g'})$ with $U_g = U_m$ and $U_{g'} = U_f$ over the items of the **LFM-2B-100k** (left) and **ML-1M** (right) datasets.

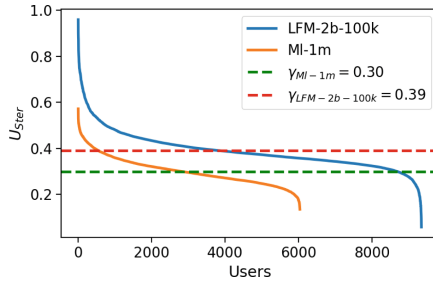


Fig. 2. User group stereotypicality of users from the **LFM-2B-100k** and **ML-1M** datasets, with users in order of descending stereotypicality. The red dotted and green dotted lines indicate the selection threshold $U_{\text{Ster}}^{\text{mean}}$ used for **LFM-2B-100k** and **ML-1M**, respectively. (Color figure online)

3.4 Stereotypicality-Based Obfuscation

Our method **SBO** consists of three main steps: 1) filtering users according to their U_{Ster} score; 2) sub-sampling candidate items; and 3) obfuscating the users' profiles. Below, we describe each step of the method separately and summarize them in Algorithm 1. First, we compute the list M_u of values of I_{Ster} according to the user's gender label g (each entry of M_u representing a different item). Then, we compute the U_{Ster} values for each user and filter the users with scores higher than the threshold γ , which is considered as a hyper-parameter. Given an obfuscation ratio ρ , the item sub-sampling consists of selecting a set of obfuscation candidates X_u^ρ for the user, containing at most $\rho \cdot |X_u|$ items. For this purpose, we define different sampling pools for the three different obfuscation strategies: *imputation*, *removal*, and *weighted*. Specifically, the sampling pool for *imputation* is $V - X_u$, and the sampling pool for *removal* is X_u ; additionally, a weighted combination of these two is the sampling pool for *weighted*. The weight $\omega \leq 1$ decides on the number of items to select for imputation ω and for removal $1 - \omega$ and is treated as a hyper-parameter.²

² We report results for $\omega = 0.5$ only.

Stereotypicality-Based Sampling. To sample the items to select for obfuscation, **SB0** first selects the items with the highest I_{Ster} scores from the set of obfuscation candidates X_u^ρ . Then, **SB0** decides on the items to obfuscate by performing a Bernoulli trial on each of the selected items, with a success rate equal to the item’s absolute I_{Ster} values. Therefore, items that have high I_{Ster} values are more likely to be obfuscated. The candidate items in X_u^ρ in which Bernoulli trials were successful are inserted in the obfuscation items list C_u , and then obfuscated. The Bernoulli trials are performed on each item independently to use the same I_{Ster} values across all user profiles.

Our aim is to obfuscate the items that are highly stereotypical in the user profile, therefore, when imputing unseen items, we choose items with the most negative I_{Ster} scores (most counter-stereotypical for u ’s gender). On the contrary, when removing items, we select the items with the most positive I_{Ster} scores (most stereotypical for u ’s gender). Following the same reasoning, we also define an additional baseline sampling strategy for comparison, *TopStereo*, where the items with the highest I_{Ster} scores in the user profile are selected for removal and the most negative for imputation. In addition, we include the *Random* strategy, which selects items uniformly at random from the user profile for removal and from the set of unexplored items for imputation. After having sub-sampled the list of candidate items X_u^ρ , we perform the selected obfuscation method using the **Obfuscate** on the user profile X_u and the obfuscation strategy m .

3.5 Attacker Network

As common in literature [9, 19], we use a simple feed-forward network as an attacker network. The network is trained on vector representations of the users’ interaction data in a supervised manner to predict the private attributes from these representations. The successful prediction of the attribute implies that the current interaction data can reveal the values of the attributes. In our case, this network takes the user preference vectors as input and aims to predict the user’s gender.

Algorithm 1: Stereotypicality-based Obfuscation

```

input : List of items the user  $u$  interacted with  $X_u$ ,
        User  $u$ 's gender label  $g$ ,
        List of unique items  $V$ ,
        User groups defined by gender  $\{U_m, U_f\}$ ,
        User stereotypicality threshold  $\gamma$ ,
        Obfuscation sampling ratio  $\rho$ ,
        Obfuscation strategy  $m$ 
output: Obfuscated list of user  $u$ 's interactions  $\tilde{X}_u$ 

// Assigning user's stereotypicality
 $S_u \leftarrow U_{\text{Ster}}(X_u)$ 
// User's obfuscation candidate items
 $C_u \leftarrow \{\}$ 
 $\tilde{X}_u \leftarrow \{\}$ 
// Defining the list of item stereotypicality values according to
// the user's gender label
if  $g == \text{male}$  then
  |  $M_u \leftarrow \{I_{\text{Ster}}(v, U_m, U_f) : v \in V\}$ 
else
  |  $M_u \leftarrow \{I_{\text{Ster}}(v, U_f, U_m) : v \in V\}$ 
end
// Evaluating the user for high stereotypicality
if  $S_u \geq \gamma$  then
  // Sub-sampling of candidate items to obfuscate
   $X_u^\rho \leftarrow \text{SubSample}(V, X_u, \rho, m)$ 
  for  $v \in X_u^\rho$  do
    |  $p \leftarrow |M_u(v)|$ 
    |  $c \leftarrow \text{BernoulliTrial}(p)$ 
    | if  $c == \text{True}$  then
    | |  $C_u \leftarrow C_u \cup \{v\}$ 
    | end
  end
  // Performing obfuscation of the user profile  $X_u$ 
   $\tilde{X}_u \leftarrow \text{Obfuscate}(X_u, C_u, m)$ 
else
  |  $\tilde{X}_u \leftarrow X_u$ 
end

```

4 Experimental Setup

4.1 Datasets

We run evaluation experiments on two popular datasets: ML-1M [10]³ and LFM-2B-100k,⁴ covering the movie and music domain, respectively (Table 1).

³ <https://grouplens.org/datasets/movielens/1m/>.

⁴ A subset of LFM-2B [23,30], derived by first selecting users with valid gender information, then randomly select $\sim 100k$ unique items that adhere to 5-core filtering.

For the training of recommendation models, we apply 5-core filtering to each dataset, randomly select 20% of each user’s interactions, and leave them out as *test* set. We apply the same split procedure on the remaining 80% of interactions to generate the *training* and *validation* sets. For the attackers’ training, we perform 5-fold cross-validation over the set of unique users, leaving 20% of them as test users in each fold, and report the average value of the evaluation metrics computed over the folds.

In order to perform obfuscation, we use the concatenation of the *train* and *validation* slices of the original datasets, then we slice the resultant set of interactions following the previously introduced procedure for both recommendation models and attacker networks.

Table 1. Statistical description of datasets

Dataset	Users (Male/Female)	Items	Interactions	Density
ML-1M	6,040 (4,331/1,709)	3,416	999,611	0.0484
LFM-2B-100K	9,364 (7,580/1,784)	99,965	1,820,903	0.0019

4.2 Dataset Obfuscation

The generation of obfuscated datasets is done before training the models with the following hyper-parameters: the user stereotypicality threshold γ is defined as the mean or median as described in Sect. 3.3, the obfuscation ratio parameter is set to $\rho = 0.1$.⁵ We perform experiments for all the obfuscation strategies and sampling methods defined in Sect. 3.4. We evaluate SBO against a state-of-the-art obfuscation approach, **Perblur**, proposed by Slokom et al. [31]. Where available, we used the code provided by the authors⁶ and implemented the missing pieces of code. Specifically, we set **Perblur**’s number of user neighbors to 50 for LFM-2B-100K and to 100 for ML-1M. From these neighbors, we collect the 200 and 500 most frequent recommended items for LFM-2B-100K and ML-1M, and used them as personalized lists. Then, we follow the procedure described by Slokom et al. [32] for selecting the 50 most indicative items for each gender. We include in our results both the performance of **Perblur** with the imputation and with the removal method.

4.3 Algorithms

Recommendation Models. Since the proposed method SBO is largely independent of the recommendation algorithm as long as those are trained on implicit feedback, we carry out our experiments on a selection of well-established recommendation algorithms from different categories: matrix factorization (BPR-MF [28]),

⁵ We also used $\rho = 0.05$, obtaining similar results, for which we refer the reader to our supplementary material (Appendix A).

⁶ <https://github.com/SlokomManel/PerBlur>.

neural network-based (MULTVAE [20]), and graph-based (LIGHTGCN [12]), hence demonstrating its performance across different types of RSs. We train the RSs for 100 epochs with a learning rate of 0.001 using the Adam optimizer with 512 as batch size. We apply early-stopping with a patience of 10 epochs, using NDCG as validation metric, computed for the top 10 predicted (i.e., recommended) items. The embedding size of all models is set to 64. We use the implementation of the RS models provided by the RecBole⁷ framework. Each model is evaluated with each of the dataset obfuscation parameters defined in Sect. 4.2.

Attacker Networks. For the attacker networks, we define the architecture $A = [|V|, l, 2]$ setting the number of nodes of the intermediate layer to $l = 128$ for ML-1M and to $l = 256$ for LFM-2B-100K. Each of the attackers is trained for 50 epochs using the Adam optimizer with 64 as batch size and 0.001 as learning rate with a Cross-Entropy (CE) minimization objective. In order to mitigate the imbalanced distribution of gender, we set proportional weights to each gender category in the CE objective. These networks are applied to all the configurations of parameters defined in Subject. 4.2.

Evaluation. To assess the recommendation performance, we report the Normalized Discounted Cumulative Gain (NDCG) for the top 10 recommended items. Additionally, we report the Balanced Accuracy (BAcc) to assess the performance of the attacker networks. To ensure the reproducibility of our research, the implementation and complete configuration of our experiments can be found in our publicly available repository.⁸

5 Results and Discussion

In this section, we describe our results, focusing first on the effect on the accuracy–privacy trade-off. We then delve into the effect of SBO’s different parameter configurations. Table 2 shows the user’s gender obfuscation capabilities of SBO in terms on BAcc for both datasets. Given that both SBO and the baseline Perblur are independent of the recommendation algorithm, the same values of BAcc are valid for the analysis of the performance of the different recommendation algorithms. We also report the results on the dataset without obfuscations, which we refer to as *original*. The BAcc values reported correspond to the best values of the average test results over 5-folds for each obfuscation parameter configuration, with the corresponding NDCG values for each recommendation algorithm, in which at most 10% of the user profiles were obfuscated ($\rho = 0.1$).

We observe that SBO in its variant with *removal* and *SBsampling*, consistently yields the best results in terms of BAcc for both datasets, proving SBO’s effectiveness in preventing the attacker’s ability to infer user’s protected attributes, at the cost of a slight decrease in NDCG. With *removal* and *SBsampling*, SBO delivers $\sim 7\%$ and $\sim 9\%$ of improvement in BAcc with respect

⁷ <https://github.com/RUCAIBox/RecBole>.

⁸ <https://github.com/hcai-mms/SBO>.

Table 2. Experimental results on the two datasets **ML-1M** and **LFM-2B-100K**. The scores in **bold** indicate the best scores across all models.

Dataset	Obf. Strat.	Sampling	BAcc↓	BPR-MF	LIGHTGCN	MULTVAE
				NDCG ↑		
LFM-2B-100K	original	original	0.5501	0.1135	0.1773	0.1483
	impute	PerBlur	0.5522	0.1042	0.1561	0.1402
		Random	0.5427	0.0990	0.1543	0.1607
		SBSampling	0.5528	0.1209	0.1764	0.1513
		TopStereo	0.5528	0.1209	0.1764	0.1513
	remove	PerBlur	0.5471	0.1155	0.1764	0.1507
		Random	0.5414	0.1070	0.1564	0.1324
		SBSampling	0.5136	0.1138	0.1731	0.1441
		TopStereo	0.5445	0.1224	0.1759	0.1518
	weighted	Random	0.5417	0.1055	0.1584	0.1504
		SBSampling	0.5528	0.1209	0.1764	0.1513
		TopStereo	0.5528	0.1209	0.1764	0.1513
ML-1M	original	original	0.6182	0.3445	0.3655	0.3650
	impute	PerBlur	0.6156	0.3344	0.3581	0.3580
		Random	0.5973	0.3389	0.3592	0.3718
		SBSampling	0.8329	0.2866	0.3174	0.3154
		TopStereo	0.8751	0.3111	0.3468	0.3499
	remove	PerBlur	0.6597	0.3437	0.3656	0.3657
		Random	0.6076	0.2904	0.3116	0.3161
		SBSampling	0.5664	0.3400	0.3608	0.3586
		TopStereo	0.6124	0.3396	0.3679	0.3650
	weighted	Random	0.6001	0.3155	0.3347	0.3441
		SBSampling	0.7255	0.3114	0.3421	0.3383
		TopStereo	0.7335	0.3243	0.3560	0.3578

to the original **LFM-2B-100K** and the original **ML-1M** dataset, respectively, at the cost of $\sim 2\%$ decrease in NDCG across all RSs. Furthermore, when compared with **Perblur**, $\sim 8\%$ and $\sim 6\%$ in improvement in BAcc is achieved on **LFM-2B-100K** and **ML-1M**, respectively, which translates into a decrease of $\sim 4\%$ in NDCG on **LFM-2B-100K**, and a $\sim 1\%$ decrease in NDCG on **ML-1M**.

We observe that when imputing items, **SBO** can have a negative impact on BAcc for most obfuscation configurations; this may be due to the size of the sampling pool. In this regard, **Perblur** shows more robustness, which might be attributed to filtering items using the user-based KNN recommendation algorithm. This emphasizes the substantial influence of the selection of obfuscation candidates for imputation of user preferences.

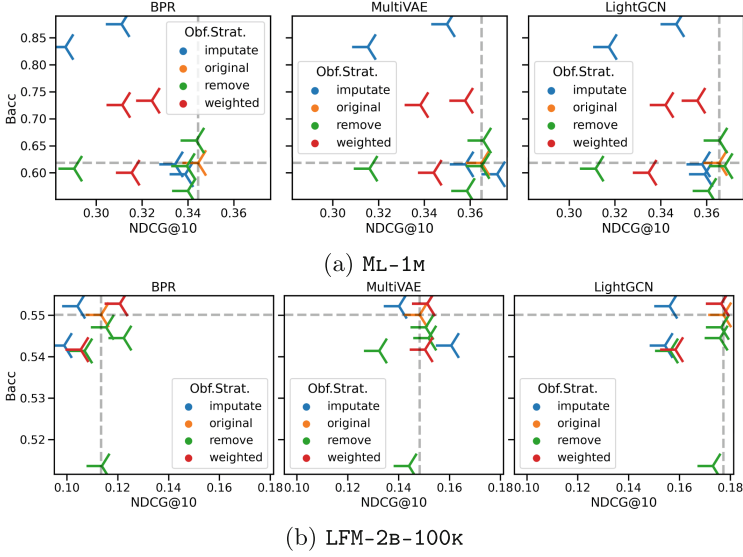


Fig. 3. Performance of the RSs and attacker (NDCG@10 and BAcc) with different obfuscation strategies on (a) $ML-1M$ and (b) $LFM-2B-100k$. The dotted lines indicate the performances on the datasets without any obfuscation in place.

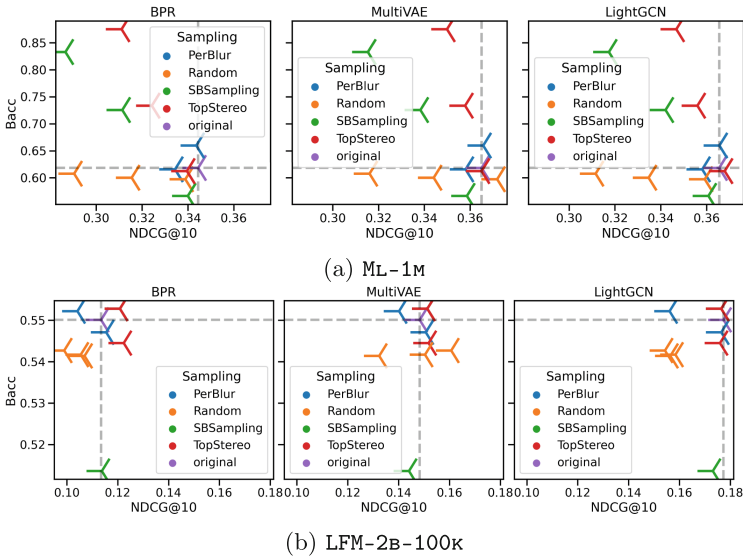


Fig. 4. Performance of the RSs and attacker (NDCG@10 and BAcc) with different sampling methods on (a) $ML-1M$ and (b) $LFM-2B-100k$. The dotted lines indicate the performances on the datasets without any obfuscation in place.

From Table 2, we can also speculate that on the original LFM-2B-100k it is already hard to infer the users’ gender attribute from their preferences, given the low BAcc values reported. In comparison, ML-1M is more exposed to adversarial attacks inferring users’ gender (higher BAcc on original dataset), and also more sensitive to the obfuscation methods applied, given the fluctuation in the values of BAcc when different obfuscation strategies are used.

Figure 3 and Fig. 4 show the results of the obfuscation strategy and sampling method obfuscation parameters from Table 2 in terms of two-dimensional plots with NDCG on the x -axis and BAcc on the y -axis, and for each recommendation algorithm. In each subplot, the points closer to the bottom-right corner provide better accuracy–privacy trade-off (higher NDCG and lower BAcc).

In Fig. 3, we see that for both datasets, *removal* is usually below the original dataset BAcc values (below the dotted line), indicating the effectiveness of *removal* in preventing adversarial attacks on protected attributes. Other points clearly show improvements in NDCG, although with a lesser impact on BAcc compared to *removal*. The effect of *removal* is larger on LFM-2B-100k. Furthermore, for the *weighted* strategy, we observe that the performance of SBO mostly falls in the central regions of the plots. Since varying $\omega \in [0, 1]$ allows adjusting the level of *imputation* and *removal*, we speculate that the parameters of *weighted* could be optimized to target better privacy-oriented results. Figure 4 compares the performance of SBO with different sampling methods. We observe that on ML-1M, *SBsampling* and *TopStereo* have decreasing behavior in terms of BAcc while increasing in NDCG values. On the other hand, *Perblur* has an ascending tendency. On the LFM-2B-100k dataset, the results are more diverse and only partially resemble the trends observed on ML-1M. More importantly, the behavior of *SBsampling* is similar across different recommendation algorithms.

6 Conclusion and Future Work

In this work, we introduced SBO, a novel probabilistic user preference obfuscation method that selects the items to obfuscate based on stereotypicality measures for users and items. Our experiments show that SBO can reach a better accuracy–privacy trade-off than the baselines used for comparison on two recommendation domains (music and movies) by removing highly stereotypical items from the users’ profiles. In addition, we show that the different configurations of SBO (obfuscation and sampling strategy) have similar behavior across different recommendation algorithms.

In this work, we limited the analysis to gender as the protected attribute and oversimplified its definition, reducing it to a binary attribute. Therefore, we plan to extend the current work by including an analysis of the effect of SBO with user attributes beyond binary categories, such as age groups or ethnicities. Additionally, our experiments hinted that *imputation* has the potential to achieve a better accuracy–privacy trade-off, a hypothesis that we leave for future work. Finally, further analyses can target the mitigation of other user privacy objectives, such as membership inference.

Acknowledgments. This research was funded in whole or in part by the FFG COMET center program, by the Austrian Science Fund (FWF): P36413, P33526, and DFH-23, and by the State of Upper Austria and the Federal Ministry of Education, Science, and Research, through grants LIT-2021-YOU-215 and LIT-2020-9-SEE-113.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Anelli, V.W., Deldjoo, Y., Di Noia, T., Ferrara, A., Narducci, F.: FedeRank: user controlled feedback with federated recommender systems. In: Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) ECIR 2021. LNCS, vol. 12656, pp. 32–47. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72113-8_3
2. Anelli, V.W., Deldjoo, Y., Noia, T.D., Merra, F.A.: Adversarial recommender systems: attack, defense, and advances. In: Ricci, F., Rokach, L., Shapira, B. (eds.) Recommender Systems Handbook, pp. 335–380. Springer, New York (2022). https://doi.org/10.1007/978-1-0716-2197-4_9
3. Deldjoo, Y., Jannach, D., Bellogin, A., Difonzo, A., Zanzonelli, D.: Fairness in recommender systems: research landscape and future directions. *User Model. User-Adapted Interact.* **34**(1) (2024)
4. Deldjoo, Y., Noia, T.D., Merra, F.A.: A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks. *ACM Comput. Surv.* **54**(2) (2021). <https://doi.org/10.1145/3439729>
5. Dwork, C.: Differential privacy: a survey of results. In: Agrawal, M., Du, D., Duan, Z., Li, A. (eds.) TAMC 2008. LNCS, vol. 4978, pp. 1–19. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79228-4_1
6. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS), pp. 214–226 (2012)
7. Ekstrand, M.D., Das, A., Burke, R., Diaz, F.: Fairness in recommender systems, pp. 603–646. Springer, New York (2022)
8. Ekstrand, M.D., Joshaghani, R., Mehrpouyan, H.: Privacy for all: ensuring fair and equitable privacy protections. In: Conference on Fairness, Accountability and Transparency, pp. 35–47. PMLR (2018)
9. Ganhör, C., Penz, D., Rekabsaz, N., Lesota, O., Schedl, M.: Unlearning protected user attributes in recommendations with adversarial training. In: Proceedings of the 45th International ACM SIGIR Conference, SIGIR 2022, pp. 2142–2147. ACM, New York (2022). <https://doi.org/10.1145/3477495.3531820>
10. Harper, F.M., Konstan, J.A.: The movielens datasets: history and context. *ACM Trans. Interact. Intell. Syst.* **5**(4), 19:1–19:19 (2016). <https://doi.org/10.1145/2827872>
11. Hashemi, H., et al.: Data leakage via access patterns of sparse features in deep learning-based recommendation systems. In: Workshop on Trustworthy and Socially Responsible Machine Learning (TSRML), in conjunction with the 36th Conference on Neural Information Processing Systems (NeurIPS) (2022)

12. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: simplifying and powering graph convolution network for recommendation. In: Huang, J.X., et al. (eds.) *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, 25–30 July 2020*, pp. 639–648. ACM (2020)
13. Jin, D., et al.: A survey on fairness-aware recommender systems. *Inf. Fusion* **100**, 101906 (2023)
14. Kim, S., Kim, J., Koo, D., Kim, Y., Yoon, H., Shin, J.: Efficient privacy-preserving matrix factorization via fully homomorphic encryption. In: *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIACCS)*, pp. 617–628 (2016)
15. Krismayer, T., Schedl, M., Knees, P., Rabiser, R.: Predicting user demographics from music listening information. *Multim. Tools Appl.* **78**(3), 2897–2920 (2019). <https://doi.org/10.1007/S11042-018-5980-Y>
16. Lacic, E., Reiter-Haas, M., Kowald, D., Reddy Daredddy, M., Cho, J., Lex, E.: Using autoencoders for session-based job recommendations. *User Model. User Adap. Inter.* **30**, 617–658 (2020)
17. Lex, E., Kowald, D., Schedl, M.: Modeling popularity and temporal drift of music genre preferences. *Trans. Int. Soc. Music Inf. Retrieval* **3**(1), 17–31 (2020)
18. Li, Y., Chen, H., Xu, S., Ge, Y., Zhang, Y.: Towards personalized fairness based on causal notion. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2021*, pp. 1054–1063. ACM, New York (2021)
19. Li, Y., et al.: Making users indistinguishable: attribute-wise unlearning in recommender systems. In: *Proceedings of the 31st ACM International Conference on Multimedia, MM 2023*, pp. 984–994. ACM, New York (2023)
20. Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T.: Variational autoencoders for collaborative filtering. In: *Proceedings of the 2018 World Wide Web Conference, WWW 2018*, pp. 689–698. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018)
21. Lin, C., Liu, B., Zhang, X., Wang, Z., Hu, C., Luo, L.: Privacy-preserving recommendation with debiased obfuscation. In: *IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2022, Wuhan, China, 9–11 December 2022*, pp. 590–597. IEEE (2022)
22. Lin, Y., et al.: Meta matrix factorization for federated rating predictions. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pp. 981–990. Springer, Cham (2020)
23. Melchiorre, A.B., Rekabsaz, N., Parada-Cabaleiro, E., Brandl, S., Lesota, O., Schedl, M.: Investigating gender fairness of recommendation algorithms in the music domain. *Inf. Process. Manag.* **58**(5), 102666 (2021)
24. Muellner, P., Kowald, D., Lex, E.: Robustness of meta matrix factorization against strict privacy constraints. In: *European Conference on Information Retrieval*, pp. 107–119 (2021)
25. Müllner, P., Lex, E., Schedl, M., Kowald, D.: ReuseKNN: neighborhood reuse for differentially-private KNN-based recommendations. *ACM Trans. Intell. Syst. Technol.* **14**(5), 1–29 (2023)
26. Müllner, P., Lex, E., Schedl, M., Kowald, D.: The impact of differential privacy on recommendation accuracy and popularity bias. In: Goharian, N., et al. (eds.) *ECIR 2024. LNCS*, vol. 14611, pp. 466–482. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-56066-8_33

27. Müllner, P., Lex, E., Schedl, M., Kowald, D.: Differential privacy in collaborative filtering recommender systems: a review. *Front. Big Data* **6** (2023)
28. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: *Proceedings of UAI*, pp. 452–461 (2009)
29. Schedl, M.: Investigating country-specific music preferences and music recommendation algorithms with the LFM-1B dataset. *Int. J. Multim. Inf. Retr.* **6**(1), 71–84 (2017)
30. Schedl, M., Brandl, S., Lesota, O., Parada-Cabaleiro, E., Penz, D., Rekabsaz, N.: LFM-2B: a dataset of enriched music listening events for recommender systems research and fairness analysis. In: *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval, CHIIR 2022*, pp. 337–341. ACM, New York (2022)
31. Slokom, M., Hanjalic, A., Larson, M.A.: Towards user-oriented privacy for recommender system data: a personalization-based approach to gender obfuscation for user profiles. *Inf. Process. Manag.* **58**(6), 102722 (2021)
32. Strucks, C., Slokom, M., Larson, M.A.: Blurm(or)e: revisiting gender obfuscation in the user-item matrix. In: Burke, R., Abdollahpouri, H., Malthouse, E.C., Thai, K.P., Zhang, Y. (eds.) *Proceedings of the Workshop on Recommendation in Multi-stakeholder Environments co-located with the 13th ACM Conference on Recommender Systems (RecSys 2019)*, Copenhagen, Denmark, 20 September 2019. *CEUR Workshop Proceedings*, vol. 2440. CEUR-WS.org (2019)
33. Vassøy, B., Langseth, H., Kille, B.: Providing previously unseen users fair recommendations using variational autoencoders. In: Zhang, J., et al. (eds.) *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023*, Singapore, Singapore, 18–22 September 2023, pp. 871–876. ACM (2023)
34. Verma, S., Rubin, J.: Fairness definitions explained. In: *Proceedings of the International Workshop on Software Fairness*, pp. 1–7 (2018)
35. Wang, Y., Ma, W., Zhang, M., Liu, Y., Ma, S.: A survey on the fairness of recommender systems. *ACM Trans. Inf. Syst.* **41**(3), 1–43 (2023)
36. Weinsberg, U., Bhagat, S., Ioannidis, S., Taft, N.: Blurme: inferring and obfuscating user gender based on ratings. In: *Proceedings of the Sixth ACM Conference on Recommender Systems*, pp. 195–202 (2012)
37. Wu, C., Wu, F., Wang, X., Huang, Y., , Xie, X.: Fairness-aware news recommendation with decomposed adversarial learning. In: *Proceedings of AAAI Conference on Artificial Intelligence*, pp. 4462–4469 (2021)
38. Wu, L., Chen, L., Shao, P., Hong, R., Wang, X., Wang, M.: Learning fair representations for recommendation: a graph-based perspective. In: *Proceedings of the Web Conference 2021, WWW 2021*, pp. 2198–2208. ACM, New York (2021)
39. Wu, Y., Cao, J., Xu, G.: Fairness in recommender systems: evaluation approaches and assurance strategies. *ACM Trans. Knowl. Discov. Data* **18**(1), 1–37 (2023)
40. Xin, X., et al.: On the user behavior leakage from recommender system exposure. *ACM Trans. Inf. Syst. (TOIS)* **41**(3), 1–25 (2023)
41. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: *International Conference on Machine Learning (ICML)*, pp. 325–333 (2013)

42. Zhang, M., Chen, Y., Lin, J.: A privacy-preserving optimization of neighborhood-based recommendation for medical-aided diagnosis and treatment. *IEEE Internet Things J.* **8**(13), 10830–10842 (2021)
43. Zhang, S., Yin, H.: Comprehensive privacy analysis on federated recommender system against attribute inference attacks. *IEEE Trans. Knowl. Data Eng. (TKDE)* (2023)