Revealing, Analyzing, and Mitigating the Adverse Impact of Shared Accounts on Personalized Recommendations for Single-User Accounts

Xingbo Jing
Taiyuan University of Technology
Jinzhong, Shanxi, China
jingxingbo1733@link.tyut.edu.cn

Tao Lian*
Taiyuan University of Technology
Jinzhong, Shanxi, China
liantao@tyut.edu.cn

Li Wang
Taiyuan University of Technology
Jinzhong, Shanxi, China
wangli@tyut.edu.cn

Abstract

Shared accounts are not uncommon in some applications, which can breed spurious associations between items that are actually consumed by distinct members. This may adversely affect the recommendations for other single-user accounts, which remains unexplored. To reveal this problem, we present a fair comparison of the recommendation performance for the same subset of single-user accounts under two settings: A controlled setting with only single-user accounts, and an experimental setting with both single-user accounts and shared accounts. The results show that the presence of shared accounts can lead to a performance decrease of up to 10%–40% for other single-user accounts. To gain deeper insights, we define two quantitative metrics to analyze the changes in the proximity of demoted relevant items and promoted irrelevant items to the user's consumed items in the latent space. The results show that the performance degradation can be attributed to the distortion of proximity relations between item embeddings. To mitigate the performance degradation, we leverage the multimodal information of items to construct a reliable item-item semantic graph which is fused with the bipartite behavior graph to counterbalance spurious second-order item-item associations arising from shared accounts. The results show that our mitigation method can effectively alleviate the performance degradation for other single-user accounts caused by shared accounts, recovering or even surpassing the recommendation performance in the controlled setting without shared accounts.

CCS Concepts

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

Keywords

Recommender System, Performance Analysis, Shared Account, Collaborative Filtering, Multimodal Recommendation

ACM Reference Format:
Xingbo Jing, Tao Lian, and Li Wang. 2025. Revealing, Analyzing, and Mitigating the Adverse Impact of Shared Accounts on Personalized Recommendations for Single-User Accounts. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author

Conference acronym 'XX, Woodstock, NY
2025. ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Collaborative filtering (CF) is a widely used approach to making personalized recommendations in many systems. It assumes that each account is used by a single user and that individuals who have exhibited similar preferences for certain items in the past are likely to share preferences for other items in the future. However, there exist many shared accounts in some applications. For example, multiple people use the same IPTV [2, 12] or share a VIP account in a web service. CF algorithms may fail in this circumstance. Fig. 1 shows a toy example. Let u be a single-user account owned by a single man who enjoys watching science fiction movies, and a denote a shared account consisting of the behavior of a father and a son, with the father favoring science fiction movies and the son preferring cartoons. Based on the fact that u and a have consumed several common items, and science fiction movies and cartoons co-occur in the consumption history of many shared accounts, CF algorithms may incorrectly recommend some animations to the single-user account u . Therefore, shared accounts may have an adverse impact on the recommendation performance for other single-user accounts in the system.

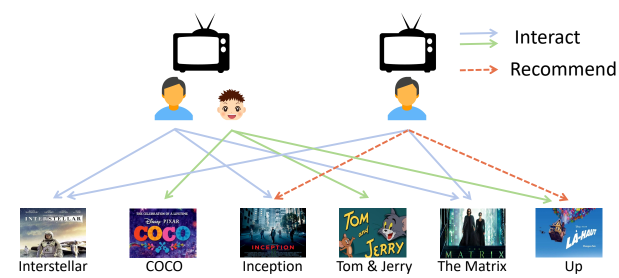


Figure 1: Illustration of the impact of a shared account on another single-user account

Although a few studies attempt to differentiate multiple members in a shared account using additional contextual information such as temporal factors [27, 32], the underlying assumption that the behavior of distinct members differs along a certain contextual dimension may not hold universally. Moreover, it is difficult to verify the accuracy of user identification, since which member invokes each interaction can hardly be recorded in many systems. Although different users can be identified via device IDs [20], mobile sensors [16], or other means such as face verification, it may not be applicable or widely adopted by users in some scenarios, for example, video streaming service on smart TV. Existing algorithmic studies [22, 24] only report the overall average recommendation performance, overlooking the coexistence and distinction of single-user accounts and shared accounts in a system. Hence, the adverse impact of shared accounts on other single-user accounts remains unexplored. To fill this gap, our research objective is to reveal, analyze, and mitigate the adverse impact of shared accounts on personalized recommendations for other single-user accounts in an environment with both types of accounts.

Revealing Phase. It is hard to quantitatively reveal the adverse impact of shared accounts on other single-user accounts, as we cannot know how a model performs for the same user in two counterfactual settings with and without shared accounts. We overcome this problem by simulating two settings with the same test set as shown in Fig. 2. The controlled setting has only single-user accounts, while the experimental setting includes both single-user accounts and shared accounts. Shared accounts are synthesized without altering the data sparsity. A fair comparison reveals that the recommendation performance for single-user accounts significantly declines with the presence of shared accounts.

Analysis Phase. We conjecture that shared accounts have a detrimental effect on item embeddings that are vital for recent model-based CF [17]. However, it remains unexplored how to analyze the changes of item embeddings between two settings. To this end, we identify the subset of demoted relevant items and promoted irrelevant items by comparing the recommendation lists for the same user under two settings. For every such item, we examine changes in its proximity to the user's consumed items in the latent space, measured by its relative positions in the K-nearest neighborhood. The analysis confirms that the performance degradation can be attributed to the distortion of proximity relations between item embeddings in the experimental setting with shared accounts.

Mitigating Phase. To mitigate performance degradation, the key is how to deal with spurious second-order associations between a pair of items consumed by two members with distinct interests in a shared account. Note that the edge between the shared account and either of them is not noise and hence cannot be removed. We circumvent this issue by leveraging the multimodal information of items that provides valuable clues about their semantic relations. Specifically, we construct a reliable item-item KNN graph from the multimodal data and integrate it with the bipartite behavior graph. During representation learning, these first-order semantic relations can counteract spurious second-order associations caused by shared accounts. Experimental results validate its efficacy in ameliorating the proximity relations between item embeddings and alleviating performance degradation for single-user accounts.

Our contributions are summarized as follows.

- We point out a problem that shared accounts can adversely affect personalized recommendations for other single-user accounts in a system.
- We design two settings with the same test set for a fair comparison: a controlled setting with only single-user accounts, and an experimental setting with both single-user accounts and shared accounts. Meanwhile, shared accounts are synthesized without changing the data sparsity.
- We define two quantitative metrics for analyzing the detrimental effect of shared accounts on the proximity relations between item embeddings.
- We design a mitigation method by constructing a reliable item-item KNN graph from the multimodal item information to counteract spurious second-order item-item associations mediated by shared accounts in the bipartite behavior graph.
- We observe a marked performance decline for single-user accounts in the setting with shared accounts. This can be attributed to the distortion of item proximity relations and mitigated by the item-item KNN graph derived from the multimodal item information.

2 Related work

2.1 Typical recommendation algorithms

Recommendation algorithms can roughly be categorized into collaborative filtering (CF) [21] and content-based filtering [19]. Recent CF methods include neural CF [8] and graph-based CF [1, 7, 13], which learn latent representations for users and items using only the user-item interaction matrix. A recent paradigm of content-based filtering is multimodal recommendation, which leverages the visual/textual content of items to enhance the item representations and user interest modeling. For example, VBPR [6] extends classical BPR [21] by incorporating the user's preference towards the item's visual features. [29] proposed to learn modality-specific user preference by feeding pre-extracted modality features of items into the initial embedding layer of a graph convolution network. In addition, noisy user-item edges can be pruned softly according to modality-specific user-item affinity scores [28]. [37] proposed an inter-modality feature alignment loss that aligns the item ID embedding with its multimodal features, encouraging the ID embeddings of items with similar multimodal features get close to each other. However, the above methods all assume that each account is used by a single user. We attempt to reveal the adverse impact caused by shared accounts and explore a viable mitigation method by exploiting the multimodal information of items.

2.2 Shared account recommendation

A common approach to personalized recommendations for shared accounts involves two phases: user identification and item recommendation. [9, 10, 27] split historical interactions into several subprofiles or clusters, assuming that different members exhibit distinct temporal habits and preferences. However, the number of members in a shared account is unknown, and the assumption may not hold universally. Verstrepen and Goethals [24] proposed an item-based CF solution for shared accounts in the absence of contextual information. Several recent works [3, 4, 11, 30] have explored session-based or sequential recommendation in scenarios with shared accounts by learning multiple latent role/persona representations. However, these studies overlook the co-existence of shared accounts and single-user accounts, leaving the impact of shared accounts on single-user accounts remain unexplored.

2.3 Recommendation performance degradation

Several factors can lead to performance degradation in recommender systems, such as data sparsity, noisy feedback, incoherent preference [5, 12, 36], and malicious attacks [25, 33, 34]. In this study, we show that shared accounts can adversely affect the recommendation performance for other single-user accounts by introducing spurious item-item associations. Unlike noisy feedback [23, 26] or fake co-visitation attack [14, 31], these spurious associations arise naturally from true historical interactions in shared accounts with no malicious intent. Understanding how CF algorithms perform in environments with both single-user accounts and shared accounts is crucial for improving the robustness [35] and fairness of recommender systems.

3 Revealing the adverse impact on recommendation performance

3.1 Experiment design

3.1.1 Controlled setting vs. experimental setting. To reveal the impact of shared accounts on the recommendation performance for other single-user accounts, we design two settings for a fair comparison, as shown in Fig. 2. In the *controlled setting*, all accounts in $\mathcal{U}^1 \cup \mathcal{U}^2$ are single-user accounts. In the *experimental setting*, there exist both single-user accounts and shared accounts. The interaction data for each account in $\mathcal{U}^1$ remains unchanged, while the accounts in $\mathcal{U}^2$ are used to synthesize shared accounts in $\mathcal{A}$, where each account contains a mixing of the interaction data of two users.

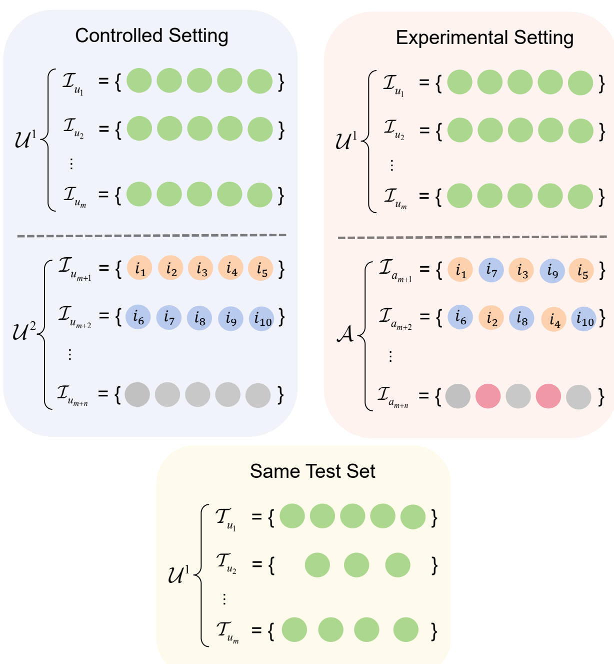


Figure 2: Experiment design: controlled setting vs. experimental setting

The test set is the same in both settings. For each $u \in \mathcal{U}^1$, 20% of consumed items are withheld as the test set $\mathcal{T}_u$, another 20% for validation, and the rest for training. The same recommendation model is trained, which may produce different top-N recommendations for the same user $u \in \mathcal{U}^1$, denoted as $\mathcal{L}_u^S$ and $\mathcal{L}_u^A$ in the controlled setting and experimental setting, respectively. By evaluating the performance of $\mathcal{L}_u^S$ and $\mathcal{L}_u^A$ according to $\mathcal{T}_u$, we can reveal the impact of shared accounts in $\mathcal{A}$ on single-user accounts in $\mathcal{U}^1$.

3.1.2 Synthesizing shared accounts. In real-world applications, there exist some common membership patterns in shared accounts, such as couples, elderly individuals and children, etc. When there exist a lot of shared accounts with the same membership pattern, there might emerge strong spurious associations between a few items. In addition, some weak noisy item-item associations might also emerge among several items.

Definition 3.1. A fake co-occurrence refers to the fact that although two items co-occur in a shared account, they are actually consumed by distinct members using the shared account.

Definition 3.2. Strong spurious item-item association means that the two items are dissimilar but there exist many fake co-occurrences of them in many shared accounts.

Definition 3.3. Weak noisy item-item association means that there are occasional fake co-occurrences of the two items in a few shared accounts.

Existing work [24] adopted a simple method for synthesizing shared accounts, which merges the interaction history of two randomly selected users into an account. This method has two limitations. (1) It fails to well simulate strong spurious item-item associations introduced by shared accounts, due to insufficient fake co-occurrences between a given pair of dissimilar items. (2) It changes the data sparsity since the number of accounts is reduced but the number of observed interactions remains the same. Note that the data sparsity has a great influence on recommendation performance, which may interfere with the impact caused by shared accounts.

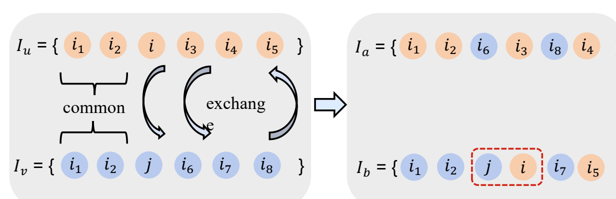


Figure 3: Illustration of the procedure for synthesizing shared accounts such that there emerges a fake co-occurrence of items i and j in one of the synthetic shared accounts and the data sparsity keeps the same

We design a novel procedure for synthesizing shared accounts, which is detailed in Appendix A. To simulate strong spurious associations between a pair of dissimilar items i and j, we first identify the subset of users in $\mathcal{U}^2$ who have consumed either i or j but not both, denoted as $\mathcal{U}_i'$ and $\mathcal{U}_j'$ respectively. Then for a pair of users $u \in \mathcal{U}_i'$ and $v \in \mathcal{U}_j'$, we swap a portion of their interacted items to form two synthetic shared accounts a and b such that there emerges a fake co-occurrence of i and j in one of them (e.g.,

b), as shown in Fig. 3. The spurious association between i and j becomes stronger after repeating the mixing operation for more user pairs in $\mathcal{U}'_i \times \mathcal{U}'_j$. The data sparsity keeps the same as in the controlled setting. In this way, we can exclude the interference of data sparsity and draw convincing results regarding the impact of shared accounts on other single-user accounts.

3.1.3 The recommendation model. We adopt LightGCN [7] as the recommendation model in two different settings. The reason is twofold: It is a strong competitor among various CF methods; it can model high-order item-item associations. In our experiments, the number of graph convolution layers was set to 2 such that it can model second-order item-item associations in shared accounts. The embedding dimension was set to 64, the learning rate was set to 0.001, and NDCG@20 was used as the early stop metric.

3.2 Quantitative comparison results

We conducted experiments on three public datasets: Clothing Shoes & Jewelry (Clothing for short), Beauty, and MicroLens. The Clothing and Beauty datasets[1] contain user shopping records, reviews, and product metadata. The multimodal features are also provided [15]: 4096-dimensional visual features, and 1024-dimensional textual features extracted by a sentence transformer. The MicroLens dataset[2] contains user interactions with micro-videos, user comments and video images. We use the multimodal features pre-processed by Ni et al. [18]. Table 1 shows the dataset statistics after 5-core filtering.

Table 1: Dataset statistics

Dataset	Clothing	Beauty	MicroLens
# of users	39387	22363	5936
# of items	23033	12101	12414
# of interactions	278677	198502	123368
Sparsity	99.97%	99.92%	99.83%

Table 2 shows a comparison of the recommendation performance for the same single-user accounts in $\mathcal{U}^1$ under the controlled setting vs. the experimental setting. The number of item pairs with strong spurious association p was set to 50, the proportion of shared accounts ρ and the mixing ratio ε were set to 0.5 (cf. Alg. 1 in Appendix A). We can observe a significant performance decrease in the experimental setting across three datasets, indicating that the presence of shared accounts has a considerable adverse impact on the recommendation performance for other single-user accounts.

4 Analyzing the adverse impact on item embeddings

For each user $u \in \mathcal{U}^1$, let $\pi\left(\mathcal{L}_u^S, i\right)$ and $\pi\left(\mathcal{L}_u^A, i\right)$ be the rank positions of item i in the recommendation lists in the controlled setting and experimental setting respectively. We can identify the following two types of items that cause the performance decline.

[1] https://github.com/sisinflab/Formal-MultiMod-Rec/tree/main/data
[2] https://recsys.westlake.edu.cn/MicroLens-Fairness-Dataset/

Table 2: Comparison of the recommendation performance for single-user accounts in the controlled setting vs. in the experimental setting

Dataset	Metric	Control	Experiment	Decrease↓
Clothing	Recall@10	0.0269	0.0168	37.55%
	Recall@20	0.0403	0.0240	40.45%
	NDCG@10	0.0168	0.0108	35.71%
	NDCG@20	0.0208	0.0130	37.50%
Beauty	Recall@10	0.0795	0.0640	19.50%
	Recall@20	0.1164	0.0927	20.36%
	NDCG@10	0.0508	0.0416	18.11%
	NDCG@20	0.0623	0.0505	18.94%
MircoLens	Recall@10	0.0371	0.0319	14.02%
	Recall@20	0.0632	0.0560	11.39%
	NDCG@10	0.0340	0.0285	16.18%
	NDCG@20	0.0447	0.0382	14.54%

Definition 4.1. The demoted relevant items $\mathcal{T}_u^{\downarrow}$ are defined as the subset of relevant items in $\mathcal{L}_u^S$ that is ranked at a later position in $\mathcal{L}_u^A$ compared to its position in $\mathcal{L}_u^S$.

$$\mathcal{T}_u^{\downarrow} = \left\{ i \in \mathcal{L}_u^S \mid i \in \mathcal{T}_u \wedge \pi\left(\mathcal{L}_u^S, i\right) \prec \pi\left(\mathcal{L}_u^A, i\right) \right\}.$$

Definition 4.2. The promoted irrelevant items $\mathcal{L}_u^{\uparrow}$ are defined as the subset of irrelevant items in $\mathcal{L}_u^A$ that is ranked at an earlier position compared to its position in $\mathcal{L}_u^S$.

$$\mathcal{L}_u^{\uparrow} = \left\{ i \in \mathcal{L}_u^A \mid i \notin \mathcal{T}_u \wedge \pi\left(\mathcal{L}_u^A, i\right) \prec \pi\left(\mathcal{L}_u^S, i\right) \right\}.$$

4.1 K-nearest neighbors in the latent space

To explore the reason for performance decline, we focus on the demoted relevant items $\mathcal{T}_u^{\downarrow}$ and promoted irrelevant items $\mathcal{L}_u^{\uparrow}$, and analyze their proximity to the set of items $\mathcal{I}_u$ consumed by u in the latent space. This could shed light on how the performance decline is related to changes in the item embeddings.

Given an item $j \in \mathcal{I}_u$, let $\mathcal{N}_j^S$ and $\mathcal{N}_j^A$ denote the K nearest neighbors of the item embedding $\mathbf{e}_j$ in the controlled setting and experimental setting respectively. We define $\pi\left(\mathcal{N}_j^S, i\right)$ as the rank position of item i in $\mathcal{N}_j^S$ according to its dot product with $\mathbf{e}_j$.

$$\pi\left(\mathcal{N}_j^S, i\right) = \begin{cases} \text{rank}\left(\mathcal{N}_j^S, i\right), & i \in \mathcal{N}_j^S; \\ +\infty, & i \notin \mathcal{N}_j^S. \end{cases} \tag{1}$$

Similarly, let $\pi\left(\mathcal{N}_j^A, i\right)$ denote the rank of item i in $\mathcal{N}_j^A$. By comparing $\pi\left(\mathcal{N}_j^S, i\right)$ and $\pi\left(\mathcal{N}_j^A, i\right)$, we define the following two quantities:

$$\text{Push}\left(i, \mathcal{I}_u\right) = \left|\left\{ j \in \mathcal{I}_u \mid \pi\left(\mathcal{N}_j^S, i\right) \prec \pi\left(\mathcal{N}_j^A, i\right) \right\}\right|; \tag{2}$$

$$\text{Pull}\left(i, \mathcal{I}_u\right) = \left|\left\{ j \in \mathcal{I}_u \mid \pi\left(\mathcal{N}_j^S, i\right) \succ \pi\left(\mathcal{N}_j^A, i\right) \right\}\right|. \tag{3}$$

Intuitively, Push $(i, \mathcal{I}_u)$ counts the number of times that item i is pushed away from the K-nearest neighborhood of any item $j \in \mathcal{I}_u$ in the experimental setting; Pull $(i, \mathcal{I}_u)$ is equal to the number of times that item i is pulled closer to the K-nearest neighborhood of any item $j \in \mathcal{I}_u$. Note that if $i \notin \mathcal{N}_j^S$ and $i \notin \mathcal{N}_j^A$, $\pi\left(\mathcal{N}_j^S, i\right) = \pi\left(\mathcal{N}_j^A, i\right) = +\infty$. In these cases, we do not take j into account when calculating Equations 2 and 3.

Then we define the following two ratios by comparing $\text{Push}\,(i, \mathcal{I}_u)$ and $\text{Pull}\,(i, \mathcal{I}_u)$.

$$\text{ratio\_push} = \frac{\sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{T}_u^{\downarrow}} \mathbb{I}\left(\text{Push}\,(i,\mathcal{I}_u) > \text{Pull}\,(i,\mathcal{I}_u)\right)}{\sum_{u\in\mathcal{U}} \left|\mathcal{T}_u^{\downarrow}\right|}; \quad (4)$$

$$\text{ratio\_pull} = \frac{\sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{L}_u^{\uparrow}} \mathbb{I}\left(\text{Push}\,(i,\mathcal{I}_u) < \text{Pull}\,(i,\mathcal{I}_u)\right)}{\sum_{u\in\mathcal{U}} \left|\mathcal{L}_u^{\uparrow}\right|}. \quad (5)$$

For a demoted relevant item $i \in \mathcal{T}_u^{\downarrow}$, if $\text{Push}\,(i, \mathcal{I}_u) > \text{Pull}\,(i, \mathcal{I}_u)$, it is pushed away from the K-nearest neighborhood of more items in $\mathcal{I}_u$; and reversely, for a promoted irrelevant item $i \in \mathcal{L}_u^{\uparrow}$, if $\text{Push}\,(i, \mathcal{I}_u) < \text{Pull}\,(i, \mathcal{I}_u)$, it is pulled closer to the K-nearest neighborhood of more items in $\mathcal{I}_u$ in the experimental setting.

4.2 KNN analysis results

We calculated the ratios defined in Equations 4 and 5 on three datasets. As shown in Table 3, the ratios are much higher than 0.5. These results indicate that the majority of demoted relevant items are pushed away from the K-nearest neighborhood of more items consumed by the user, and a large proportion of promoted irrelevant items are pulled closer to the K-nearest neighborhood of more items consumed by the user in the experimental setting. In other words, the proximity relations between item embeddings are greatly distorted due to the existence of shared accounts.

Table 3: KNN analysis of item embeddings

Dataset	K	ratio_push	ratio_pull
Clothing	10	0.8665	0.8648
	20	0.8660	0.8578
	50	0.8861	0.8455
Beauty	10	0.8290	0.8375
	20	0.8306	0.8310
	50	0.8357	0.8208
MicroLens	10	0.7891	0.7543
	20	0.8410	0.7676
	50	0.9014	0.7910

5 Mitigating the adverse impact by leveraging the multimodal information

The distortion of proximity relations between item embeddings can be attributed to spurious or noisy second-order item-item associations introduced by shared accounts. According to the neighborhood aggregation function of LightGCN, the second-order association strength for an item pair can be calculated as $c_{i,j} = \frac{1}{\sqrt{|\mathcal{U}_i|}\sqrt{|\mathcal{U}_j|}} \sum_{u\in\mathcal{U}_i\cap\mathcal{U}_j} \frac{1}{|\mathcal{I}_u|}$. Due to fake item-item co-occurrences in shared accounts, the set of accounts who have consumed both items i and j vary across the controlled setting and the experimental setting. To alleviate the detrimental effect on representation learning, we resort to the multimodal information of items which provides value clues about their semantic relations and is not contaminated by fake item-item co-occurrences in shared accounts.

5.1 Our mitigation method

Fig. 4 shows the workflow of our mitigation method. The core is to construct a reliable item-item KNN graph from the multimodal item data that can counteract spurious second-order item-item associations in the bipartite behavior graph caused by shared accounts.

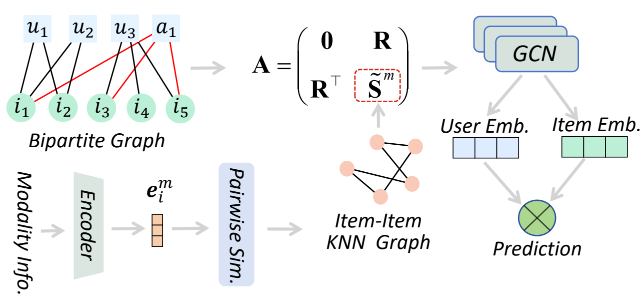


Figure 4: The workflow of our mitigation method

Firstly, we utilize a pre-trained text/image encoder to obtain the multimodal features for each item, denoted as $\mathbf{e}_i^m$. Then we can construct an item-item semantic graph, where the edge weight between items i and j is defined by the semantic similarity between their textual/visual features:

$$S_{i,j}^m = \frac{\mathbf{e}_i^{m\top}\mathbf{e}_j^m}{\|\mathbf{e}_i^m\|\|\mathbf{e}_j^m\|}. \quad (6)$$

Considering that not all semantic relations between items are useful for recommendation, especially those pairs with small semantic similarity, for each item i, we only preserve the edges with its K most similar items.

$$\tilde{S}_{i,j}^m = \begin{cases} 1, & S_{i,j}^m \in \text{top-}K\left(\mathbf{S}_{i,:}^m\right), \\ 0, & \text{otherwise}. \end{cases} \quad (7)$$

K is empirically set to 10 in our experiment. In this way, we can derive a reliable and sparse item-item semantic graph from their multimodal information.

Then we integrate the item-item KNN graph with the bipartite behavior graph, yielding a fused graph that adds a few reliable semantic edges on the item side. The corresponding adjacency matrix can be formulated as

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^\top & \tilde{\mathbf{S}}^m \end{pmatrix}. \quad (8)$$

$\mathbf{R} \in \{0,1\}^{|\mathcal{U}^1\cup\mathcal{A}|\times|\mathcal{I}|}$ indicates whether account u has interacted with item i. $\tilde{\mathbf{S}}^m \in \{0,1\}^{|\mathcal{I}|\times|\mathcal{I}|}$ encodes the reliable semantic relations between items derived from the multimodal item information. The upper-left block remains zero, as there are no explicit social relations between the accounts in some systems and the preference similarities involving shared accounts are noisy.

Finally, we stack multiple light graph convolution layers on the fused graph. In each layer, the representation for each item not only depends on the representations for accounts that have interacted with the item, but also directly aggregates the representations of first-order semantically similar items which can counterbalance the detrimental effect of second-order spurious item-item associations. The final embeddings $\mathbf{E} \in \mathbb{R}^{(|\mathcal{U}^1 \cup \mathcal{A}|+|\mathcal{I}|)\times d}$ are obtained by averaging across all layers and used to calculate the predicted scores by dot product.

5.2 Alleviating the performance degradation

To validate the effectiveness of our mitigation method, we trained two new LightGCN models by incorporating an item-item KNN graph constructed from either image features or text features in the experimental setting with shared accounts. Table 4 reports the recommendation performance for single-user accounts in $\mathcal{U}^1$.

We can draw the following conclusions. (1) Our mitigation method effectively alleviates the performance degradation for single-user accounts in the experimental setting, using either image (i.e., Ours-V) or text features (i.e., Ours-T). This can be attributed to the reliable item-item semantic relations from the image/text modality, which can counteract spurious second-order item-item associations during graph convolution. (2) Leveraging text features yields better results than using image features on three datasets. Perhaps item images also contain noisy visual features, while text features better capture the item semantic relations. (3) Our mitigation method also outperforms other multimodal recommendation models in the experimental setting, including VBPR [6], MMGCN [29], GRCN [28] and BM3 [37]. Although they also consider the multimodal item information, they mainly model the user-item bipartite graph, and hence are not robust to spurious second-order item-item associations incurred by shared accounts. In contrast, our method explicitly integrates an item-item graph derived from textual/visual features, which is conductive to counterbalancing those spurious associations. (4) On the Clothing dataset, our method achieves larger improvement, even surpassing performance in the controlled setting. This is likely because the most sparse Clothing dataset benefits more from the reliable item-item semantic relations derived from the multimodal item information.

Fixing the rank changes. We also conducted further analysis to see whether our mitigation method can fix the rank changes of demoted relevant items $\mathcal{T}_u^{\downarrow}$ and promoted irrelevant items $\mathcal{L}_u^{\uparrow}$ in the experimental setting. Let $\mathcal{L}_u^A$ denote the recommendation list for user u in the experimental setting as before, and $\mathcal{L}_u^M$ be the recommendation list for user u after exploiting our mitigation method by using the text features of items. We calculated the following two metrics:

$$\text{fix\_demot} = \frac{\sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{T}_u^{\downarrow}} \frac{1}{\log_2(\pi(\mathcal{L}_u^M,i)+1)} - \frac{1}{\log_2(\pi(\mathcal{L}_u^A,i)+1)}}{\sum_{u\in\mathcal{U}} \left|\mathcal{T}_u^{\downarrow}\right|}; \tag{9}$$

$$\text{fix\_promo} = \frac{\sum_{u\in\mathcal{U}} \sum_{i\in\mathcal{L}_u^{\uparrow}} \frac{1}{\log_2(\pi(\mathcal{L}_u^M,i)+1)} - \frac{1}{\log_2(\pi(\mathcal{L}_u^A,i)+1)}}{\sum_{u\in\mathcal{U}} \left|\mathcal{L}_u^{\uparrow}\right|}. \tag{10}$$

The results are reported in Table 5. For demoted relevant items $\mathcal{T}_u^{\downarrow}$, the values of fix_demot are positive, indicating that they are moved forward in the recommendation list by our mitigation method. For promoted irrelevant items $\mathcal{L}_u^{\uparrow}$, the values of fix_promo are negative, indicating that they are moved backward in the recommendation list. Thus, our mitigation method can counteract the adverse impact of shared accounts on recommendation lists for other single-user accounts.

5.3 Ameliorating the item embeddings

To validate the benefit of our mitigation method for representation learning, we analyzed the K-nearest neighbors of item embeddings after exploiting our mitigation method, similar to the analysis in Section 4.1. Let $\mathcal{N}_j^M$ denote the K-nearest neighbors of the item embedding $\mathbf{e}_j$ after incorporating the item-item semantic graph into the LightGCN model in the experimental setting, and $\mathcal{N}_j^S$ be the K-nearest neighbors of $\mathbf{e}_j$ learned by LightGCN in the controlled setting as in Section 4.1. We modify Equations 2 and 3 by replacing $\pi\left(\mathcal{N}_j^A, i\right)$ with $\pi\left(\mathcal{N}_j^M, i\right)$ as follows:

$$\text{Push}\,(i, \mathcal{I}_u) = \left|\left\{j \in \mathcal{I}_u \mid \pi\left(\mathcal{N}_j^S, i\right) \prec \pi\left(\mathcal{N}_j^M, i\right)\right\}\right|; \tag{11}$$

$$\text{Pull}\,(i, \mathcal{I}_u) = \left|\left\{j \in \mathcal{I}_u \mid \pi\left(\mathcal{N}_j^S, i\right) \succ \pi\left(\mathcal{N}_j^M, i\right)\right\}\right|. \tag{12}$$

Then we recalculated ratio_push and ratio_pull defined in Equations 4 and 5 based on the new definitions of Push $(i, \mathcal{I}_u)$ and Pull $(i, \mathcal{I}_u)$.

The results in Table 6 show that our mitigation method reduces ratio_push and ratio_pull significantly compared to the original model. Thus, the item-item semantic graph can partly remedy the detrimental effect on proximity relations between item embeddings caused by shared accounts. The decrease in ratio_push is smaller than that in ratio_pull, suggesting that repairing relations with demoted relevant items is harder than dissolving relations with promoted irrelevant items due to the scarcity of relevant items.

Counteracting strong spurious associations. Finally, we show the effectiveness of our mitigation method in counterbalancing strong spurious item-item associations. Given a pair of items i and j with strong spurious association, let s_1, s_2 and s_3 denote the cosine similarities of their embeddings in the controlled setting, the experimental setting before and after applying our mitigation method, respectively. We calculated $s_2 - s_1$ and $s_3 - s_2$ for 50 item pairs, and visualized their distributions in Fig. 5. The positive values of $s_2 - s_1$ indicate that item pairs with strong spurious associations have higher cosine similarities in the experimental setting than in the controlled setting. Our mitigation method significantly reduces these similarities, as shown by the negative values of $s_3 - s_2$. Thus, the reliable item-item semantic graph derived from the multimodal information can effectively counteract the adverse impact of spurious second-order item-item associations caused by shared accounts during graph convolution.

Table 4: The recommendation performance for single-user accounts when leveraging the image/text features of items in the experimental setting

Dataset	Metric	Controlled	Experimental						
		LightGCN	LightGCN	VBPR	MMGCN	GRCN	BM3	Ours-V	Ours-T
Clothing	Recall@10	0.0269	0.0168	0.0094	0.0083	0.0248	<u>0.0275</u>	0.0253	**0.0339**
	Recall@20	0.0403	0.0240	0.0142	0.0153	0.0387	<u>0.0408</u>	0.0381	**0.0515**
	NDCG@10	0.0168	0.0108	0.0059	0.0045	0.0148	<u>0.0175</u>	0.0161	**0.0212**
	NDCG@20	0.0208	0.0130	0.0073	0.0065	0.0188	<u>0.0214</u>	0.0198	**0.0264**
Beauty	Recall@10	0.0795	0.0640	0.0389	0.0394	0.0628	0.0699	<u>0.0750</u>	**0.0765**
	Recall@20	0.1164	0.0927	0.0578	0.0632	0.0950	0.1070	<u>0.1084</u>	**0.1108**
	NDCG@10	0.0508	0.0416	0.0245	0.0236	0.0394	0.0419	<u>0.0484</u>	**0.0493**
	NDCG@20	0.0623	0.0505	0.0304	0.0307	0.0491	0.0526	<u>0.0587</u>	**0.0600**
MicroLens	Recall@10	0.0371	0.0319	0.0263	0.0207	0.0330	0.0259	<u>0.0343</u>	**0.0365**
	Recall@20	0.0632	0.0560	0.0462	0.0400	0.0577	0.0473	<u>0.0581</u>	**0.0616**
	NDCG@10	0.0340	0.0285	0.0238	0.0194	0.0288	0.0236	<u>0.0310</u>	**0.0331**
	NDCG@20	0.0447	0.0382	0.0319	0.0273	0.0389	0.0317	<u>0.0407</u>	**0.0431**

Table 5: Fixing rank changes of demoted relevant items and promoted irrelevant items

Dataset	Clothing	Beauty	MicroLens
fix_demot	0.0559	0.0601	0.0538
fix_promo	-0.2678	-0.2546	-0.2125

Table 6: Analysis of K-nearest neighbors of item embeddings with $K = 20$: the original LightGCN vs. our mitigation method in the experimental setting

Dataset	Clothing		Beauty		MicroLens	
	Ori.	Our	Ori.	Our	Ori.	Our
ratio_push	0.8660	0.7579	0.8306	0.7426	0.8410	0.8175
ratio_pull	0.8578	0.2604	0.8310	0.3960	0.7676	0.3529

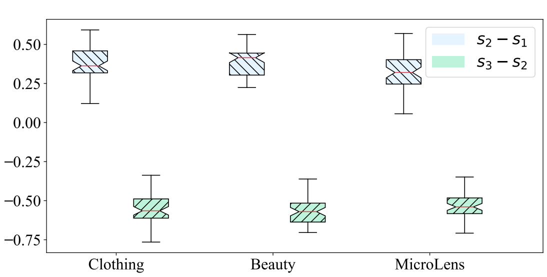


Figure 5: Boxplots of the similarity difference for item pairs with strong spurious association: s_1 is in the controlled setting; s_2 and s_3 are in the experimental setting before and after using our mitigation method

6 Conclusion and discussion

In this article, we carried out a systematic study to reveal, analyze, and mitigate the adverse impact of shared accounts on the recommendation performance for other single-user accounts. To reveal the problem, we conducted a fair comparison of the recommendation performance for the same user under two settings with the same test set: the controlled setting has only single-user accounts; the experimental setting has both single-user accounts and shared accounts. We observed a marked performance decline for other single-user accounts in the experimental setting. To explore the underlying reasons, we focused on demoted relevant items and promoted irrelevant items in the recommendation lists, and compared its proximity to the user's consumed items in the embedding space between two settings. The analyses show that the majority of demoted relevant items (resp. promoted irrelevant items) are pushed away from (resp. pulled closer to) the K-nearest neighborhood of the user's consumed items in the experimental setting. To alleviate the adverse impact, we presented a mitigation method that constructs a reliable item-item KNN graph from the multimodal information and integrates it with the bipartite behavior graph for enhanced representation learning. Experiment results demonstrate that our mitigation method can counteract the detrimental effect of spurious second-order item-item associations, ameliorate the proximity relations between item embeddings, and hence effectively alleviate the performance degradation for single-user accounts caused by shared accounts.

At last, we discuss the limitations of our study, and envision promising future work. The synthetic shared accounts may not perfectly reflect the behavior of shared accounts in real-world systems. For instance, we do not consider that some shared accounts may be used by more than two members. The effectiveness of our mitigation method depends on the quality of the multimodal information of items. If the information in a certain modality is very noisy or of little importance in influencing the user's choice, the effect may be limited or even negative. In the future, it is worth designing more delicate methods for synthesizing shared accounts by considering the patterns of user behavior and membership composition widely observed in real-world systems. In addition, we will explore more advanced multimodal recommendation methods that can leverage the valuable information from multiple modalities and meanwhile avoid the potential negative effect of fake co-occurrences in the collaborative signal as well as the noise in the multimodal information. Last but not least, our methodology for analyzing the distortion of proximity relations between item embeddings can be extended to other research topics in recommender systems, such as analyzing the performance changes resulting from attacking, defensing, and denoising methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62102279.

References

[1] Vito Walter Anelli, Daniele Malitesta, Claudio Pomo, Alejandro Bellogin, Eugenio Di Sciascio, and Tommaso Di Noia. 2023. Challenging the Myth of Graph Collaborative Filtering: a Reasoned and Reproducibility-driven Analysis. In *RecSys '23*. 350–361. doi:10.1145/3604915.3609489

[2] Payal Bajaj and Sumit Shekhar. 2016. Experience Individualization on Online TV Platforms Through Persona-based Account Decomposition. In *MM '16*. 252–256. doi:10.1145/2964284.2967221

[3] Lei Guo, Li Tang, Tong Chen, Lei Zhu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2021. DA-GCN: A Domain-aware Attentive Graph Convolution Network for Shared-account Cross-domain Sequential Recommendation. In *IJCAI '21*. 2483–2489. doi:10.24963/ijcai.2021/342

[4] Lei Guo, Jinyu Zhang, Tong Chen, Xinhua Wang, and Hongzhi Yin. 2023. Reinforcement Learning-Enhanced Shared-Account Cross-Domain Sequential Recommendation. *TKDE* 35, 7 (2023), 7397–7411. doi:10.1109/TKDE.2022.3185101

[5] Wenshuo Guo, Karl Krauth, Michael I. Jordan, and Nikhil Garg. 2021. The Stereotyping Problem in Collaboratively Filtered Recommender Systems. In *EAAMO '21*. Article 6, 10 pages. doi:10.1145/3465416.3483298

[6] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI '16*. 144–150. doi:10.1609/aaai.v30i1.9973

[7] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR '20*. 639–648. doi:10.1145/3397271.3401063

[8] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW '17*. 173–182. doi:10.1145/3038912.3052569

[9] Jyun-Yu Jiang, Cheng-Te Li, Yian Chen, and Wei Wang. 2018. Identifying Users behind Shared Accounts in Online Streaming Services. In *SIGIR '18*. 65–74. doi:10.1145/3209978.3210054

[10] Clément Lesaege, François Schnitzler, Anne Lambert, and Jean-Ronan Vigouroux. 2016. Time-Aware User Identification with Topic Models. In *ICDM '16*. 997–1002. doi:10.1109/ICDM.2016.0126

[11] Zihao Li, Yunfan Xie, Wei Emma Zhang, Pengfei Wang, Lixin Zou, Fei Li, Xiangyang Luo, and Chenliang Li. 2024. Disentangle interest trend and diversity for sequential recommendation. *IPM* 61, 3 (2024), 103619. doi:10.1016/j.ipm.2023.103619

[12] Tao Lian, Zhengxian Li, Zhumin Chen, and Jun Ma. 2017. The Impact of Profile Coherence on Recommendation Performance for Shared Accounts on Smart TVs. In *CCIR '17*. 30–41. doi:10.1007/978-3-319-68699-8_3

[13] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-Enriched Contrastive Learning. In *WWW '22*. 2320–2329. doi:10.1145/3485447.3512104

[14] Tropa Mahmood and Muhammad Abdullah Adnan. 2022. Detecting Fake Co-Visitation Injection Attack in Graph-Based Recommendation Systems. In *NSysS '22*. 30–40. doi:10.1145/3569551.3569556

[15] Daniele Malitesta, Giandomenico Cornacchia, Claudio Pomo, Felice Antonio Merra, Tommaso Di Noia, and Eugenio Di Sciascio. 2024. Formalizing Multimedia Recommendation through Multimodal Deep Learning. *TORS* (2024). doi:10.1145/3662738

[16] Kaili Mao, Jianwei Niu, Xuefeng Liu, Shaojie Tang, Lizi Liao, and Tat-Seng Chua. 2022. A Patience-Aware Recommendation Scheme for Shared Accounts on Mobile Devices. *TKDE* 34, 12 (2022), 5571–5585. doi:10.1109/TKDE.2021.3069002

[17] Leandro Balby Marinho, Júlio Barreto Guedes da Costa, Denis Parra, and Rodrygo L. T. Santos. 2022. Similarity-Based Explanations Meet Matrix Factorization via Structure-Preserving Embeddings. In *IUI '22*. 782–793. doi:10.1145/3490099.3511104

[18] Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. 2023. A Content-Driven Micro-Video Recommendation Dataset at Scale. arXiv:2309.15379

[19] Yilena Pérez-Almaguer, Raciel Yera, Ahmad A. Alzahrani, and Luis Martínez. 2021. Content-based group recommender systems: A general taxonomy and further improvements. *ESWA* 184 (2021), 115444. doi:10.1016/j.eswa.2021.115444

[20] Jiarui Qin, Jiachen Zhu, Yankai Liu, Junchao Gao, Jianjie Ying, Chaoxiong Liu, Ding Wang, Junlan Feng, Chao Deng, Xiaozheng Wang, Jian Jiang, Cong Liu, Yong Yu, Haitao Zeng, and Weinan Zhang. 2023. Learning to Distinguish Multi-User Coupling Behaviors for TV Recommendation. In *WSDM '23*. 204–212. doi:10.1145/3539597.3570374

[21] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI '09*. 452–461.

[22] Wenchao Sun, Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2023. Parallel Split-Join Networks for Shared Account Cross-Domain Sequential Recommendations. *TKDE* 35, 4 (2023), 4106–4123. doi:10.1109/TKDE.2021.3130927

[23] Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao. 2022. Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering. In *SIGIR '22*. 122–132. doi:10.1145/3477495.3531889

[24] Koen Verstrepen and Bart Goethals. 2015. Top-N Recommendation for Shared Accounts. In *RecSys '15*. 59–66. doi:10.1145/2792838.2800170

[25] Shilei Wang, Peng Zhang, Hui Wang, Hongtao Yu, and Fuzhi Zhang. 2022. Detecting shilling groups in online recommender systems based on graph convolutional network. *IPM* 59, 5 (2022), 103031. doi:10.1016/j.ipm.2022.103031

[26] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising Implicit Feedback for Recommendation. In *WSDM '21*. 373–381. doi:10.1145/3437963.3441800

[27] Zhijin Wang and Liang He. 2016. User identification for enhancing IP-TV recommendation. *KBS* 98 (2016), 68–75. doi:10.1016/j.knosys.2016.01.018

[28] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. In *MM '20*. 3541–3549. doi:10.1145/3394171.3413556

[29] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *MM '19*. 1437–1445. doi:10.1145/3343031.3351034

[30] Xinyu Wen, Zhaohui Peng, Shanshan Huang, Senzhang Wang, and Philip S. Yu. 2021. MISS: A Multi-user Identification Network for Shared-Account Session-Aware Recommendation. In *DASFAA '21*. 228–243. doi:10.1007/978-3-030-73200-4_15

[31] Guolei Yang, Nei Zhenqiang Gong, and Ying Cai. 2017. Fake Co-visitation Injection Attacks to Recommender Systems. In *NDSS '17*. doi:10.14722/ndss.2017.23020

[32] Shuo Yang, Somdeb Sarkhel, Saayan Mitra, and Viswanathan Swaminathan. 2017. Personalized Video Recommendations for Shared Accounts. In *ISM*. 256–259. doi:10.1109/ISM.2017.43

[33] Yang Yu, Qi Liu, Likang Wu, Runlong Yu, Sanshi Lei Yu, and Zaixi Zhang. 2023. Untargeted attack against federated recommendation systems via poisonous item embeddings and the defense. In *AAAI '23*. 4854–4863. doi:10.1609/aaai.v37i4.25611

[34] Wei Yuan, Quoc Viet Hung Nguyen, Tieke He, Liang Chen, and Hongzhi Yin. 2023. Manipulating Federated Recommender Systems: Poisoning with Synthetic Users and Its Countermeasures. In *SIGIR '23*. 1690–1699. doi:10.1145/3539618.3591722

[35] Kaike Zhang, Qi Cao, Yunfan Wu, Fei Sun, Huawei Shen, and Xueqi Cheng. 2024. Understanding and Improving Adversarial Collaborative Filtering for Robust Recommendation. arXiv:2410.22844

[36] Yong Zheng. 2024. An Empirical Comparison of Outlier Detection Methods for Identifying Grey-Sheep Users in Recommender Systems. In *AINA '24*. 62–73. doi:10.1007/978-3-031-57853-3_6

[37] Xin Zhou, Hongyu Zhou, Yong Liu, Zhiwei Zeng, Chunyan Miao, Pengwei Wang, Yuan You, and Feijun Jiang. 2023. Bootstrap Latent Representations for Multimodal Recommendation. In *WWW '23*. 845—-854. doi:10.1145/3543507.3583251

A Procedure for synthesizing shared accounts

Algorithm 1 presents the details of our procedure for synthesizing shared accounts. (1) It can establish strong spurious associations between a few item pairs while synthesizing shared accounts. In Step 1, we first randomly select two relatively popular items i and j that are not similar as measured by their overlap consumers in the controlled setting with only single-user accounts (cf. Lines 5–6). Next we identify the subset of users in $\mathcal{U}^2$ who have consumed

Algorithm 1: Synthesizing shared accounts

Input: $\mathcal{U}^2 = \{u_{m+1}, u_{m+2}, \ldots, u_{m+n}\}$, the number of item pairs with strong spurious association p, the mixing ratio ε, the proportion of shared accounts ρ

Output: The set of shared accounts $\mathcal{A}$

1 Let $\mathcal{I}'$ be the top 50% most popular items in $\mathcal{I}$;
2 $\mathcal{A} = \emptyset$;
`// Step 1: Simulating strong spurious item-item associations`
3 $c = 0$;
4 **while** $c < p$ **do**
5   Randomly select an item i from $\mathcal{I}'$;
6   Find item $j = \arg\min_{j \in \mathcal{I}'} \frac{|\mathcal{U}_i \cap \mathcal{U}_j|}{\sqrt{|\mathcal{U}_i| \cdot |\mathcal{U}_j|}}$;
7   Let $\mathcal{U}_i^2$ be the subset of users in $\mathcal{U}^2$ who have consumed item i ;
8   Let $\mathcal{U}_j^2$ be the subset of users in $\mathcal{U}^2$ who have consumed item j ;
9   $\mathcal{U}_i' \leftarrow \mathcal{U}_i^2 \setminus \mathcal{U}_j^2$;
10   $\mathcal{U}_j' \leftarrow \mathcal{U}_j^2 \setminus \mathcal{U}_i^2$;
11   $q = 0$;
12   **while** $q < \min\left(\left|\mathcal{U}_i'\right|, \left|\mathcal{U}_j'\right|\right)$ **do**
13   Randomly select a user $u \in \mathcal{U}_i'$ and a user $v \in \mathcal{U}_j'$;
14     $C_{u,v} = \mathcal{I}_u \cap \mathcal{I}_v$;
15     $O_u = \mathcal{I}_u \setminus C_{u,v} \setminus \{i\}$;
16     $O_v = \mathcal{I}_v \setminus C_{u,v} \setminus \{j\}$;
17     $\delta = \min\left(\lfloor \varepsilon \cdot \min\left(|\mathcal{I}_u|, |\mathcal{I}_v|\right)\rfloor, \min\left(|O_u|, |O_v|\right)\right)$, where $0 < \varepsilon < 1$;
18   Randomly select $\delta - 1$ items from O_u as $\Delta_{u \to v}$;
19   Randomly select δ items from O_v as $\Delta_{v \to u}$;
20     $\mathcal{I}_a \leftarrow (O_u \setminus \Delta_{u \to v}) \cup \Delta_{v \to u} \cup C_{u,v}$;
21     $\mathcal{I}_b \leftarrow (O_v \setminus \Delta_{v \to u}) \cup \Delta_{u \to v} \cup C_{u,v} \cup \{i, j\}$;
22     $\mathcal{U}^2 \leftarrow \mathcal{U}^2 \setminus \{u, v\}$, $\mathcal{I}' \leftarrow \mathcal{I}' \setminus \{i, j\}$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{a, b\}$;
23     $q = q + 1$;
24   **end**
25   $c = c + 1$;
26 **end**
`// Step 2: Mimicking weak noisy item-item associations`
27 **while** $|\mathcal{A}| < \lfloor \rho \cdot \left|\mathcal{U}^1 \cup \mathcal{U}^2\right| \rfloor \wedge \mathcal{U}^2 \neq \emptyset$ **do**
28   Randomly select two users u and v from $\mathcal{U}^2$;
29   $C_{u,v} = \mathcal{I}_u \cap \mathcal{I}_v$, $O_u = \mathcal{I}_u \setminus C_{u,v}$, $O_v = \mathcal{I}_v \setminus C_{u,v}$;
30   $\delta = \min\left(\lfloor \varepsilon \cdot \min\left(|\mathcal{I}_u|, |\mathcal{I}_v|\right)\rfloor, \min\left(|O_u|, |O_v|\right)\right)$, where $0 < \varepsilon < 1$;
31   Randomly select δ items from O_u as $\Delta_{u \to v}$;
32   Randomly select δ items from O_v as $\Delta_{v \to u}$;
33   $\mathcal{I}_a \leftarrow (O_u \setminus \Delta_{u \to v}) \cup \Delta_{v \to u} \cup C_{u,v}$;
34   $\mathcal{I}_b \leftarrow (O_v \setminus \Delta_{v \to u}) \cup \Delta_{u \to v} \cup C_{u,v}$;
35   $\mathcal{U}^2 \leftarrow \mathcal{U}^2 \setminus \{u, v\}$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{a, b\}$;
36 **end**

Table 7: The impact of the number of item pairs p with strong spurious association

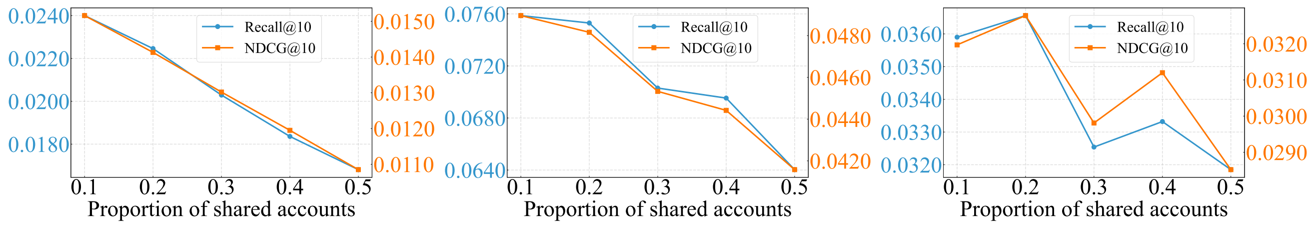
Dataset	p	% shared accounts	Recall@20	NDCG@20
Clothing	0	0	0.0403	0.0208
	50	1.39%	0.0396	0.0205
	100	2.69%	0.0392	0.0203
Beauty	0	0	0.1164	0.0623
	50	3.23%	0.1135	0.0616
	100	6.13%	0.1117	0.0609
MicroLens	0	0	0.0632	0.0447
	50	6.00%	0.0578	0.0411
	100	10.04%	0.0550	0.0382

either i or j but not both, denoted as $\mathcal{U}_i'$ and $\mathcal{U}_j'$ respectively (cf. Lines 7–10). Then for a pair of users $u \in \mathcal{U}_i'$ and $v \in \mathcal{U}_j'$, we swap a portion of their interacted items to form two synthetic shared accounts a and b (cf. Lines 13–21) such that there emerges a fake co-occurrence of items i and j in one of them (e.g., b). The spurious association between items i and j will become stronger when the mixing operation is repeated for more user pairs in $\mathcal{U}_i' \times \mathcal{U}_j'$. (2) It can also mimic weak noisy associations between relatively more items. The operations in Step 2 (cf. Lines 28–34) are similar to those in Step 1 (cf. Lines 13–21); the difference is that the two random users here do not necessarily have consumed item i or j . (3) The data sparsity keeps the same as in the controlled setting, since the number of accounts and the total amount of observed interactions remain unchanged.

B Hyper-parameter sensitivity

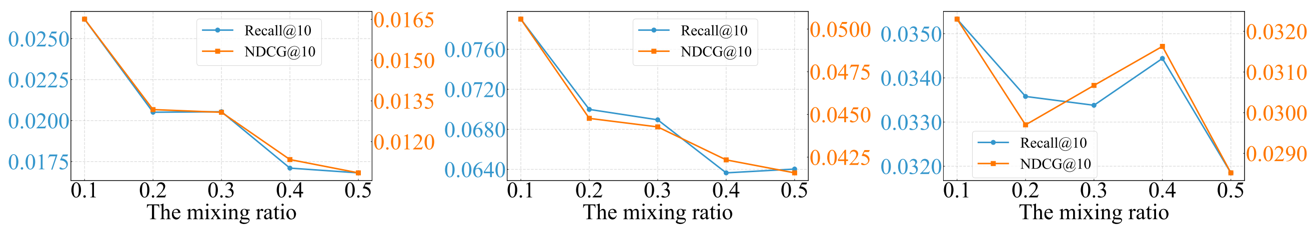
First we explored the influence of the number of item pairs with strong spurious association on the recommendation performance. We incrementally varied p in the range {0, 50, 100} with $\varepsilon = 0.5$. Note that ρ was set to 0, which in fact skipped Step 2 of Algorithm 1 that only mimics weak noisy item-item associations in shared accounts. The results are presented in Table 7. Note that there are no shared accounts when $p = 0$, which is the same as the controlled setting. When p increases to 50 and further to 100, although the number of item pairs with strong spurious association is only about 1% of the total number of items in the dataset, there is a marked decrease in the recommendation performance, especially on the MicroLens dataset. The performance decline on the Clothing dataset is less obvious, since the number of shared accounts resulting from Step 1 of Algorithm 1 only accounts for about 2% of the total number of both single-user accounts and shared accounts. Although depending on the dataset, a few item pairs with strong spurious association caused by a small proportion of shared accounts can result in a marked performance decline for other single-user accounts.

Next we explored the influence of the proportion of shared accounts on the recommendation performance. We gradually increased ρ with $\varepsilon = 0.5$ and $p = 50$, and observe the recommendation performance for single-user accounts in $\mathcal{U}^1$. The results are shown



(a) Clothing (b) Beauty (c) MicroLens

Figure 6: Variation of the performance for single-user accounts w.r.t. the proportion of shared accounts



(a) Clothing (b) Beauty (c) MicroLens

Figure 7: Variation of the performance for single-user accounts w.r.t. the mixing ratio

in Figure 6. We can observe a continuous performance decline as ρ increases, which could be attributed to the increasing amount of weak noisy item-item associations caused by more and more shared accounts in Step 2 of Algorithm 1.

Finally we investigated how the recommendation performance for single-user accounts varies with respect to the mixing ratio ε (i.e., the proportion of two users' consumed items that are swapped to form two synthetic shared accounts). We adjusted ε in the range $\{0.1, 0.2, 0.3, 0.4, 0.5\}$, with $p = 50$ and $\rho = 0.5$. As shown in Figure 7, both Recall@10 and NDCG@10 for single-user accounts gradually decrease with the increasing of ε. The possible reason is that more and more fake co-occurrences are introduced among items by shared accounts, misleading the model about the relations between items.