

Graph Collaborative Signals Denoising and Augmentation for Recommendation

Ziwei Fan, Ke Xu
University of Illinois Chicago
USA
{zfan20,kxu25}@uic.edu

Zhang Dong
Amazon Ads
USA
andydong@amazon.com

Hao Peng*
Beihang University
China
penghao@act.buaa.edu.cn

Jiawei Zhang
University of California, Davis
USA
jiawei@ifmlab.org

Philip S. Yu
University of Illinois Chicago
USA
psyu@uic.edu

ABSTRACT

Graph collaborative filtering (GCF) is a popular technique for capturing high-order collaborative signals in recommendation systems. However, GCF's bipartite adjacency matrix, which defines the neighbors being aggregated based on user-item interactions, can be noisy for users/items with abundant interactions and insufficient for users/items with scarce interactions. Additionally, the adjacency matrix ignores user-user and item-item correlations, which can limit the scope of beneficial neighbors being aggregated.

In this work, we propose a new graph adjacency matrix that incorporates user-user and item-item correlations, as well as a properly designed user-item interaction matrix that balances the number of interactions across all users. To achieve this, we pre-train a graph-based recommendation method to obtain users/items embeddings, and then enhance the user-item interaction matrix via top-K sampling. We also augment the symmetric user-user and item-item correlation components to the adjacency matrix. Our experiments demonstrate that the enhanced user-item interaction matrix with improved neighbors and lower density leads to significant benefits in graph-based recommendation. Moreover, we show that the inclusion of user-user and item-item correlations can improve recommendations for users with both abundant and insufficient interactions. The code is in <https://github.com/zfan20/GraphDA>.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Collaborative filtering*.

KEYWORDS

Collaborative Filtering, Denoising, Augmentation, Graph Recommendation

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org).
SIGIR '23, July 23-27, 2023, Taipei, Taiwan, China

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN XXXXXXXXXXXXXXXX...\$15.00
<https://doi.org/XXXXXXXXXXXXX>

ACM Reference Format:

Ziwei Fan, Ke Xu, Zhang Dong, Hao Peng, Jiawei Zhang, and Philip S. Yu. 2023. Graph Collaborative Signals Denoising and Augmentation for Recommendation. In *Proceedings of the ACM SIGIR Conference 2023 (SIGIR '23)*, July 23-27, 2023, Taipei, Taiwan, China. ACM, New York, NY, USA, 6 pages. <https://doi.org/XXXXXXXXXXXXX>

1 INTRODUCTION

Graph-based recommender systems use neighborhood information to infer user/item embeddings, where the adjacency matrix defines the neighborhood structure. High-order collaborative signals are typically aggregated by stacking multiple layers [3, 7, 13, 19, 20, 23, 30, 33, 35]. However, the quality of the neighborhood information depends on the definition of the adjacency matrix. The widely adopted adjacency matrix is built upon the user-item interaction matrix, which potentially encounters noises [10, 31, 32], sparsity [2, 11, 12, 22], biases [1, 6, 38], and long tail [9, 26, 28] issues. As shown in Fig. (1), we can observe that both users and items are following the long-tail distribution, where majority of users/items have limited interactions. Moreover, one counter-intuitive observation in Fig. (1) is that users with rich interactions (i.e., active users) are poorly modeled, compared with users with scarce interactions (i.e., inactive users) [17]. Arguably, the underlying reason is that highly active users have abundant noisy interactions, which even might be harmful to the user preference modeling. Furthermore, more noises are introduced when the graph model stacks more layers of graph convolutions [10]. From the item side, we can observe that items with limited interactions are performed unsatisfactorily.

Based on these observations, we argue that the current definition of the bipartite adjacency matrix in graph-based recommender systems is inadequate. As shown in Figure (2), the bipartite adjacency matrix A is constructed directly from the user-item interaction matrix R to define the neighborhood structure for users/items. However, R suffers from noisy and sparse interactions, making it insufficient to represent inactive users/items. Additionally, the bipartite adjacency matrix A overlooks the user-user [21, 37] and item-item correlations [8, 14, 27] in the neighborhood definition, even in the enhanced solution [5, 34]. Although high-order collaborative signals can uncover these correlations via multi-hop message passing, recent studies have shown that long-distance message passing can create new learning problems and lead to suboptimal representations [4]. Therefore, we propose a novel adjacency matrix design to improve the graph-based recommendation.

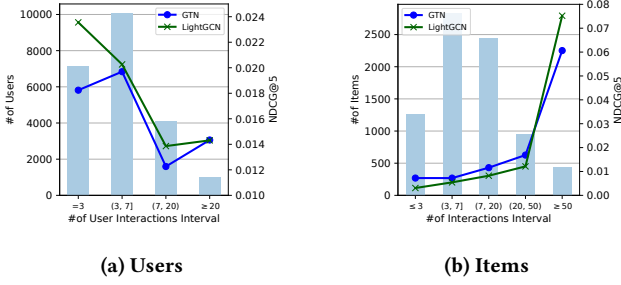


Figure 1: The interactions amount distribution of users/items (bar) and corresponding NCDG@5 (line) on Amazon Beauty dataset by two graph models LightGCN [13] and GTN [10].

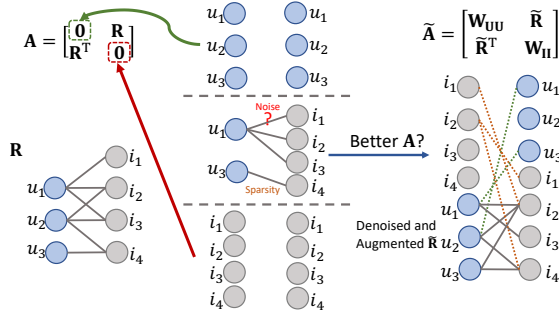


Figure 2: Motivation figure with a toy example. The A and R are the bipartite adjacency matrix and the user-item interaction matrix for graph-based recommendations, respectively.

It is challenging to properly design a better adjacency matrix. Some relevant distillation methods [15, 36] learn a smaller but useful graph data for graph modeling. However, one significant difference with distillation methods is that the GCF utilizes users and items IDs as inputs, and thus existing works of graph condensation are not applicable in our GCF setting. Moreover, these distillation methods assume the availability of features while users' features are sometimes not accessible due to privacy constraints.

To this end, we propose a pre-training and enhancing pipeline framework, namely GraphDA, to denoise and augment the user-item matrix. Within GraphDA, we capture the user-user and item-item correlations in the bipartite adjacency matrix for the GCF. Specifically, we first pre-train an encoder to generate the users/items embeddings from existing user-item interactions. With pre-trained embeddings, we adopt the top-K sampling process to generate the denoised and augmented user-item matrix, non-zero user-user and item-item correlations. Our contributions include:

- We investigate the deficiency of the existing definition of the bipartite adjacency matrix for GCF and study the potential of introducing a better adjacency matrix.
- We propose a better adjacency matrix generation for the graph-based recommendation, with a novel pipeline GraphDA for denoising for active users and augmenting for inactive users.

- Comprehensive experiments show that the proposed GraphDA significantly benefits the graph-based recommendation, especially on highly active users and inactive users, who demand denoising and augmentation, respectively.

2 GRAPH COLLABORATIVE FILTERING

In the graph collaborative filtering (GCF), we denote the user set as $\mathcal{U}$ and the item set as $\mathcal{I}$, where the user and item are indexed by u and i . With either implicit or explicit feedback, the user-item interaction matrix is given as $R \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$, where R_{ui} denotes the feedback on the item i given by the user u . For example, R_{ui} of implicit feedback is either 1 or 0. As the R is a user-item bipartite graph, the adjacency matrix is further formatted as: $A = \begin{bmatrix} \mathbf{0} & R \\ R^T & \mathbf{0} \end{bmatrix}$, where

$A \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times (|\mathcal{I}|+|\mathcal{U}|)}$. This problem can also be interpreted as the link prediction problem between user and item nodes. The user and item embeddings are randomly initialized and optimized with existing user-item interactions. Specifically, we denote the user and item embeddings table as $E \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|) \times d}$, where d denotes the latent dimension of embeddings. GCF incorporates high-order collaborative signals [13, 30] by stacking multiple layers of graph convolution on the user-item adjacency matrix A . Specifically, the output embeddings generation process with N graph convolution layers is given as:

$$E^{(N)} = \text{Encoder}(A, E) = (L)^{N-1} E^{(0)}, \quad (1)$$

where $E^{(0)} = E$, L refers to the bipartite Laplacian matrix, which is defined as the normalized symmetric Laplacian matrix $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, and D is the degree matrix. The representative work LightGCN [13] averages the generated embeddings over all layers as the final output embedding. The user preference prediction between the user u and item i is given as:

$$P(i|u, A) = \sigma(e_u^T e_i) \quad \text{where } i \in \mathcal{I} \setminus \mathcal{I}_u^+, \quad (2)$$

where $\sigma(\cdot)$ denotes the sigmoid activation function, $\mathcal{I}_u^+$ denotes the observed interacted item set by the user u , e_u and e_i are the user and item output embeddings of $E^{(N)}$.

3 PROPOSED FRAMEWORK

In this section, we introduce our proposed framework GraphDA with both user-item interaction matrix, user-user and item-item correlations enhancements. The framework consists of two steps, including the users/items representations pre-trained from a graph encoder and neighbors generation processes for enhancement.

3.1 Pre-Trained Users/Items Embeddings

With the arguably imperfect graph Laplacian matrix L from the original adjacency matrix A , we pre-train a graph encoder to obtain the users/items representations, which is shown as pre-train in the left part of Fig. (3). Specifically, we use N graph convolution layers to obtain users/items embeddings $E^{(N)}$ as described in Eq. (1). The pre-train step is optimized with the training data using the BPR loss as:

$$\mathcal{L} = - \sum_{(u, i^+, i^- \in \mathcal{R})} \log \sigma(e_u^T e_{i^+} - e_u^T e_{i^-}), \quad (3)$$

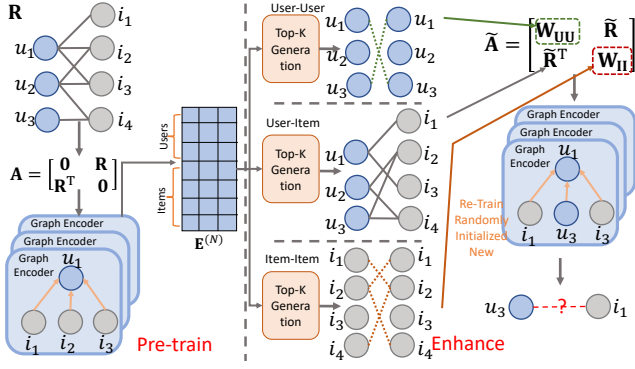


Figure 3: Workflow Diagram of GraphDA. GraphDA consists of two steps: 1. the pre-train step infers users/items embeddings; 2. utilize embeddings to generate top-K neighbors for the user-item component, the user-user component, and the item-item component, the enhanced adjacency matrix is used to re-train a graph encoder.

where i^+ denotes the positively interacted item of the user u , i^- is a sampled negative item without interaction with the user u , and e_u and e_i are the user and item output embeddings in $E^{(N)}$. Note that several encoders with various architectures modeling user-item interactions can be the alternative choice, such as the classical matrix factorization [29].

3.2 Enhanced Bipartite Adjacency

The pre-trained $E^{(N)}$ encodes user-item collaborative signals. However, its crucial component in GCF, i.e., the adjacency matrix A , is arguably less satisfactory for users/items embeddings learning, due to the biased interactions observed as the long-tailed distribution, noisy interactions for active users, and the ignoring of direct user-user and item-item correlations. Specifically, we enhance the adjacency matrix in three components, as shown in the central component of Fig. (3).

User-item Interactions Enhanced. With the pre-trained $E^{(K)}$, we generate the top-K neighbors for both users and items. For users, the top-K neighbors define the preference while neighbors for items represent the concept of the user group being marketed. From the user side, we define a hyper-parameter U_k to control the number of neighbors being selected. For all users, the number of neighbors is the same U_k . With equal number of neighbors, users with abundant neighbors in the original data are denoised while users with scarce neighbors are augmented. Specifically, for the user u , the top-K neighbors are generated by selecting the top U_k elements with largest values from the output scores $e_u^T E_I^{(N)}$, i.e.,:

$$\arg \max_{\{i_1, i_2, \dots, i_{U_k} \in I\}} e_u^T E_I^{(N)}, \quad (4)$$

where $E_I^{(N)}$ denotes output item embeddings. Similarly, for the item side, we also define the hyper-parameter I_k to generate the top-K neighbors with the similar process. We adopt the union of generated user-item interactions from both user and item sides to obtain the enhanced $\tilde{R}$.

User-User and Item-Item Correlations. For the recommender system dataset with unknown user-user/item-item interactions, the corresponding sub-matrices in conventional adjacency matrix A representation (as indicated at the beginning of Section 2) will be filled with zeros. We propose to complement these two all-zero sub-matrices with W_{UU} and W_{II} to further enhance the adjacency matrix. Specifically, for a user u , we extract the top- UU_k similar users as:

$$\arg \max_{\{u_1, u_2, \dots, u_{UU_k} \in U\}} e_u^T E_u^{(N)}, \quad (5)$$

where $E_u^{(N)}$ denotes the output user embeddings, and $W_{UU}[u, u_k] = 1$ for $u_k \in \{u_1, u_2, \dots, u_{UU_k}\}$. The similar process is conducted for W_{II} with I_k to control the number of similar items being chosen. Note that we enforce the W_{UU} and W_{II} to be symmetric.

GCF Re-learn with Enhanced Bipartite Adjacency. With the enhanced graph adjacency matrix, we re-learn the graph encoder (i.e., randomly initialized) to generate embeddings for user-item interaction predictions, which is shown in the right part in Fig. (3). To better illustrate each component's contribution, we propose two version of enhancements, Enhanced-UI and GraphDA. The Enhanced-UI only adopts the user-item enhanced interactions, i.e., its adjacency matrix $\tilde{A} = \begin{bmatrix} 0 & \tilde{R} \\ \tilde{R}^T & 0 \end{bmatrix}$. The complete version GraphDA includes user-user and item-item correlations, having the adjacency matrix $\tilde{A} = \begin{bmatrix} W_{UU} & \tilde{R} \\ \tilde{R}^T & W_{II} \end{bmatrix}$.

4 EXPERIMENTS

This section presents experiments for demonstrating the effectiveness of the proposed framework GraphDA. We answer following research questions (RQs): **RQ1:** Does GraphDA achieve better recommendation performances than existing baselines? **RQ2:** What are the effects of hyper-parameters and each component in GraphDA? **RQ3:** Does GraphDA achieve the goals of denoising and augmentation simultaneously?

4.1 Experimental Settings

Datasets. We use the public Amazon Reviews dataset [25] with three benchmark categories [10, 13, 18, 30], including: (1) *Beauty* has 22,363 users, 12,101 items, and 198,502 interactions with 0.05% density; (2) *Toys and Games* (Toys) has 19,412 users, 11,924 items, and 167,597 interactions with 0.07% density; (3) *Tools and Home* (Tools) has 16,638 users, 10,217 items, and 134,476 interactions with 0.08% density. We follow the 5-core setting as existing works on users and the same transformation [10, 13, 30] of treating the existence of reviews as positives. We sort each user's interactions chronologically and adopt the leave-one-out setting, with the last interacted item for testing and the second last interaction for validation.

Evaluations. We adopt the widely used standard ranking evaluation metrics to evaluate the averaged ranking performance over all users, including Recall@N and NDCG@N, which are widely used in existing works [10, 13, 30]. For the fair comparison without the sampling bias, we adopt the **all items ranking evaluation** [16]. Moreover, we use **only one training negative item** during the training process. Recall@N measures the correctness of ranking the

Table 1: Overall Comparison Table in HR@20 and NDCG@20. The best baseline and best model are underlined and in bold. ‘Improv.’ indicates the relative improvements over the best baseline.

Dataset	Beauty				Toys				Tools				Office			
	H@10	N@10	H@20	N@20	H@10	N@10	H@20	N@20	H@10	N@10	H@20	N@20	H@10	N@10	H@20	N@20
NGCF	0.0447	0.0232	0.0724	0.0299	0.0461	0.0251	0.0672	0.0306	0.0329	0.0179	0.0480	0.0216	0.0261	0.0159	0.0453	0.0208
UltraGCN	0.0451	0.0234	0.0728	0.0304	0.0464	0.0250	0.0675	0.0308	0.0331	0.0179	0.0481	0.0217	0.0302	0.0171	0.0471	0.0210
GTN	0.0446	0.0230	0.0680	0.0289	0.0453	0.0248	0.0661	0.0301	0.0337	0.0184	0.0484	<u>0.0221</u>	0.0283	0.0161	0.0453	0.0204
LightGCN	<u>0.0471</u>	<u>0.0244</u>	<u>0.0730</u>	<u>0.0309</u>	<u>0.0512</u>	<u>0.0273</u>	<u>0.0716</u>	<u>0.0325</u>	0.0334	0.0182	0.0482	0.0219	<u>0.0355</u>	<u>0.0197</u>	<u>0.0522</u>	<u>0.0238</u>
Enhanced-UI	0.0486	0.0252	0.0755	0.0317	0.0530	0.0276	0.0765	0.0335	0.0364	0.0195	0.0527	0.0236	0.0363	0.0208	0.0565	0.0259
Improv.	+3.2%	+3.3%	+3.1%	+2.9%	+3.5%	+1.1%	+6.8%	+3.3%	+8.0%	+6.0%	+4.3%	+6.6%	+2.3%	+5.6%	+8.2%	+8.8%
GraphDA	0.0514	0.0264	0.0804	0.0336	0.0549	0.0289	0.0795	0.0347	0.0373	0.0205	0.0532	0.0245	0.0383	0.0225	0.0561	0.0270
Improv.	+9.1%	+8.2%	+8.7%	+7.9%	+7.2%	+5.9%	+11.1%	+6.9%	+10.7%	+11.4%	+5.4%	+10.8%	+7.9%	+14.2%	+7.5%	+13.4%

ground-truth items in the top-N list, regardless of the ranking position. NDCG@N extends to further give higher weights to higher ranking positions. We only report the $N = 20$ due to limited space. **Baselines.** We compare several graph-based recommendation methods, including LightGCN [13], NGCF [30], UltraGCN [24], and GTN [10].

Hyper-Parameters Grid Search. For all methods, we use the dimension as 128 and learning rate as 0.001. We search the L2 regularization weight in $\{0.0, 0.01, 0.001, 0.0001\}$. For NGCF, we search the node and message dropouts from $\{0.1, 0.3, 0.5\}$. For GTN, we search its specific β and λ from $\{0.001, 0.005, 0.01, 0.05, 0.1\}$. For UltraGCN, we search their weights from $\{1, 1e-3, 1e-5, 1e-7\}$ and negative weights from $\{10, 50, 100\}$. All hyper-parameters are grid-searched and the test performance is reported based on the best validation performance. We search the U_k and I_k from $\{0, 3, 5, 7, 9\}$. We also search the UU_k and II_k from $\{0, 3, 5, 7\}$.

4.2 Overall Comparison (RQ1)

We obtain several observations from the overall comparison Table 1:

- **Enhancing the bipartite Laplacian matrix benefits the graph-based recommendation.** The proposed framework GraphDA and its variant Enhanced-UI achieve significant improvements over existing graph-based recommendation methods. The superiority of GraphDA and Enhanced-UI demonstrate the benefits of enhancing the bipartite Laplacian matrix.
- **User-user and item-item correlations are beneficial.** The user-user and item-item correlations enhancements on the bipartite Laplacian contributes to further performance improvements. When we compare Enhanced-UI (with only the user-item component) and GraphDA (with additional user-user and item-item correlations), GraphDA outperforms Enhanced-UI.
- Among existing graph recommendation methods, LightGCN achieves the best baseline performance. The second best baseline is GTN or UltraGCN, depending on datasets. Due to its simplicity, LightGCN can achieve satisfactory performances, and GTN proposes the trend filtering mechanism to denoise the user interactions in the LightGCN framework. UltraGCN infuses the item-item similarities in the graph recommendation. These comparisons demonstrate the necessity of incorporating item-item correlations and user interactions denoising process in the graph recommendation.

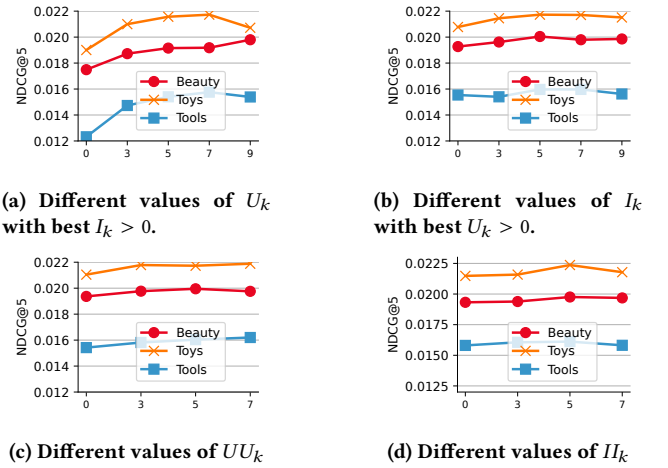


Figure 4: Different values of hyper-parameters, including U_k , I_k , UU_k for W_{UU} , II_k for W_{II} , with definitions in Section 3.2.

4.3 Hyper-Parameters Sensitivity (RQ2)

We visualize changes of NDCG@5 from hyper-parameters U_k , I_k , UU_k , and II_k , which control the number of neighbors from different semantics for purposes in Fig. (4). We have following observations:

- **The enhanced collaborative neighbors benefit the overall performance.** From Fig. (4a) and Fig. (4b), we can see that the proper choices of $U_k > 0$ and $I_k > 0$ improve the performances. The special cases $U_k = 0$ and $I_k = 0$ happen when we generate neighbors from either only the item side ($U_k = 0$ and $I_k > 0$) in Fig. (4a) or only the user side ($I_k = 0$ and $U_k > 0$) in Fig. (4b).
- **The user side generated neighbors U_k are important.** From Fig. (4a), we can observe that removing neighbors from the user side ($U_k = 0$) causes significant performance degradation. From Fig. (4b), having more neighbors from the item side ($I_k > 0$) benefits but with marginal improvements.
- **Small numbers of neighbors in W_{UU} and W_{II} are sufficient.** From Fig. (4c) and Fig. (4d), we can observe that non-zero neighbors on W_{UU} and W_{II} can benefit the recommendation, which is also observed in Table 1. However, the larger values of UU_k and II_k do not bring significant improvements, where the lines in Fig. (4c) and Fig. (4d) are smooth. This observation demonstrates that the potentially small UU_k and II_k can be sufficient.

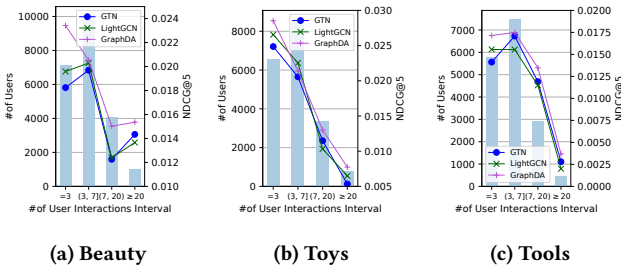


Figure 5: Improvements analysis of users grouped by the number of interactions. We only show two strong baselines to avoid cluttering figures.

4.4 Improvements Analysis (RQ3)

We validate that the proposed GraphDA can denoise and augment for different groups of users, grouped by numbers of interactions, as shown in Fig. (5). We have following observations:

- **The augmented neighbors by GraphDA benefit inactive users significantly in the graph-based recommendation.** For inactive users with least interactions (i.e., with less than 3 interactions), the proposed GraphDA achieves significant improvements over the best baseline with range from 3.7% to 19.6%.
- **The denoised interactions by GraphDA improve highly active users with noisy interactions, especially in datasets with potentially more noises.** For highly active users with abundant interactions (i.e., with more than 7 interactions), the proposed GraphDA achieves comparative and mostly better performance than LightGCN, with improvements from 5.1% to 67.8%. The proposed GraphDA and GTN both benefit the highly active users with a large margin over LightGCN in the Tools dataset, which potentially includes more noisy user interactions. Table 1 also shows potential noises in the Tools, where the denoising GTN is the best baseline.

5 CONCLUSIONS

We empirically investigate the existing deficiencies of graph-based recommendations, and arguably identify that issues come from the unsatisfactory definition of the bipartite adjacency matrix. To generate a better bipartite adjacency matrix, we propose the denoising and augmentation pipeline GraphDA with pre-training and enhancing steps to generate a better user-item matrix, user-user correlations, and also the item-item correlations. Experiments show the superiority of GraphDA, especially for highly active users and inactive users.

ACKNOWLEDGMENTS

This paper was supported by the National Key R&D Program of China through grant 2022YFB3104703, NSFC through grant 62002007, Natural Science Foundation of Beijing Municipality through grant 4222030, and S&T Program of Hebei through grant 21340301D. Philip S. Yu was supported by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941. This work is also partially supported by NSF through grants IIS-1763365 and IIS-2106972.

REFERENCES

- [1] Himan Abdollahpour, Masoud Mansoury, Robin Burke, and Bamshad Mobasher. 2019. The unfairness of popularity bias in recommendation. *arXiv preprint arXiv:1907.13286* (2019).
- [2] Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. 2017. A generic coordinate descent framework for learning from implicit feedback. In *Proceedings of the Web Conference*. 1341–1350.
- [3] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2018. Graph convolutional matrix completion. In *Proceedings of the Deep Learning Day@ACM SIGKDD*.
- [4] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI*, Vol. 34. 3438–3445.
- [5] Huiyuan Chen, Lan Wang, Yusan Lin, Chin-Chia Michael Yeh, Fei Wang, and Hao Yang. 2021. Structured graph convolutional networks with stochastic masks for recommender systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 614–623.
- [6] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* (2020).
- [7] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the ACM SIGIR*. 765–774.
- [8] Evangelia Christakopoulou and George Karypis. 2016. Local item-item models for top-n recommendation. In *Proceedings of the ACM RecSys*. 67–74.
- [9] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.
- [10] Wendi Fan, Xiaorui Liu, Wei Jin, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2022. Graph trend filtering networks for recommendation. In *Proceedings of the ACM SIGIR*. 112–121.
- [11] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the ACM WSDM*. 265–273.
- [12] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the Web Conference*. 507–517.
- [13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the ACM SIGIR*. 639–648.
- [14] Peng Hu, Rong Du, Yao Hu, and Nan Li. 2019. Hybrid Item-Item Recommendation via Semi-Parametric Embedding. In *Proceedings of the IJCAI*. 2521–2527.
- [15] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. 2022. Graph Condensation for Graph Neural Networks. In *Proceedings of the ICLR*.
- [16] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1748–1757.
- [17] Xuan Nhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. 2008. Addressing cold-start problem in recommendation systems. In *Proceedings of the IEEE ICOM*. 208–211.
- [18] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time interval aware self-attention for sequential recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 322–330.
- [19] Xu Lin, Panagiotis Ili, and Jason Polakis. 2020. Fill in the blanks: Empirical analysis of the privacy threats of browser form autofill. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 507–519.
- [20] Xu Lin, Panagiotis Ili, Saumya Solanki, and Jason Polakis. 2022. Phish in Sheep's Clothing: Exploring the Authentication Pitfalls of Browser Fingerprinting. In *31st USENIX Security Symposium (USENIX Security 22)*. 1651–1668.
- [21] Siwei Liu, Zaiqiao Meng, Craig Macdonald, and Iadh Ounis. 2023. Graph Neural Pre-Training for Recommendation with Side Information. *ACM Transactions on Information Systems* 41, 3 (2023).
- [22] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and S Yu Philip. 2020. Basket recommendation with multi-intent translation graph neural network. In *Proceedings of the BigData*. 728–737.
- [23] Zhiwei Liu, Liangwei Yang, Ziwei Fan, Hao Peng, and Philip S Yu. 2022. Federated Social Recommendation with Graph Neural Network. *ACM Trans. Intell. Syst. Technol.* (2022), 1–24.
- [24] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of ACM CIKM*. 1253–1262.
- [25] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the ACM SIGIR*. 43–52.
- [26] Staša Milojević. 2010. Power law distributions in information science: Making the case for logarithmic binning. *Journal of the American Society for Information Science and Technology* 61, 12 (2010), 2417–2425.

- [27] Xia Ning and George Karypis. 2011. Slim: Sparse linear methods for top-n recommender systems. In *Proceedings of the IEEE ICDM*. 497–506.
- [28] Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the ACM RecSys*. 11–18.
- [29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the UAI* (Montreal, Quebec, Canada) (*UAI '09*). AUAI Press, Arlington, Virginia, USA, 452–461.
- [30] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
- [31] Yu Wang, Zhiwei Liu, Ziwei Fan, Lichao Sun, and Philip S. Yu. 2021. *DSKReG: Differentiable Sampling on Knowledge Graph for Recommendation with Relational GNN*. Association for Computing Machinery, New York, NY, USA, 3513–3517. <https://doi.org/10.1145/3459637.3482092>
- [32] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the ACM SIGIR*. 726–735.
- [33] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S Yu. 2022. Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In *Proceedings of the ACM Web Conference 2022*. 3376–3386.
- [34] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2021. Enhanced graph learning for collaborative filtering via mutual information maximization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 71–80.
- [35] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the SIGKDD*, Yike Guo and Faisal Farooq (Eds.). 974–983.
- [36] Wenqing Zheng, Edward W Huang, Nikhil Rao, Sumeet Katariya, Zhangyang Wang, and Karthik Subbian. 2022. Cold Brew: Distilling Graph Node Representations with Incomplete or Missing Neighborhoods. In *Proceedings of the ICLR*.
- [37] Ziwei Zhu, Jianling Wang, and James Caverlee. 2019. Improving top-k recommendation via jointcollaborative autoencoders. In *The World Wide Web Conference*. 3483–3482.
- [38] Ziwei Zhu, Jianling Wang, and James Caverlee. 2020. Measuring and mitigating item under-recommendation bias in personalized ranking systems. In *Proceedings of the ACM SIGIR*. 449–458.