SuGeR: A Subgraph-based Graph Convolutional Network Method for Bundle Recommendation

Zhenning Zhang
zz45@illinois.edu
University of Illinois at Urbana-Champaign
Champaign, Illinois, USA

Boxin Du
boxindu2@illinois.edu
University of Illinois at Urbana-Champaign
Champaign, Illinois, USA

Hanghang Tong
htong@illinois.edu
University of Illinois at Urbana-Champaign
Champaign, Illinois, USA

ABSTRACT

Bundle recommendation is an emerging research direction in the recommender system with the focus on recommending customized bundles of items for users. Although Graph Neural Networks (GNNs) have been applied to this problem and achieved superior performance, existing methods underexplore the graph-level GNN methods, which exhibit great potential in traditional recommender system. Furthermore, they usually lack the transferability from one domain with sufficient supervision to another domain which might suffer from the label scarcity issue. In this work, we propose a subgraph-based Graph Neural Network model, SuGeR, for bundle recommendation to handle these limitations. SuGeR generates heterogeneous subgraphs around the user-bundle pairs and then maps those subgraphs to the users' preference predictions via neural relational graph propagation. Experimental results show that SuGeR significantly outperforms the state-of-the-art baselines in the basic and the transfer bundle recommendation tasks by up to 77.17% by NDCG@40. The source code is available at: https://github.com/Zhang-Zhenning/SUGER.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Recommender systems** → *Bundle Recommendation.*

KEYWORDS

recommender system, graph neural networks

ACM Reference Format:
Zhenning Zhang, Boxin Du, and Hanghang Tong. 2022. SuGeR: A Subgraph-based Graph Convolutional Network Method for Bundle Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management(CIKM'22), October 17-21,2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557707

1 INTRODUCTION

Bundle recommendation is a newly emerging research direction in the recommender system. Generally, bundle recommendation aims at recommending a bundle of items which collectively might be more appealing to users, compared with recommending single items. For example, large online game and music distributors such as Steam, Tencent, Netease, and large e-commerce platforms such as Amazon and Taobao have already begun to sell products in bundles [1, 4, 5]. This recommendation strategy would benefit both sellers and customers mutually.

However, despite its importance, less effort has been devoted to this direction. Existing approaches for traditional recommender systems tend to fall short of the bundle recommendation for two main reasons. First, the bundle recommendation problem contains three types of interactions: user-item preference, user-bundle preference, and bundle-item membership. It is non-trivial for traditional user-item recommendation methods to incorporate various types of interactions. Second, the label scarcity problem and the cold-start problem for certain domains are crucial in a newly rising direction. One possible solution is to learn and transfer knowledge from other domains with sufficient supervision. However, most traditional methods do not have transferability between various domains.

Recently, GNNs demonstrate the potential strength on a variety of applications including recommender system [2, 3, 6, 7, 14]. However, leveraging the graph-level GNN in the bundle recommendation scenario is still an open question. Futhermore, the label leakage issue for bundle recommendation is often neglected. But it has a significant impact in terms of overfitting and performance. In addition, most current methods could not be effectively generalized to unseen data or even transferred from one domain to another.

In this paper, we propose a Subgraph-based Graph Convolutional Network model (SuGeR), to tackle all of the aforementioned limitations. The key high-level idea is to construct a heterogeneous subgraph for each user-bundle pair and map it to predict the user's preference via graph-level GNNs. Our method can handle both the basic bundle recommendation problem and the transfer bundle recommendation problem, in which the goal is to apply the model learned in one domain to another.

Related works. The classic recommendation model such as [16] could not be applied to the bundle task. Recently, many research works begin to introduce novel techniques for recommender system, such as GNN [11] and co-clustering [10]. Some GNN-based recommendation models such as NGCF [12] focuses on homogeneous graph and have a inferior performance under bundle context. Representative bundle-oriented models are DAM [4] and BGCN [2] where the latter is the state-of-the-art. The GNN-based matrix completion methods like IGMC [18] provide new inspiration. The key of IGMC is to extract enclosing graph of the user-item pairs, and adopt the graph-level GNN to map the user-item pair to prediction scores. However, IGMC is not particularly designed for bundle scenarios. The distinctive difference between SuGeR and IGMC

CIKM '22, October 17-21, 2022, Atlanta, GA, USA.

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557707

includes the generation of the k-hop heterogeneous subgraph, layer structure design, and handling label leakage issue.

2 PROBLEM DEFINITION

We use $\mathcal{U}, \mathcal{I}, \mathcal{B}$ to denote user set, item set and bundle set for training and testing. Then, we use $|\mathcal{U}|, |\mathcal{I}|, |\mathcal{B}|$ to denote the number of users, items and bundles. We define $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ as user-bundle, user-item, and bundle-item interaction matrices respectively (the left side of Figure 1). $\mathbf{X}$ and $\mathbf{Y}$ represent users' bundle and item preference, and $\mathbf{Z}$ represents bundle-item membership. Taking $\mathbf{X}$ as an example. $\mathbf{X}(u, b) = 1$ if there exists observed interactions between user u and bundle b where $u \in \mathcal{U}, b \in \mathcal{B}$. $\mathbf{Y}$ and $\mathbf{Z}$ follow similar definitions. These three matrices then be split into training and test sets. Based on these fundamental symbol definitions, the basic bundle recommendation problem is defined as follows:

Problem 1. ***Basic Bundle Recommendation:***

Given: *The user set $\mathcal{U}$, item set $\mathcal{I}$ and bundle set $\mathcal{B}$ with their corresponding observed user-bundle, user-item, and bundle-item interaction matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ for training.*

Output: *The predicted user-bundle preference score matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{B}|}$ for all unobserved user-bundle pairs. $\mathbf{R}(u, b)$ is a real number in $(0, 1)$ indicating the probability of user u favors bundle b.*

Furthermore, we also study the problem of transfer bundle recommendation, where the goal is to learn a user-bundle encoder on the source domain and then apply the learned encoder on the target domain. The transfer bundle recommendation problem is formally defined as follows.

Problem 2. ***Transfer Bundle Recommendation:***

Given: *The user set $\mathcal{U}_s$, item set $\mathcal{I}_s$ and bundle set $\mathcal{B}_s$ with their corresponding observed user-bundle, user-item, and bundle-item interaction matrices $\mathbf{X}_s, \mathbf{Y}_s, \mathbf{Z}_s$ for training from the source domain. The user set $\mathcal{U}_t$, item set $\mathcal{I}_t$ and bundle set $\mathcal{B}_t$ with their corresponding observed user-bundle, user-item, and bundle-item interaction matrices $\mathbf{X}_t, \mathbf{Y}_t, \mathbf{Z}_t$ for test from the target domain.*

Output: *The predicted user-bundle preference score matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}_t|\times|\mathcal{B}_t|}$ for all of the unobserved user-bundle pairs in the target domain $\{\mathcal{U}_t, \mathcal{I}_t, \mathcal{B}_t\}$. $\mathbf{R}(u, b) \in (0, 1)$ indicating the probability of user $u \in \mathcal{U}_t$ favors bundle $b \in \mathcal{B}_t$.*

Note that the observed user-bundle pairs in the target domain are divided into training and testing splits where there is no overlap.

3 THE PROPOSED MODEL

The overall model pipeline is shown in Figure 1. There are four main stages: (S1) heterogeneous subgraph generation and feature initialization; (S2) relational graph neural propagation; (S3) information aggregation, and (S4) prediction.

3.1 Subgraph generation and Relational graph neural propagation

In this subsection, we introduce the subgraph generation process and the model structure. We use heterogeneous graph for the subgraph construction for user-bundle pairs. Compared to homogeneous graph, heterogeneous graph models different levels of information propagations between categories of nodes and edges.

The heterogeneous subgraph we construct is an directed graph $\mathcal{G}_s = \{\mathcal{V}, \mathcal{E}\}$. For each user-bundle pair, we generate one k-hop subgraph. We divide nodes $v \in \mathcal{V}$ into several types. For the centered user-bundle pair, we mark the centered user as 0 and the centered bundle as 1. Then, we assign types to other users, bundles, and items. For bundle, user and item in hop k_i, we assign categorical attributes $3k_i - 1, 3k_i, 3k_i + 1$ to them. The reason for categorizing in this way is for different hops the nodes have different impacts on the central pair. When $k = 1$, there are five types of nodes.

For edges, we model edges $e \in \mathcal{E}$ into six categories: user→bundle and bundle→user (if $\mathbf{X}(u, b) = 1$) , user→item and item→user (if $\mathbf{Y}(u, i) = 1$), bundle→item and item→bundle (if $\mathbf{Z}(b, i) = 1$).

Based on the above subgraph construction, a given user, bundle, or item may have different embeddings after training if they appear in different heterogeneous subgraphs. In other words, different subgraphs may have different embeddings for the same node, which is different from previous works. For notation simplicity, we denote the embeddings of the targeted bundle, user, and item as e_b, e_u, and e_i respectively in the following description. The dimension of e_b, e_u, and e_i is d.

We fuse type and feature to generate initial embeddings. We use one-hot encoding for node types to obtain the first part. We generate a Gaussian distribution vector for another part. The reason for using Gaussian distribution is similar to the idea of free embeddings from [13], which could serve as a regularization for the embedding learning. Then we concatenate two parts to form the initial embeddings.

Now we introduce the model structure and details of information propagation. Firstly, let us discuss information propagation from the bundle's perspective. Intuitively, items in a bundle have certain connections with each other. Furthermore, the customers who buy bundles could influence bundles' semantics. SuGeR captures this internal relationship between user and bundle. For bundles, there are two levels of propagation ($i \to b$, $u \to b$) as follows:

$$\mathbf{e}_{b,1}^{(\ell+1)} = \frac{1}{|\mathcal{N}_{i,b}|}\sigma\left(\mathbf{W}_1^{(\ell)}\left(\mathbf{e}_b^{(\ell)} + \tau\left(\{\mathbf{e}_i^{(\ell)} \mid i \in \mathcal{N}_{i,b}\}\right)\right) + \mathbf{c}_1^{(\ell)}\right)$$

$$\mathbf{e}_{b,2}^{(\ell+1)} = \frac{1}{|\mathcal{N}_{u,b}|}\sigma\left(\mathbf{W}_2^{(\ell)}\left(\mathbf{e}_u^{(\ell)} + \tau\left(\{\mathbf{e}_u^{(\ell)} \mid u \in \mathcal{N}_{u,b}\}\right)\right) + \mathbf{c}_2^{(\ell)}\right)$$

Here, $\mathbf{e}_{b,1}^{(\ell+1)}$ and $\mathbf{e}_{b,2}^{(\ell+1)}$ are the two parts of bundle embeddings in the $(\ell + 1)$th layer, and $W_1^{(\ell)}, W_2^{(\ell)}$, are the two updating weight matrices based on edge type in heterogeneous graph. The reason why there are two parts is we gather information from user and item nodes to refine embeddings. $\tau(\cdot)$ is the aggregation function, here we simply choose sum and $\sigma(\cdot)$ is *LeakyRelu*.

Secondly, we consider the user's perspective. The bundle can be regarded as several organized items, so the user's preference for one bundle is affected by items. For centered and non-centered bundle nodes, the information will be aggregated from the connected user and item. The user's decision on a bundle will not only be affected by the item itself but also by the extra information when items appear as a set. Another trick is that we can get bundle similarity by comparing the items of two bundles. The propagation layer generates similar embeddings of two similar bundles to recommend. To let SuGeR learn the semantic of item combination and similarity

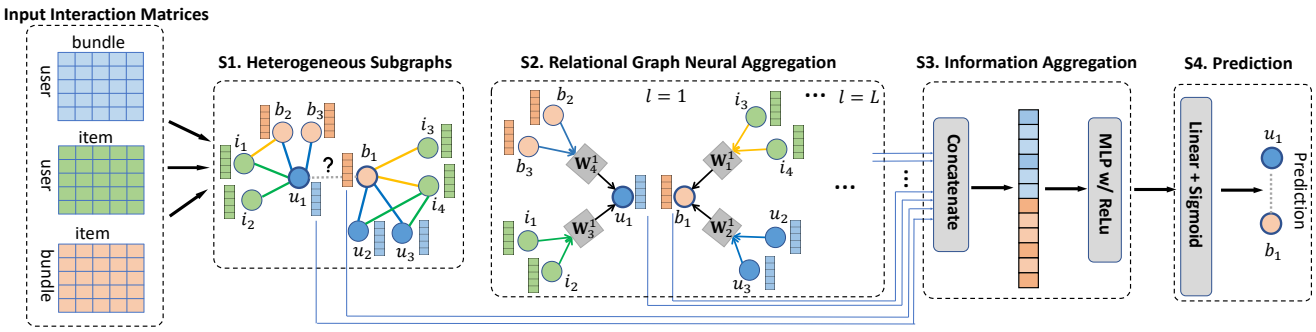

Figure 1: The overall pipeline of SuGeR model. Here we use the centered user-bundle pair (u_1, b_1) as an example, and we only show 1-hop heterogeneous subgraph. More details are presented in Section 3.

between bundles, we have the following two equations:

$$\mathbf{e}_{u,1}^{(\ell+1)} = \frac{1}{|\mathcal{N}_{i,u}|}\sigma\left(\mathbf{W}_3^{(\ell)}\left(\mathbf{e}_u^{(\ell)} + \tau\left(\{\mathbf{e}_i^{(\ell)} \mid i \in \mathcal{N}_{i,u}\}\right)\right) + \mathbf{c}_3^{(\ell)}\right)$$

$$\mathbf{e}_{u,2}^{(\ell+1)} = \frac{1}{|\mathcal{N}_{b,u}|}\sigma\left(\mathbf{W}_4^{(\ell)}\left(\mathbf{e}_u^{(\ell)} + \tau\left(\{\mathbf{e}_b^{(\ell)} \mid b \in \mathcal{N}_{b,u}\}\right)\right) + \mathbf{c}_4^{(\ell)}\right)$$

Here $\mathbf{e}_{u,1}^{(\ell+1)}$ and $\mathbf{e}_{u,2}^{(\ell+1)}$ are two parts of user embedding in the $(\ell+1)$-th layer, and $\mathbf{W}_3^{(\ell)}, \mathbf{W}_4^{(\ell)}$ are the two updating weight matrices for item→user and bundle→user edges. $\tau(\cdot)$ is the aggregation function, and here we choose summation. $\sigma(\cdot)$ is *LeakyRelu*.

3.2 Information Aggregation and Prediction

After obtaining $\mathbf{e}_{u,1}^{(\ell+1)}$, $\mathbf{e}_{u,2}^{(\ell+1)}$, $\mathbf{e}_{b,1}^{(\ell+1)}$ and $\mathbf{e}_{b,2}^{(\ell+1)}$, we average them to obtain $\mathbf{e}_u^{(\ell+1)}$ and $\mathbf{e}_b^{(\ell+1)}$. To utilize all the information, we concatenate all $\mathbf{e}_u^i$ and $\mathbf{e}_b^i$ from different layer to form $\mathbf{e}_u$ and $\mathbf{e}_b$. Then we concatenate these two vectors to get $\mathbf{e}_{sub}$, the graph-level embedding indicating the preference of user u on bundle b. We use Sigmoid function to get the final prediction, as shown in Figure 1.

3.3 Discussion

A - Label Leakage Issue. As proposed by [17] and [5], label leakage is a common but implicit issue on GNN models. Label leakage issue happens when predicting links by aggregating information from nearby nodes and the target link is also included in the aggregating function. In this case, the mapping learned by the model will have a self-mapping problem. Specifically, if we try to predict an edge q between target bundle b and target user u, then in every iteration bundle b aggregates information from u and user u also aggregates information from b. Therefore, the model tends to learn a self mapping $f_\theta(q, ...) = q$ [5]. This is not an ideal mapping function, since the information of target bundle b and target user u are lost as their embeddings are smoothed by each other, and the model tends to become overfitting. In our SuGeR model, this problem is even more critical since the core of SuGeR is to use user-bundle-centered subgraph to predict the center edge. Although we apply different weight matrices for user → bundle and bundle → user, this could not solve the problem thoroughly.

The key solution to this issue is to ensure that the edge to be predicted will not be involved in the information propagation process. To achieve this, we delete edges between centered user and bundle when training the model to cut off such self-mapping loops and to ensure the information is not propagated between centered bundle b and centered user u when the model learns how to predict the edge $q = (b, u)$ itself.

B - Transfer Bundle Recommendation. One important contribution of this paper is that the proposed model enables transfer bundle recommendation. SuGeR generates k-hop subgraphs to learn the local graph pattern of a pair of bundle and user. In the graph context, all such locally centered subgraphs will have one edge with high probability and other edges with relatively low probability during prediction. By using heterogeneous subgraph, SuGeR tries to learn a subgraph encoder to map subgraphs into embedding space. Some similar graph structures might be shared by the data from different domains, which will be captured by the encoder. Furthermore, our encoder focuses on the local pattern near a centered user-bundle pair, while existing methods (e.g., BGCN [3]) focus on learning a global embedding of users/bundles when predicting a particular user-bundle. Under the transfer context, it might not be reasonable for the encoder to learn the whole graph pattern which is dependent on the dataset. No side information such as ages and genders is used in our model. We try to use only domain-independent information. We regard this as an important contribution of this work because it has significant impact on the practical application.

4 EXPERIMENTS

In this section, we conduct experiments to verify the effectiveness and transferability of SuGeR model.

4.1 Experimental Setting and Dataset Statistics

Dataset Statistics. The datasets we use in this paper are Netease and Youshu, and the statistics can be found in [4].

Baselines. We use four baselines for the overall performance experiment, they are **GCN-BG** [11], **NGCF-BG** [12], **DAM** [4] and **BGCN** [2]. For **GCN** [11] and **NGCF** [12], both have two implementations ((Bipar-Graph and Tripar-Graph)) and we use user-bundle bipartite graph to train and test the performance. For these four baselines, **GCN-BG** [11] and **NGCF-BG** [12] are GNN models for traditional recommendation, and are not designed specifically for bundle recommendation. **DAM** [4] and **BGCN** [2] are newly published bundle-oriented GNN models.

Table 1: Performance comparison with four baselines and two model variants for ablation study.

Model	Youshu						Netease					
	Recall@20	Recall@40	Recall@80	NDCG@20	NDCG@40	NDCG@80	Recall@20	Recall@40	Recall@80	NDCG@20	NDCG@40	NDCG@80
NGCF-BG	0.1343	0.2076	0.2447	0.0618	0.0925	0.1420	0.0551	0.0927	0.1356	0.0265	0.0464	0.0768
GCN-BG	0.1431	0.2191	0.2432	0.0739	0.1137	0.1652	0.0567	0.0982	0.1439	0.0292	0.0501	0.0799
DAM	0.1670	0.2086	0.2918	0.0757	0.0939	0.1532	0.0747	0.1290	0.1695	0.0351	0.0593	0.0840
BGCN	0.2531	0.3674	0.4755	0.1623	0.1858	0.2432	0.1494	0.1825	0.2421	0.0786	0.1069	0.1428
SuGeR No subgraph	0.0301	0.0353	0.0469	0.0141	0.0176	0.0218	0.0315	0.0349	0.0392	0.0157	0.0173	0.0195
SuGeR Leakage	0.1301	0.2041	0.2886	0.0637	0.0912	0.1475	0.0692	0.1345	0.1688	0.0349	0.0621	0.0837
SuGeR	**0.3529**	**0.5438**	**0.6682**	**0.2041**	**0.2961**	**0.3960**	**0.2835**	**0.3539**	**0.4301**	**0.1361**	**0.1894**	**0.2359**
%Improv.	39.43%	48.01%	40.53%	25.75 %	59.36 %	62.83 %	89.76%	93.92 %	77.65%	73.16%	77.17%	65.20 %

Table 2: Transfer bundle recommendation experimental results.

Model	Youshu						Netease					
	Recall@20	Recall@40	Recall@80	NDCG@20	NDCG@40	NDCG@80	Recall@20	Recall@40	Recall@80	NDCG@20	NDCG@40	NDCG@80
GCN-BG	0.0611	0.0925	0.1233	0.0305	0.0474	0.0735	0.0244	0.0415	0.0639	0.0121	0.0207	0.0348
BGCN	0.1801	0.2809	0.3476	0.1354	0.1419	**0.1876**	**0.1061**	0.1496	0.1819	**0.0559**	0.0848	0.1014
SuGeR	**0.3107**	**0.3926**	**0.4255**	**0.1537**	**0.1725**	0.1801	0.0955	**0.1706**	**0.2302**	0.0408	**0.0850**	**0.1374**

Metric. We use two widely used metrics *Recall@K* [12] and *NDCG@K* [8] to test the performance. *Recall@K* measures the ratio of real positive ones in top-K bundles. *NDCG@K* considers the rank position and gives higher scores to the real positive bundle at a higher rank. *NDCG@K* is a more rigorous metric.

Experimental Setting. For all experiments, we split two datasets into training and test sets. We split the positive user-bundle interactions into two parts, so during training, the model cannot use the positive u-b pair from the test split. 60% of the user-bundle interactions are in the training set. We use random negative sampling for training to ensure the number of negative triplets is the same as the number of positive samples in the training set.

4.2 Overall Performance Comparison

We conduct experiments on two benchmark datasets with four baselines and two metrics as listed above. The metric is the same as [2]. The overall performance is shown in Table 1.

From the result, we can see that SuGeR model outperforms all the baselines with a significant improvement when using *Recall@K* and *NDCG@K* as the metrics. SuGeR could achieve an average of 45% improvement on Youshu and an average of 79% improvement on Netease when compared to the state-of-the-art model **BGCN** [2]. That proves the effectiveness of our user-bundle-centered subgraph-based GNN model.

We conduct an ablation study on the usage of the heterogeneous subgraph and the prevention method of label leakage as shown in the last 2 rows in Table 1. *No subgraph* denotes that we do not generate subgraphs and only initialize embeddings. *Leakage* means we do not delete target edges during training. As listed in the table, the performance becomes worse in these two cases, which shows the effectiveness of both using heterogeneous subgraphs and label leakage prevention.

4.3 Transfer Bundle Recommendation Experiments

We first train two models on the training sets of Youshu and Netease, then conduct test experiment on the test sets of Youshu using model trained on Netease, and test on the test set of Netease using model trained on Youshu. We adopt the same setting on **BGCN**[2] and **NGCF**[12]. The result is shown in Table 2. We can observe that SuGeR has the best transfer performance compared to the two baselines. Another observation is that the performance of the transfer setting still decreases compared with the performance in Table 1. These experimental results follow our intuition that the subgraph-based bundle recommendation model we propose has strong transferability, which is elaborated in Section 3.1. The result shows that the model is able to learn the internal purchasing logic, which is not platform-dependent. As a result, the heterogeneous subgraph we generate could contain some platform-independent patterns.

5 CONCLUSION

In this paper, we propose a subgraph-based GNN model to handle both the basic bundle recommendation and transfer bundle recommendation problem. The model uses three interaction matrices as input to generate k-hop heterogeneous user-bundle centered subgraph to learn the embeddings for target bundles and users. The model has strong transferability when facing unseen domains. We also propose a solution for the implicit label leakage issue in our model to avoid the self-mapping problem. Extensive experiments demonstrate the significant improvement of the effectiveness and transferability of our model over all the baseline models. Future directions include further studying the knowledge transfer in recommender system, and applying our approach to cross-domain bundle recommendation scenario [9, 15].

ACKNOWLEDGMENTS

This work is supported by NSF (1947135,and 2134079), the NSF Program on Fairness in AI in collaboration with Amazon (1939725), DARPA (HR001121C0165),NIFA (2020-67021-32799), and ARO (W911NF2110088). The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

[1] Jinze Bai, Chang Zhou, Junshuai Song, Xiaoru Qu, Weiting An, Zhao Li, and Jun Gao. 2019. Personalized bundle list recommendation. In *The World Wide Web Conference*. 60–71.

[2] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2020. Bundle Recommendation with Graph Convolutional Networks. In *Proceedings of the 43nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Xi'an, China, July 25-30, 2020*.

[3] Jianxin Chang, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. 2021. Bundle Recommendation and Generation with Graph Neural Networks. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[4] Liang Chen, Yang Liu, Xiangnan He, Lianli Gao, and Zibin Zheng. 2019. Matching User with Item Set: Collaborative Bundle Recommendation with Deep Attention Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2095–2101.

[5] Qilin Deng, Kai Wang, Minghao Zhao, Zhene Zou, Runze Wu, Jianrong Tao, Changjie Fan, and Liang Chen. 2020. Personalized Bundle Recommendation in Online Games. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2381–2388.

[6] Boxin Du, Changhe Yuan, Robert Barton, Tal Neiman, and Hanghang Tong. 2021. Hypergraph Pre-training with Graph Neural Networks. *arXiv preprint arXiv:2105.10862* (2021).

[7] Boxin Du, Si Zhang, Yuchen Yan, and Hanghang Tong. 2021. New Frontiers of Multi-Network Mining: Recent Developments and Future Trend. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 4038–4039.

[8] Mohsen Jamali and Martin Ester. 2009. Using a Trust Network to Improve Top-N Recommendation. In *Proceedings of the Third ACM Conference on Recommender Systems*. Association for Computing Machinery, 181–188.

[9] Meng Jiang, Peng Cui, Xumin Chen, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2015. Social recommendation with cross-domain transferable knowledge. *IEEE transactions on knowledge and data engineering* 27, 11 (2015), 3084–3097.

[10] Baoyu Jing, Yuchen Yan, Yada Zhu, and Hanghang Tong. 2022. COIN: Co-Cluster Infomax for Bipartite Graphs. *arXiv preprint arXiv:2206.00006* (2022).

[11] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *arXiv preprint arXiv:1706.02263* (2017).

[12] 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*. Association for Computing Machinery, 165–174.

[13] Le Wu, Junwei Li, Peijie Sun, Richang Hong, Yong Ge, and Meng Wang. 2020. Diffnet++: A neural influence and interest diffusion network for social recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[14] Yikun Xian, Tong Zhao, Jin Li, Jim Chan, Andrey Kan, Jun Ma, Xin Luna Dong, Christos Faloutsos, George Karypis, Shan Muthukrishnan, et al. 2021. Ex3: Explainable attribute-aware item-set recommendations. In *Fifteenth ACM Conference on Recommender Systems*. 484–494.

[15] Yuchen Yan, Si Zhang, and Hanghang Tong. 2021. Bright: A bridging algorithm for network alignment. In *Proceedings of the Web Conference 2021*. 3907–3917.

[16] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 269–277.

[17] Jiani Zhang, Xingjian Shi, Shenglin Zhao, and Irwin King. 2019. Stacked and reconstructed graph convolutional networks for recommender systems. *28th International Joint Conference on Artificial Intelligence*, 4264–4270.

[18] Muhan Zhang and Yixin Chen. 2019. Inductive matrix completion based on graph neural networks. *arXiv preprint arXiv:1904.12058* (2019).