Computing and Informatics, Vol. 43, 2024, 687–708, doi: 10.31577/cai_2024_3_687

MGRF: MULTI-GRAPH RECOMMENDATION FRAMEWORK WITH HETEROGENEOUS AND HOMOGENEOUS GRAPH ITERATIVE FUSION

Xiang LIN

School of Information Engineering
Xuzhou College of Industrial Technology
No. 1, Xiangwang Road, Xuzhou, Jiangsu, 221140, China
&
School of Computer Science and Technology
China University of Mining and Technology
No. 1, Daxue Road, Xuzhou, Jiangsu, 221116, China
e-mail: l_n@cumt.edu.cn

Fangyu HAN, Xiaobin RUI, Chengcheng SUN, Zhixiao WANG*

School of Computer Science and Technology
China University of Mining and Technology
No. 1, Daxue Road, Xuzhou, Jiangsu, 221116, China
e-mail: zhxwang@cumt.edu.cn

Lijun YAN

College of Information, Mechanical and Electrical Engineering
Shanghai Normal University
No. 100, Guilin Road, Xuhui, Shanghai, 200234, China
e-mail: flying@shnu.edu.cn

Abstract. With the development of deep learning, deep neural methods have been introduced to boost the performance of Collaborative Filtering (CF) models. How-

* Corresponding author

ever, most of the models rely solely on the user-item heterogeneous graph and only implicitly capture homogenous information, which limits their performance improvement. Although some state-of-the-art methods try to utilize additional graphs to make up, they either merely aggregate the information of multiple graphs in the step of initial embedding or only merge different multi-graph information in the step of final embedding. Such one-time multi-graph integration leads to the loss of interactive and topological information in the intermediate process of propagation. This paper proposes a novel Multi-Graph iterative fusion Recommendation Framework (MGRF) for CF recommendation. The core components are dual information crossing interaction and multi-graph fusing propagation. The former enables repeated feature crossing between heterogeneous nodes throughout the whole embedding process. The latter repeatedly integrates homogeneous nodes as well as their topological relationships based on the constructed user-user and item-item graphs. Thus, MGRF can improve the embedding quality by iteratively fusing user-item heterogeneous graph, user-user and item-item homogeneous graphs. Extensive experiments on three public benchmarks demonstrate the effectiveness of MGRF, which outperforms state-of-the-art baselines in terms of Recall and NDCG.

Keywords: Recommender systems, multi-graph fusion, graph neural networks, embedding propagation

Mathematics Subject Classification 2010: 68-T99

1 INTRODUCTION

In the era of information explosion, recommender systems play a vital role in struggling with information overload [1]. They help users to discover items of interest, which are suitable for many online services, including travelling [2], news feeding [3], and online shopping [4]. Hence, recommender systems have attracted great attention in both industry [4] and academia [5].

Collaborative filtering (CF), as one of the most influential and widely used recommendation methods, has emerged to expeditiously filter out the items that users are interested in and accurately capture user preferences [6]. The common paradigm of CF is to parameterize users and items by learning vector representations (a.k.a. embeddings) from historical interactions data (e.g., ratings and clicks), and perform prediction based on the pairwise similarity of embedding vectors between users and items [7].

With the development of deep learning, deep neural methods have been introduced to boost the performance of CF models, such as Deep Crossing [8], Wide & Deep [9] and DeepFM [10]. Deep neural methods are more expressive to learn complex non-linear relationships and mine hidden patterns between users and items [11]. Graph is a natural struct for representing rich pairwise relationship in recommendation [12], due to its powerful capability of learning representation from non-Euclidean structure data [13, 14]. Therefore, GCN-based models are adopted to exploit multi-hop information from user-item interactions [14]. The key idea of GCNs is iteratively aggregating feature information from graph neighborhood, which can capture and aggregate the high-order information of user-item bipartite graph, thereby improving the accuracy of user and item embeddings [15]. Typical GCN-based models include GC-MC [16], PinSage [17], NGCF [18], LR-GCCF [19], LightGCN [20], DGCF [21] and UltraGCN [22]. Neural Graph Collaborative Filtering (NGCF) [18] propagates the embeddings of users and items via utilizing multiple GCN layers to capture high-order connectivity. LR-GCCF [19] removes non-linear activation and LightGCN [20] takes a further step that removes all transformation parameters and activation function in the convolutional layer, which greatly improves the performance. DGCF [21] models the user's latent intention distribution on each pair of user-item interactions and maps the latent intention to form a separate representation. UltraGCN [22] directly approximates the convergence state to avoid over-smoothing problem of multi-layer message propagation.

All these methods are based on user-item heterogeneous graph, which can indeed provide interactive information between users and items for CF recommendation. However, those methods do not involve user-user or item-item homogeneous graphs. In fact, these two kinds of graphs also contain meaningful topology information for CF recommendation, which reveals the behavioral similarity between users (or items). Take Figure 1 as a toy example. The *book* was purchased by *user A*, *B*, *C* and *D*. Meanwhile, *user B*, *C* and *D* purchased *computer*. Thus, from the perspective of user-item heterogeneous graph, *user A* may be interested in *computer* and we will recommend *computer* first to *user A*. However, just with user-item heterogeneous graph, we cannot effectively rank *watch*, *ruler* and *phone* for the recommendation to *user A*. The reason is that, according to user-based CF, the interest scores of the three items are the same for *user A*.

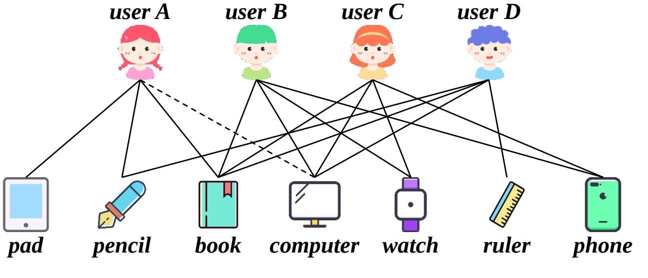


Figure 1. A toy example

In fact, a user-user homogeneous graph can be constructed from user-item heterogeneous graph according to the following strategy: If there is a *user i*–*item k*–*user j* path in user-item heterogeneous graph, then there will be an edge between *user i* and *user j* in user-user homogeneous graph. The corresponding user-user graph is shown in Figure 2, where the weight of an edge represents the interest similarity between the corresponding two users. The interest similarity can be calculated by the similarity of actions [23]. Just for example, we simply calculate the interest similarity of *user A* and *user B* by the following Cosine similarity formula,

$$\text{similarity}(A, B) = \text{COS}\langle A, B\rangle = \frac{|N(A) \cap N(B)|}{\sqrt{|N(A)||N(B)|}} = \frac{1}{\sqrt{3 * 4}} = \mathit{0.289},$$

where $N(\cdot)$ denotes the collection of purchased items. Similarly, the similarity(A, C) and similarity(A, D) can be obtained, which are 0.289 and 0.577, respectively.

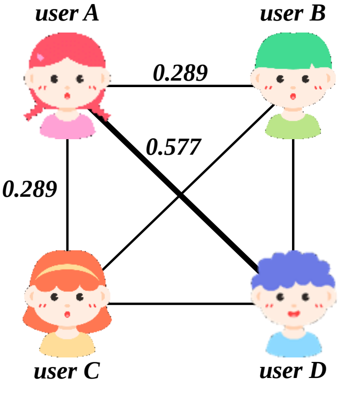


Figure 2. User-user homogeneous graph

Figure 2 shows that *user A* and *user D* are more similar, which implies the potential interest of *user A* and *user D* are more similar. Based on this additional social information, we will preferentially recommend *ruler* to *user A*, rather than *watch* and *phone*. In this way, the above problem of differentiating *watch*, *ruler* and *phone* for the recommendation to *user A* can be solved.

This example shows that merely relying on the user-item heterogeneous graph cannot ensure effective recommendation in some scenarios. Although two-hop neighborhoods in user-item heterogeneous graph can implicitly capture homogeneous information to some extent by those methods, it is reasonable to believe that the performance of CF recommendation can be improved by integrated learning from explicitly encoded homogeneous graphs. The experimental results of Multi-GCCF [12] and FBNE [24] confirmed our assertion. However, Multi-GCCF only merges different multi-graph information in the step of final embedding, FBNE merely aggregates the information from related nodes within multiple bipartite graphs in the step of initial embedding. Such one-time multi-graph integration leads to the loss of interactive and topological information in the intermediate process of propagation, resulting in limited performance improvement. Therefore, how to fully fuse and leverage multi graphs becomes very critical.

Fundamentally different from the one-time multi-graph integration in Multi-GCCF [12] and FBNE [24], this paper presents a novel multi-graph recommendation framework (named MGRF) for CF by iteratively fusing user-item heterogeneous graph, user-user and item-item homogeneous graphs. Heterogeneous graph contains rich interactive information while homogeneous graphs provide meaningful topological information for CF recommendation. The main contributions of this paper are summarized as follows:

- We propose a multi-graph recommendation framework with heterogeneous and homogeneous graph **iterative fusion**, which effectively leverages crossing interaction between user and item (from heterogeneous graph), and repeatedly integrates the inner high-order connectivity of users and items (from homogeneous graph), thereby embedding the crossing interaction and features of users and items in high quality.
- We design the dual information crossing interaction and multi-graph fusing propagation for MGRF. The former employs user-item interactions in user-item heterogeneous graph to explore the interaction information for crossing propagation throughout the whole embedding process; The latter utilizes user-item heterogeneous graph to directly construct user-user and item-item graphs, then repeatedly integrates homogeneous nodes (users or items) and their topological relationships. Thus, MGRF cannot only guarantee crossing interaction between heterogeneous nodes, but also explicitly encode the topological information between homogeneous nodes.
- We perform extensive experiments on three real-world datasets that all comprise more than one million user-item interactions. The experimental results demonstrate the effectiveness of our proposed MGRF. It can effectively integrate heterogeneous and homogeneous graphs to improve the embedding quality of the recommendation model, outperforming the state-of-the-art baselines in terms of Recall and NDCG.

2 METHODOLOGY

In this section, we present the Multi-Graph Recommendation Framework (MGRF) with heterogeneous and homogeneous graph iterative fusion. Heterogeneous graph contains rich interactive information while homogeneous graphs provide meaningful topological information for CF recommendation.

As Figure 3 shows, the proposed framework consists of three main layers. *Firstly*, the embedding layer initializes user embeddings and item embeddings. *Secondly*, the propagation layer iteratively fuses interactive information between heterogeneous nodes and topological information between homogeneous nodes to refine initial user and item embeddings. *Thirdly*, the prediction layer aggregates the refined embeddings and residual connections as final representations, and then outputs the affinity score of each user-item pair for the top-N recommendation.

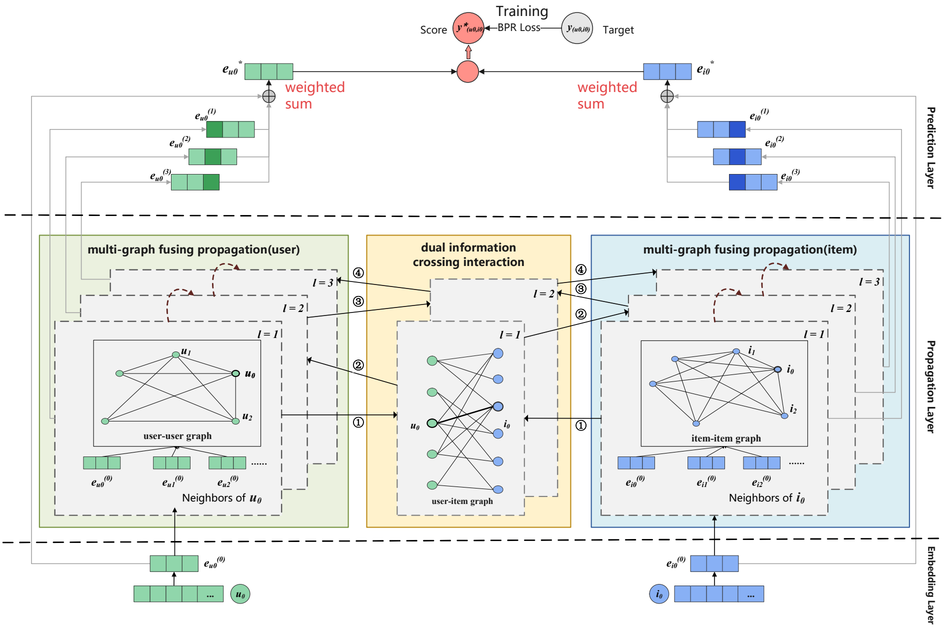


Figure 3. The overall architecture of MGRF (the arrows present the flow of information). The representations of user u_0 (left) and item i_0 (right) are initialized in embedding layer, then refined in propagation layer whose outputs are finally combined in prediction layer for top-N recommendation. In the propagation layer, the dual information crossing interaction (left, right) and multi-graph fusing propagation (middle) iteratively cross information between the user-item heterogeneous graph and two homogeneous graphs (i.e. user-user graph and item-item graph) according to the order of arrows.

2.1 Embedding Layer

In a typical recommendation scenario, user IDs and item IDs are usually encoded as one-hot vectors. Suppose there are n users and m items, the sets of user and item are $\{\boldsymbol{x_{u_1}}, \boldsymbol{x_{u_2}}, \dots, \boldsymbol{x_{u_n}}\}$ and $\{\boldsymbol{x_{i_1}}, \boldsymbol{x_{i_2}}, \dots, \boldsymbol{x_{i_m}}\}$, respectively. In the embedding layer, MGRF transforms those user IDs and item IDs into embedding vectors $\boldsymbol{e_u} \in R^d$ and $\boldsymbol{e_i} \in R^d$ following mainstream models [7, 18]:

$$\boldsymbol{e_u} = \boldsymbol{E} \cdot \boldsymbol{x_u};\ \boldsymbol{e_i} = \boldsymbol{E} \cdot \boldsymbol{x_i}, \tag{1}$$

where $\boldsymbol{E} \in R^{(n+m)*d}$ refers to the initial embedding vectors for users and items, d denotes the size of embeddings, $\boldsymbol{x_u}$ and $\boldsymbol{x_i}$ represent user ID, item ID, respectively. The initial user and item embeddings of this layer will be optimized in an end-to-end fashion [18].

2.2 Propagation Layer

Propagation layer ultilizes the message-passing architecture of convolution neural networks to iteratively capture CF information from different graph structures and refine the initial embeddings of users and items. This propagation layer consists of two key components: dual information crossing interaction and multi-graph fusing propagation. The former is designed to minimize the loss of fine-grained features in propagation and guarantee the iterative crossing interaction between heterogeneous nodes. The latter is devised to highlight the intrinsic differences of users (items), and explicitly encode the topological information between homogeneous nodes. The arrows marked ①②③④ in Figure 3 describe the collaborative process of the two components, which is detailed as follows. The initial user embeddings and item embeddings in the embedding layer serve as inputs for the first iteration of the propagation layer. These information is aggregated on two homogenous graphs (i.e. user-user graph and item-item graph) separately, then both sent to the user-item graph for further crossing, as shown in the arrow marked ①. After dual information crossing interaction receives both user and item embeddings from the multi-graph fusing propagation, it crosses them on the heterogeneous user-item graph. The crossed embeddings will serve as the input for the next iteration of multi-graph fusing propagation, as shown in the arrow marked ②. The crossed embeddings from user-item heterogeneous graph as well as the uncrossed embeddings from homogenous graphs (i.e. user-user graph and item-item graph) are all sent to the multi-graph fusion propagation to respectively update the representations of users and items for the next iteration. The propagation layer repeats the above operations to iteratively fuse heterogeneous and homogenous graphs until the optimal embeddings are obtained.

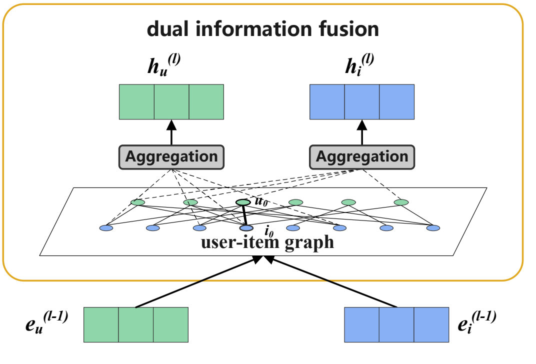


Figure 4. Dual information crossing interaction in the user-item graph

2.2.1 Dual Information Crossing Interaction

Interactive information between items and users is crucial for recommendation. Therefore, MGRF designs a component named dual information crossing interaction, which adopts GCN for repeated feature crossing in user-item bipartite graph. Dual information crossing interaction propagates forward to capture collaborative information between heterogeneous nodes, and minimize the loss of fine-grained feature information. The basic idea of GCN is to learn node representations by smoothing features over the graph, which updates the node representations in the $(l+1)^{\text{th}}$ iteration by normalizing and aggregating the representations of their neighbors in l^{th} iteration. The representations of user u and item i in the $(l+1)^{\text{th}}$ iteration can be respectively formulated as follows:

$$\boldsymbol{h}_{\boldsymbol{u}}^{(\boldsymbol{l}+\boldsymbol{1})} = \sigma\left(\boldsymbol{W}_{\boldsymbol{u}}^{(\boldsymbol{l})} \cdot \left[\boldsymbol{h}_{\boldsymbol{u}}^{(\boldsymbol{l})}; \boldsymbol{h}_{\boldsymbol{N}_{(\boldsymbol{u})}}^{(\boldsymbol{l})}\right]\right), \quad \boldsymbol{h}_{\boldsymbol{u}}^{\boldsymbol{0}} = \boldsymbol{e}_{\boldsymbol{u}}, \tag{2}$$

$$\boldsymbol{h}_{\boldsymbol{i}}^{(\boldsymbol{l}+\boldsymbol{1})} = \sigma\left(\boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{l})} \cdot \left[\boldsymbol{h}_{\boldsymbol{i}}^{(\boldsymbol{l})}; \boldsymbol{h}_{\boldsymbol{N}_{(\boldsymbol{i})}}^{(\boldsymbol{l})}\right]\right), \quad \boldsymbol{h}_{\boldsymbol{i}}^{\boldsymbol{0}} = \boldsymbol{e}_{\boldsymbol{i}}, \tag{3}$$

where $[;]$ represents concatenation, $\boldsymbol{W}_{\boldsymbol{u}}^{(\boldsymbol{l})}$ and $\boldsymbol{W}_{\boldsymbol{i}}^{(\boldsymbol{l})}$ refer to the user and item transformation weight matrix of $(l+1)^{\text{th}}$ iteration, $\boldsymbol{h}_{\boldsymbol{N}_{(\boldsymbol{u})}}^{(\boldsymbol{l})}$ and $\boldsymbol{h}_{\boldsymbol{N}_{(\boldsymbol{i})}}^{(\boldsymbol{l})}$ denote the learned neighborhood embeddings. $\sigma(\cdot)$ is the tanh activation function.

As Figure 4 shows, the dual information crossing interaction first takes user embedding $\boldsymbol{e}_{\boldsymbol{u}}^{(\boldsymbol{l}-\boldsymbol{1})}$ and item embedding $\boldsymbol{e}_{\boldsymbol{i}}^{(\boldsymbol{l}-\boldsymbol{1})}$ as input, which is from multi-graph fusing propagation in the $(l-1)^{\text{th}}$ iteration. Then, it exploits the user-item graph to cross features and integrate user-item interactions into the embedding function. Finally, it aggregates the propagated messages to obtain the fused embeddings with cross information. The output $\boldsymbol{h}_{\boldsymbol{u}}^{(\boldsymbol{l})}$ and $\boldsymbol{h}_{\boldsymbol{i}}^{(\boldsymbol{l})}$ will serve as the input for multi-graph fusing propagation in the following l^{th} iteration.

When aggregating user-item interactions, MGRF applies element-wise weighted mean aggregator to achieve permutation invariance in the neighborhood for a user u, namely

$$\boldsymbol{h}_{\boldsymbol{N}_{(\boldsymbol{u})}}^{(\boldsymbol{l})} = \sigma\left(\text{MEAN}\left(\left\{\boldsymbol{e}_{\boldsymbol{i}}^{(\boldsymbol{l}-\boldsymbol{1})} \cdot \boldsymbol{Q}_{\boldsymbol{u}}^{(\boldsymbol{l})}, i \in N(u)\right\}\right)\right), \tag{4}$$

where $\boldsymbol{Q}_{\boldsymbol{u}}^{(\boldsymbol{l})}$ refers to the user aggregator weight matrix in the l^{th} iteration, which is trainable and shared across all users in l^{th} iteration, $N(u)$ denotes the set of neighbor items of user u in the user-item graph, and $\text{MEAN}(\cdot)$ represents the mean of the vectors. Similarly, the embedding of a target item i can be generated by using another set of user transformation and aggregator weight matrices:

$$\boldsymbol{h}_{\boldsymbol{N}_{(\boldsymbol{i})}}^{(\boldsymbol{l})} = \sigma\left(\text{MEAN}\left(\left\{\boldsymbol{e}_{\boldsymbol{u}}^{(\boldsymbol{l}-\boldsymbol{1})} \cdot \boldsymbol{Q}_{\boldsymbol{i}}^{(\boldsymbol{l})}, u \in N(i)\right\}\right)\right), \tag{5}$$

where $\boldsymbol{Q}_{\boldsymbol{i}}^{(\boldsymbol{l})}$ refers to the item aggregator weight matrix in the l^{th} iteration, which is trainable and shared across all items in l^{th} iteration, $N(i)$ denotes the set of neighbor users of item i in the user-item graph.

With the dual information crossing interaction, the propagation layer can integrate heterogeneous information on the opposite side for message propagation. Consequently, MGRF cannot only mimimize the loss of fine-grained feature information in propagation but also guarantee the crossing interaction between heterogeneous nodes, i.e., users and items.

2.2.2 Multi-Graph Fusing Propagation

User-user and item-item relationships are essential homogeneous information for recommendation, which reveals the behavioral similarity between users (or items). Therefore, MGRF designs a component named multi-graph fusing propagation, which utilizes the user-item bipartite graph to construct user-user and item-item graphs and then iteratively learns embeddings from multi-graphs, including user-item heterogeneous graph, user-user, and item-item homogeneous graphs. As shown in Figure 5, the multi-graph fusion propagation consists of three phases: **forward sampling**, **iterative aggregation**, and **information fusion**.

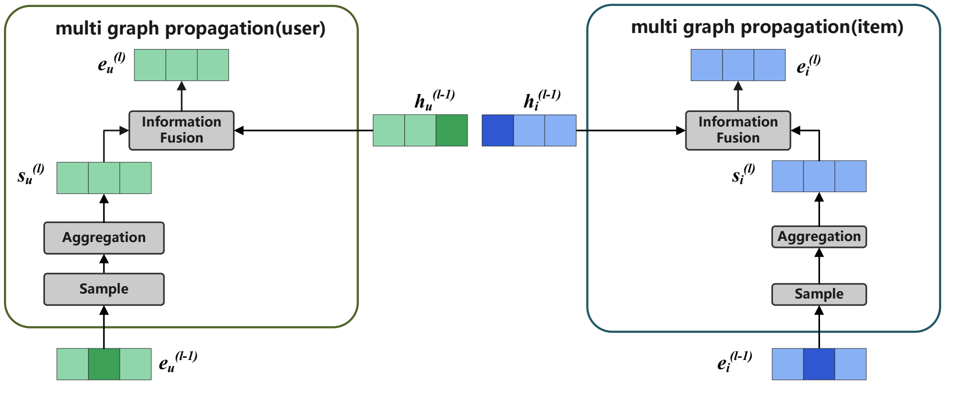


Figure 5. Multi-graph fusion propagation

Forward Sampling. CF is based on the assumption that similar users would exhibit similar preferences on items. Therefore, in the forward sampling phase, we construct user-user (or item-item) graphs based on the similarities between users (or items). Meanwhile, samples of similar nodes within the pre-top K list are associated in homogeneous graphs. In this paper, we attenuate the effect of popular items in the common interest list on their similarity. Specifically, we measure similarity s_{pq} between two homogeneous nodes p, q as:

$$s_{pq} = \frac{\sum_{\alpha \in N(p) \cap N(q)} \frac{1}{\ln(1+|N(\alpha)|)}}{\sqrt{|N(p)||N(q)|}}, \tag{6}$$

where $N(p)$ and $N(q)$ denote the set of neighbors of nodes p and q in the user-item graph, respectively, α refers to the common neighbor of nodes p and q, $|N(\alpha)|$ indicates the degree of denotes node α. Computing all pairwise cosine similarities is expensive. Hence, some pairs should be pruned. Referring to the existing method [12] that build user-user or item-item graphs based on all adjacent interaction relationships, MGRF takes the pre-top K neighbors as valid connections for user u (item i) to decrease the difficulty and training cost. In Section 3.5.1, a study of the effect of the sample number K on the performance for processing examples is provided.

Iterative Aggregation. The iterative aggregation phase follows the idea of graph convolutional network [18] to reveal latent information. For a connected user-user pair $\langle u_1, u_2\rangle$ (or item-item pair $\langle i_1, i_2\rangle$), MGRF defines the message from u_2 to u_1 (or i_2 to i_1) as:

$$f\left(\boldsymbol{e_{u_1}}, \boldsymbol{e_{u_2}}\right) = \frac{1}{\sqrt{\left|N_s'\left(u_1\right)\right|\left|N_s'\left(u_2\right)\right|}}\boldsymbol{e_{u_2}}, \tag{7}$$

$$f\left(\boldsymbol{e_{i_1}}, \boldsymbol{e_{i_2}}\right) = \frac{1}{\sqrt{\left|N_s'\left(i_1\right)\right|\left|N_s'\left(i_2\right)\right|}}\boldsymbol{e_{i_2}}, \tag{8}$$

where $\boldsymbol{e_{u_1\leftarrow u_2}}$ and $\boldsymbol{e_{i_1\leftarrow i_2}}$ denote the user and item message (i.e., information to be propagated) embeddings, and $N_s'(\cdot)$ denotes the one-hop neighborhood in the corresponding homogeneous graph. $f(\cdot)$ refers to the message encoding function, which takes embeddings of u_1 and u_2 as input. The backward aggregation phase will further generate user (item) embeddings by aggregating neighborhood features through a one-hop graph convolution layer and a sum aggregator:

$$\boldsymbol{S_u^{(l)}} = \sum_{v\in N_s'(u)} \frac{1}{\sqrt{|N_s'(u)|\,|N_s'(v)|}}\boldsymbol{e_v^{(l-1)}}, \tag{9}$$

$$\boldsymbol{S_i^{(l)}} = \sum_{j\in N_s'(i)} \frac{1}{\sqrt{|N_s'(i)|\,|N_s'(j)|}}\boldsymbol{e_j^{(l-1)}}. \tag{10}$$

Information Fusion. The information fusion phase combines user (item) embeddings from heterogeneous and homogeneous graphs by introducing the element-wise sum:

$$\boldsymbol{e_u^{(l)}} = \boldsymbol{S_u^{(l)}} + \boldsymbol{h_u^{(l-1)}};\ \boldsymbol{e_i^{(l)}} = \boldsymbol{S_i^{(l)}} + \boldsymbol{h_i^{(l-1)}}. \tag{11}$$

With representations augmented by first-order connectivity modeling, we can stack more multi-graph fusion propagation to explore the high-order connectivity information. Its excellent performance will be verified in Section 3.5.2 by comparing it with other alternatives such as concatenation and attention aggregators. Such high-order connectivities are crucial to encoding the collaborative information and estimating the relevance score between a user and an item. In this way, MGRF cannot only highlight the intrinsic differences between users and items but also explicitly encode the topological information between homogeneous nodes.

With dual information crossing interaction and multi-graph fusing propagation, MGRF propagates messages alternately across heterogeneous and homogeneous graphs and iteratively refines both user and item embeddings.

2.3 Prediction Layer

By stacking embeddings from the propagation layer, a user (or an item) is capable of receiving the messages propagated from its l-hop neighbors. MGRF adopts the holistic connection that combines the embeddings from multi-graph fusing propagation of all the iterations (including the initial embedding) as a node's final feature representation. Since embeddings from different iterations of multi-graph fusing propagation contain information of different receptive fields, combining these embeddings is more informative and also effective for alleviating the over-smoothing problem. The holistic connection is formulated as:

$$\boldsymbol{e_u^*} = \frac{1}{L+1}\sum_{l=0}^{L}\boldsymbol{e_u^{(l)}}, \tag{12}$$

$$\boldsymbol{e_i^*} = \frac{1}{L+1}\sum_{l=0}^{L}\boldsymbol{e_i^{(l)}}, \tag{13}$$

where $\boldsymbol{e_u^*}$, $\boldsymbol{e_i^*}$ denote the final embedding of user u and item i, respectively. Here we use an element-wise average aggregator as the information fusion strategy for the prediction layer. When composing the final embedding, different iterations of the embedding have different importance and thus can be considered as manually tuned hyper-parameters, or model parameters for automatic optimization. However, existing studies [20] show that setting the importance uniformly to $1/(L+1)$ usually leads to good performance, to avoid complicating model unnecessarily and to keep its simplicity. Finally, we conduct the inner product $\hat{y}_{ui}$ as a score to estimate user preference towards the target item:

$$\hat{y}_{ui} = \boldsymbol{e_u^{*T}} \cdot \boldsymbol{e_i^*}, \tag{14}$$

which is used as the ranking score for top-N recommendation.

2.4 Optimization

To learn model parameters, MGRF optimizes the pairwise Bayesian Personalized Recommendation (BPR) loss [25], which has been intensively used in recommender systems. BPR assumes that observed interactions, which are more reflective of user preferences, should be given higher values than non-observed interactions. We adapt our model to forward and backward propagation for mini-batches of triplet pairs $\{u, i, j\}$. To be more specific, we select unique user u and item $\{i, j\}$ from mini-batch pairs, and obtain low-dimensional embeddings of them after forward propagation. Then, the predicted scores of user u for positive sample i and negative sample j are calculated based on these embedding. Finally, the stochastic gradient descent method was used to minimize the BPR loss for optimizing the recommendation model. The objective function is as follows:

$$\text{loss}_{bpr} = \sum_{(u,i,j)\in\mathcal{O}} -\log\sigma\left(\boldsymbol{e}_{\boldsymbol{u}}^{*}\cdot\boldsymbol{e}_{\boldsymbol{i}}^{*}-\boldsymbol{e}_{\boldsymbol{u}}^{*}\cdot\boldsymbol{e}_{\boldsymbol{j}}^{*}\right)+\lambda\left(\left\|\boldsymbol{e}_{\boldsymbol{u}}^{*}\right\|_2^2+\left\|\boldsymbol{e}_{\boldsymbol{i}}^{*}\right\|_2^2+\left\|\boldsymbol{e}_{\boldsymbol{j}}^{*}\right\|_2^2\right), \quad (15)$$

where $\mathcal{O} = \{(u, i, j) \mid (u, i) \in R^+, (u, j) \in R^-\}$ denotes the training batch. R^+ indicates observed positive interactions. R^- indicates sampled unobserved negative interactions. We conduct regularization on both model parameters λ and generated embeddings to prevent overfitting.

3 EXPERIMENTS

In this section, we conduct extensive experiments on three widely used datasets to evaluate our recommendation framework and answer the following research questions.

RQ1: Compared with state-of-the-art models and frameworks, how does MGRF perform?

RQ2: How do embeddings benefit from the dual information crossing interaction and multi-graph fusing propagation of MGRF?

RQ3: How do different hyper-parameter settings (including sample number and information fusion method) affect the results of MGRF?

3.1 Dataset Description and Evaluation Metrics

Dataset	**Users**	**Items**	**Interactions**	**Density**
Gowalla	29 858	40 981	1 027 370	0.000840
Yelp2018	31 668	38 048	1 561 406	0.001296
Amazon-Book	52 643	91 599	2 984 108	0.000619

Table 1. Statistics of the datasets

We use three benchmark datasets [18, 20]: **Gowalla**, **Yelp2018**, and **Amazon-Book**, which are publicly accessible and released by baseline algorithms [22, 21, 20, 12]. The statistics of datasets are summarized in Table 1 with various domains, size and sparsity. In our experiments, two widely used metrics are adopted to evaluate top-N recommendations: Recall@k and NDCG@k (normalized discounted cumulative gain) [26]. Given that many state-of-the-art recommender models [7, 18, 20, 12] exhibit their performance with Recall@20 and NDCG@20, we compare framework with them and report the corresponding results of Recall@20 and NDCG@20.

3.2 Baseline Algorithms and Parameter Settings

We implement our MGRF framework in PyTorch. To demonstrate the effectiveness of our proposed MGRF, we compare it with various types of the state-of-the-art models, including MF-based models (MF-BPR [27], NeuMF [7] and ENMF [28]), graph embedding-based models (DeepWalk [29], LINE [30] and Node2Vec [31]), GCN-based models (GC-MC [16], PinSage [17], Multi-GCCF [12], NGCF [18], LightGCN [20], DGCF [21], and UltraGCN [22]). All models in experiments are optimized by the Adam optimizer [32] with the Xavier initialization [33]. For all baseline models, the embedding size is fixed to 64 and the batch size is set to 2048. Grid search is applied to tune the learning rate and the coefficient of L_2 normalization in the range of $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$.

3.3 Performance Comparison (RQ1)

Table 2 reports the results of the overall performance comparison. We obtain the following observations:

1. Our proposed MGRF achieves the best performance for all metrics in all datasets. By crossing features between users and items, MGRF is capable of exploring the fine-grained features and collaborative information in an explicit way. This verifies the importance of capturing collaborative information in the dual information crossing interaction. Moreover, compared with LightGCN (currently the state-of-the-art GCN-based recommendation model), MGRF fuses multi-graph information to infer user preference while LightGCN merely considers the user-item bipartite graph. This demonstrates that MGRF can exploit the latent fine-grained information by fusing multiple graphs and integrating different embeddings.
2. For the recommendation task, viewing the user-item interactive relationship as a heterogeneous bipartite graph can achieve better performance. This could explain why graph embedding-based models have poor results. Traditional random walk or heuristic mining strategies used in many graph embedding methods can hardly capture collaborative information for effective recommendation. The reason that LINE performs much better than DeepWalk and Node2Vec lies in the random walk strategy of LINE is more similar to a BFS (Breadth-First-Search) way, which could aggregate close-hop neighbors with more interactive information between users and items.
3. Based on the user-item heterogeneous bipartite graph, further exploring the homogeneous information can bring improvement in recommendation. As ob-

	Gowalla		Yelp2018		Amazon-Book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
MF-based Models						
MF-BPR	0.1291	0.1109	0.0433	0.0354	0.0250	0.0196
NeuMF	0.1399	0.1212	0.0451	0.0363	0.0258	0.0200
ENMF	0.1523	0.1315	0.0624	0.0515	0.0359	0.0281
Graph Embedding-based Models						
DeepWalk	0.1034	0.0740	0.0476	0.0378	0.0346	0.0264
LINE	0.1335	0.1056	0.0549	0.0446	0.0410	0.0318
Node2Vec	0.1019	0.0709	0.0452	0.0360	0.0402	0.0309
GCN-based Models						
GM-MC	0.1395	0.1204	0.0462	0.0379	0.0288	0.0224
PinSage	0.1380	0.1196	0.0471	0.0393	0.0282	0.0219
Multi-GCCF	0.1595	0.1326	<u>0.0667</u>	0.0510	0.0363	0.0256
NGCF	0.1570	0.1327	0.0579	0.0477	0.0344	0.0263
LightGCN	0.1830	0.1554	0.0649	0.0530	0.0411	0.0315
DGCF	0.1842	0.1561	0.0654	0.0534	0.0422	0.0324
UltraGCN	<u>0.1845</u>	**0.1566**	<u>0.0667</u>	<u>0.0552</u>	<u>0.0504</u>	<u>0.0393</u>
MGRF	**0.1857**	<u>0.1563</u>	**0.0681**	**0.0561**	**0.0537**	**0.0412**

Table 2. Overall performance comparison on Gowalla, Yelp2018, and Amazon-Book (the best one in bold and the 2^{nd} best with underline)

served in Table 2, traditional deep recommendation models are not sufficient to yield optimal embeddings because they consider only user and item features. Most matrix factorization only considers user-item interactions when developing embeddings, while GCN-based models could involve homogeneous interactions when they contain two layers. User-user and item-item relationships are also very important information, and two-hop neighborhoods in the bipartite graph can capture these homogeneous information to some extent. This could explain why GCN-based models are better than most MF-based models. ENMF achieves good results because it adopts a strategy of Efficient Non-Sampling to modify the loss function and alleviate the imbalance of positive and negative samples in the recommendation, thus improving its performance.

4. Fusing heterogeneous interaction and homogeneous information in an explicit way can further improve the recommendation results. This could explain why Multi-GCCF outperforms GM-MC, PinSage and NGCF. Though both considering higher-order user-item interactions, Multi-GCCF beats NGCF since it integrates the proximal information by building and processing user-user and item-item homogeneous graphs. Although Multi-GCCF is trained with additional information from different graph, it merely merges different embeddings in the step of final embedding, without fully integrating the interactive information of heterogeneous graph and topological information of homogeneous graphs, resulting in limited performance improvement.

In summary, iteratively fusing heterogeneous interactions with explicitly encoded homogeneous information could effectively improve the recommendation quality, which reveals the superiority of our proposed MGRF.

3.4 Ablation Analysis (RQ2)

Ablations	Gowalla		Yelp2018	
	Recall	NDCG	Recall	NDCG
Best baseline (LightGCN)	0.1830	0.1554	0.0649	0.0530
MGRF-C	0.1817	0.1527	0.0563	0.0459
MGRF-L	0.1812	0.1525	0.0607	0.0492
MGRF-I	0.1832	0.1554	0.0654	0.0532
MGRF-U	0.1836	0.1557	0.0664	0.0548
MGRF	**0.1857**	**0.1563**	**0.0681**	**0.0561**

Table 3. Ablation analysis on Gowalla and Yelp2018

To evaluate and verify the effectiveness of the components (i.e. dual information crossing interaction and multi-graph fusing propagation) of our proposed MGRF model, we derive four different models (namely MGRF-C, MGRF-I, MGRF-U, MGRF-L) and conduct an ablation analysis on Gowalla and Yelp2018. Compared with MGRF, MGRF-C only includes the proposed component of dual information crossing interaction in the user-item bipartite graph. Based on MGRF-C, MGRF-U and MGRF-I also include the proposed component of multi-graph fusing propagation on the user side or item side, respectively. MGRF-L includes the proposed dual information crossing interaction and multi-graph fusing propagation on both user and item sides, but only applies the dual information crossing interaction one-time in the step of final iteration. It is notable that if we replace the element-wise weighted mean aggregator (cf. Equations (4) and (5)) in MGRF-C with the weighted sum aggregator, we will obtain LightGCN, which thus serves as our best baseline. Table 3 illustrates the performance of models with different component combinations. The embedding size is 64 for all ablation experiments.

In MGRF-C, topological relationships between user-user and item-item are implicitly learned with user-item relationships through the same message passing layers. MGRF-C ignores the intrinsic difference between the two types of nodes and fails to capture the relative importance of user-user and item-item relationships, resulting in poor performance. In contrast, MGRF-I and MGRF-U are not only capable of explicit message passing so that we can separately learn item-item or user-user relationships, but also enable us to manually adjust the relative importance of different relationships. Besides, we notice that the performance of MGRF-I is slightly worse than MGRF-U, and both lower than MGRF. The reason may be that only one side of the additional information (user-user graph or item-item graph) is fused and the other side is ignored, resulting in the information collected from the other side being very sparse for most entities (users or items). It can be seen from Table 1 that the corpus of items can be enormously large and always larger than users. In MGRF-I, active users can get more information from the side of items, resulting in more biased embeddings and slightly worse performance. Though MGRF-L keeps all the components, its performance is obviously worse than MGRF-U and MGRF-I, implying that fusing the user-item heterogeneous graph with the other two homogeneous graphs only one time is clearly not enough to obtain high quality embeddings.

In summary, the combination of dual information crossing interaction and multi-graph fusing propagation in MGRF is demonstrated to be effective. It can flexibly and separately learn user-item, user-user and item-item relationships, and the iteratively fusing of different embeddings can effectively capture more comprehensive information to improve recommendation performance.

3.5 Hyper-Parameter Studies (RQ3)

3.5.1 Sample Number

The forward sampling phase is designed to deal with the long-tailed nature of the degree distributions in the user-item bipartite graph. In this subsection, hyper-parameter experiments are carried out to study the effect of sample number, and the corresponding results are shown in Table 4. We make the following observations: when the number of samples approaches the range of $10 \sim 15$, it achieves better performance. One reason might be that filtering the low-quality messages of target nodes' neighbors makes the embeddings effectively not only aggregate the information of the important neighbors but also mitigate the noise of irrelevant nodes, resulting in an optimal balance.

K	Gowalla		Yelp2018	
	Recall	NDCG	Recall	NDCG
5	0.1793	0.1499	0.0632	0.0504
10	**0.1857**	**0.1563**	0.0672	0.0543
15	0.1843	0.1559	**0.0681**	**0.0561**
20	0.1836	0.1557	0.0654	0.0532

Table 4. Effect of sample number K

3.5.2 Information Fusion Method

In this subsection, we compare different information fusion methods that summarize set of embeddings into one single embedding vector. These information fusion methods include element-wise sum, concatenation and self-attention mechanisms.

Table 5 shows the experimental results on Gowalla and Yelp2018. We make the following observations: element-wise sum performs much better than concatenation and self-attention. The reason is described as follows: element-wise sum generates an embedding of the same dimension as the component embeddings and does not involve any additional learnable parameters. The additional flexibility of self-attention and concatenation may harm the generalization capability of the model.

Fusion methods	Gowalla		Yelp2018	
	Recall	NDCG	Recall	NDCG
Self-attention	0.1836	0.1557	0.0654	0.0532
Concatenation	0.1793	0.1499	0.0632	0.0504
Element-wise sum	**0.1857**	**0.1563**	**0.0681**	**0.0561**

Table 5. Effect of information fusion method

4 CONCLUSION

In this paper, we present a novel Multi-Graph Recommendation Framework (MGRF) that iteratively incorporates multiple graphs to explicitly represent user-item, user-user and item-item relationships. MGRF contains two key components: dual information crossing interaction and multi-graph fusing propagation. Dual information crossing interaction ensures feature crossing between heterogeneous nodes throughout the whole embedding process, while multi-graph fusing propagation makes full use of homogeneous relationships to explicitly encode topological information. MGRF iteratively interlaces these two components until the optimal embeddings are obtained. Therefore, MGRF can sufficiently integrate heterogeneous and homogeneous graphs to improve the quality of the final embeddings. Extensive experiments on three real-world datasets demonstrate the effectiveness of our proposed MGRF, and the ablation studies quantitatively verify that each component in MGRF makes a necessary contribution. We now just explicitly capture homogenous information based on the existing similarity measurement method. In future, we will extend MGRF to auxiliary information, such as temporal signals.

REFERENCES

[1] Raza, S.—Ding, C.: News Recommender System: A Review of Recent Progress, Challenges, and Opportunities. Artificial Intelligence Review, Vol. 55, 2022, No. 1, pp. 749–800, doi: 10.1007/s10462-021-10043-x.

[2] Chen, L.—Cao, J.—Wang, Y.—Liang, W.—Zhu, G.: Multi-View Graph Attention Network for Travel Recommendation. Expert Systems with Applications, Vol. 191, 2022, Art. No. 116234, doi: 10.1016/j.eswa.2021.116234.

[3] Wu, C.—Wu, F.—Qi, T.—Liu, Q.—Tian, X.—Li, J.—He, W.—Huang, Y.—Xie, X.: FeedRec: News Feed Recommendation with Various User Feedbacks. Proceedings of the ACM Web Conference 2022 (WWW '22), 2022, pp. 2088–2097, doi: 10.1145/3485447.3512082.

[4] Shen, Q.—Wen, H.—Tao, W.—Zhang, J.—Lv, F.—Chen, Z.—Li, Z.: Deep Interest Highlight Network for Click-Through Rate Prediction in Trigger-Induced Recommendation. Proceedings of the ACM Web Conference 2022 (WWW '22), 2022, pp. 422–430, doi: 10.1145/3485447.3511970.

[5] Chen, J.—Wang, Y.—Huang, Q.—Jiang, B.—Liu, P.: Open APIs Recommendation with an Ensemble-Based Multi-Feature Model. Expert Systems with Applications, Vol. 196, 2022, Art. No. 116574, doi: 10.1016/j.eswa.2022.116574.

[6] Resnick, P.—Iacovou, N.—Suchak, M.—Bergstrom, P.—Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work (CSCW '94), 1994, pp. 175–186, doi: 10.1145/192844.192905.

[7] He, X.—Liao, L.—Zhang, H.—Nie, L.—Hu, X.—Chua, T. S.: Neural Collaborative Filtering. Proceedings of the 26th International Conference on World Wide Web (WWW '17), 2017, pp. 173–182, doi: 10.1145/3038912.3052569.

[8] Shan, Y.—Hoens, T. R.—Jiao, J.—Wang, H.—Yu, D.—Mao, J. C.: Deep Crossing: Web-Scale Modeling Without Manually Crafted Combinatorial Features. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 255–262, doi: 10.1145/2939672.2939704.

[9] Cheng, H. T.—Koc, L.—Harmsen, J.—Shaked, T.—Chandra, T.—Aradhye, H. et al.: Wide & Deep Learning for Recommender Systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016), 2016, pp. 7–10, doi: 10.1145/2988450.2988454.

[10] Guo, H.—Tang, R.—Ye, Y.—Li, Z.—He, X.: DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI '17), 2017, pp. 1725–1731, doi: 10.48550/arXiv.1703.04247.

[11] Batmaz, Z.—Yurekli, A.—Bilge, A.—Kaleli, C.: A Review on Deep Learning for Recommender Systems: Challenges and Remedies. Artificial Intelligence Review, Vol. 52, 2019, No. 1, pp. 1–37, doi: 10.1007/s10462-018-9654-y.

[12] Sun, J.—Zhang, Y.—Ma, C.—Coates, M.—Guo, H.—Tang, R.—He, X.: Multi-Graph Convolution Collaborative Filtering. 2019 IEEE International Conference on Data Mining (ICDM), 2019, pp. 1306–1311, doi: 10.1109/ICDM.2019.00165.

[13] Goyal, P.—Ferrara, E.: Graph Embedding Techniques, Applications, and Performance: A Survey. Knowledge-Based Systems, Vol. 151, 2018, pp. 78–94, doi: 10.1016/j.knosys.2018.03.022.

[14] Wu, S.—Sun, F.—Zhang, W.—Xie, X.—Cui, B.: Graph Neural Networks in Recommender Systems: A Survey. ACM Computing Surveys (CSUR), Vol. 55, 2022, No. 5, Art. No. 97, doi: 10.1145/3535101.

[15] Wang, S.—Hu, L.—Wang, Y.—He, X.—Sheng, Q. Z.—Orgun, M.—

Cao, L.—Wang, N.—Ricci, F.—Yu, P. S.: Graph Learning Approaches to Recommender Systems: A Review. CoRR, 2020, doi: 10.48550/arXiv.2004.11718.

[16] van den Berg, R.—Kipf, T. N.—Welling, M.: Graph Convolutional Matrix Completion. KDD '18 Deep Learning Day Workshop, 2018, doi: 10.48550/arXiv.1706.02263.

[17] Ying, R.—He, R.—Chen, K.—Eksombatchai, P.—Hamilton, W. L.—Leskovec, J.: Graph Convolutional Neural Networks for Web-Scale Recommender Systems. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '18), 2018, pp. 974–983, doi: 10.1145/3219819.3219890.

[18] Wang, X.—He, X.—Wang, M.—Feng, F.—Chua, T. S.: Neural Graph Collaborative Filtering. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), 2019, pp. 165–174, doi: 10.1145/3331184.3331267.

[19] Chen, L.—Wu, L.—Hong, R.—Zhang, K.—Wang, M.: Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, No. 1, pp. 27–34, doi: 10.1609/aaai.v34i01.5330.

[20] He, X.—Deng, K.—Wang, X.—Li, Y.—Zhang, Y.—Wang, M.: LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), 2020, pp. 639–648, doi: 10.1145/3397271.3401063.

[21] Wang, X.—Jin, H.—Zhang, A.—He, X.—Xu, T.—Chua, T. S.: Disentangled Graph Collaborative Filtering. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), 2020, pp. 1001–1010, doi: 10.1145/3397271.3401137.

[22] Mao, K.—Zhu, J.—Xiao, X.—Lu, B.—Wang, Z.—He, X.: UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM '21), 2021, pp. 1253–1262, doi: 10.1145/3459637.3482291.

[23] Zhao, Z. D.—Shang, M. S.: User-Based Collaborative-Filtering Recommendation Algorithms on Hadoop. 2010 Third International Conference on Knowledge Discovery and Data Mining, IEEE, 2010, pp. 478–481, doi: 10.1109/WKDD.2010.54.

[24] Chen, H.—Yin, H.—Chen, T.—Wang, W.—Li, X.—Hu, X.: Social Boosted Recommendation with Folded Bipartite Network Embedding. IEEE Transactions on Knowledge and Data Engineering, Vol. 34, 2022, No. 2, pp. 914–926, doi: 10.1109/TKDE.2020.2982878.

[25] Rendle, S.—Freudenthaler, C.—Gantner, Z.—Schmidt-Thieme, L.: BPR: Bayesian Personalized Ranking from Implicit Feedback. Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09), 2009, pp. 452–461, doi: 10.48550/arXiv.1205.2618.

[26] Järvelin, K.—Kekäläinen, J.: Cumulated Gain-Based Evaluation of IR Techniques. ACM Transactions on Information Systems (TOIS), Vol. 20, 2002, No. 4,

pp. 422–446, doi: 10.1145/582415.582418.

[27] Koren, Y.—Bell, R.—Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. Computer, Vol. 42, 2009, No. 8, pp. 30–37, doi: 10.1109/MC.2009.263.

[28] Chen, C.—Zhang, M.—Zhang, Y.—Liu, Y.—Ma, S.: Efficient Neural Matrix Factorization Without Sampling for Recommendation. ACM Transactions on Information Systems (TOIS), Vol. 38, 2020, No. 2, Art. No. 14, doi: 10.1145/3373807.

[29] Perozzi, B.—Al-Rfou, R.—Skiena, S.: DeepWalk: Online Learning of Social Representations. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14), 2014, pp. 701–710, doi: 10.1145/2623330.2623732.

[30] Tang, J.—Qu, M.—Wang, M.—Zhang, M.—Yan, J.—Mei, Q.: LINE: Large-Scale Information Network Embedding. Proceedings of the 24th International Conference on World Wide Web (WWW '15), 2015, pp. 1067–1077, doi: 10.1145/2736277.2741093.

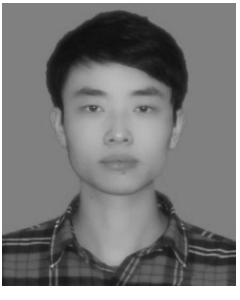
[31] Grover, A.—Leskovec, J.: node2vec: Scalable Feature Learning for Networks. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 2016, pp. 855–864, doi: 10.1145/2939672.2939754.

[32] Kingma, D. P.—Ba, J.: Adam: A Method for Stochastic Optimization. In: Bengio, Y., LeCun, Y. (Eds.): 3rd International Conference on Learning Representations (ICLR 2015). 2015, doi: 10.48550/arXiv.1412.6980.

[33] Glorot, X.—Bengio, Y.: Understanding the Difficulty of Training Deep Feedforward Neural Networks. In: Teh, Y. W., Titterington, M. (Eds.): Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research (PMLR), Vol. 9, 2010, pp. 249–256, `https://proceedings.mlr.press/v9/glorot10a.html`.

Xiang Lin received his Master's degree from the China University of Mining and Technology. Currently, he serves as Assistant Professor in the School of Information Engineering at Xuzhou College of Industrial Technology. His research interests include recommender systems, graph embedding, and physical information neural networks.

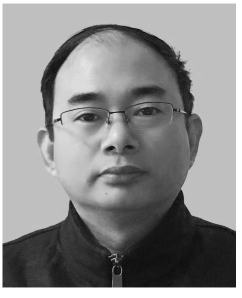
Fangyu Han is a third post-graduate year student at the School of Computer Science and Technology, China University of Mining and Technology. Her research interests include collaborative filtering, fair recommendation, and explainable recommendation.



Xiaobin Rui received his Ph.D. degree from the China University of Mining and Technology. Currently, he serves as a Lecturer at the School of Computer Science and Technology in China University of Mining and Technology. He has published more than 10 papers in international conferences and journals. His research interests include social network analysis, influence maximization, and graph embedding.

Chengcheng Sun is a Ph.D. student at the School of Computer Science and Technology, China University of Mining and Technology. His research interests include social network analysis, graph deep learning, and graph neural networks.

Zhixiao Wang received his Ph.D. degree from the Tongji University. Currently, he serves as Professor at the School of Computer Science and Technology in China University of Mining and Technology. He has published more than 30 papers in international conferences and journals. His research interests include complex network, social network analysis and data mining.



Lijun Yan received M. Sc. in 2005 and Ph.D. in 2016, both from the Shanghai Normal University. He is currently an Engineer in the College of Information, Mechanical and Electrical Engineering at Shanghai Normal University, Shanghai, China. He is mainly engaged in research and development in the interdisciplinary fields of information technology and environmental science.