Semantic and Structural view Fusion Modeling for Social Recommendation

Kun Yuan, Guannan Liu, Junjie Wu, Hui Xiong, *Fellow, IEEE*

Abstract—Existing studies have shown that user-item interaction data and social relation data can be jointly used for enhancing the performance of social recommendation. However, limited research has a focus on investigating how to deeply exploit different views of social interaction structures and rating behavior differences for further improving social recommendation. To this end, in this paper, we propose to integrate information from both semantic and structural views for social recommendation. Specifically, we first design a collective intelligence-based strategy to reveal high-quality implicit relations for both users and items. Then, by reformulating all available nodes and relations as a heterogeneous graph, we define multiple semantic metapaths to capture diverse preferences for comprehensive user and item representations. While various metapaths enlarge the representation capacity of users and items, they also introduce noise and irrelevant information. We recall that, for the user-item interaction graph, different structure sizes (e.g. local and global structures) provide diverse and complementary information for recommendation. Motivated by this, we propose a semantic and structural view fusion framework for social recommendation (S4Rec), which consists of a deep graph model and a wide attentive SVD (Singular Value Decomposition) model for rating prediction by taking the local and global structure as input and aggregating messages along the predefined metapaths. Finally, the two predicted results are adaptively fused to achieve the final both accurate and stable prediction. In addition, we treat the user's rating behavior difference as the relative position difference problem in the embedding space, and model it with TransH to improve the generalization ability of the main rating model. Extensive experiments on three open datasets demonstrate the superiority of our framework compared with state-of-the-art methods. Particularly, our model outperforms other baselines under different sparsity conditions, further validating the effectiveness on cold-start users. We release the source code at https://github.com/lcwy220/Social-Recommendation.

Index Terms—Recommender systems, social recommendation, graph neural network, wide and deep, dropout.

1 Introduction

Social recommendation, which incorporates social relations such as friends, trust or colleagues into recommender systems, has developed and shown significant potential to improve recommendation performance [1]. As well supported by the social influence theories, users with explicit observed social relations tend to share similar preferences, and their decisions may also be influenced by friends, which corresponds to homophily [2] and social influence [3], respectively. Motivated by these theories, early efforts resort to Matrix Factorization techniques to learn latent factors of users and items by factorizing user-item interaction matrix and the adjacency matrix of the social graph simultaneously. They can be classified into co-factorization methods [4], [5], ensemble methods [6], [7] and regularization methods [8], [9], which merely consider the first-order relations for social recommendation. Recently, the emerging Graph Neural Networks (GNNs) have shown powerful capacity in advancing social recommendation by regarding the user-item interaction data and observed social relations as two graphs [10], [11], [12], [13]. These methods iteratively aggregate information from the neighbors of the two graphs and concatenate them to learn fused representations.

However, despite the great performance of existing social recommendation methods, they still suffer from two limitations. First, it's well known that the sparsity and unbalanced distribution of the observed social relations hinder further improvement of social recommendation. To address this, some work resorts to mining implicit relations of users and items to enrich data [14], [15]. Specifically, user implicit relation can be established between two users with unobserved social relation but sharing similar preferences, while item implicit relation denotes items which are preferred by the same user. However, the low cost of constructing social relations and the openness nature of social networks may inevitably incur enormous noises, which degrade the validity of both explicit and implicit relations. Though there exist some other methods that focus on how to learn robust graph structure, quite a few methods have simultaneously considered the implicit relation augmentation and noise reduction. In addition, for the user rating behaviors, different levels of the ratings also denote *relative position difference* between users and items in the embedding space, i.e., high-rating item should be closer to the user than low-rating item. However, this inductive bias is not explicitly preserved and generalized to other similar users and items, which hiders further improvement of the model.

In this paper, we propose to exploit the implicit relations and user rating behavior to enhance social recommendation. Technically, different from previous work [14], [15], we define implicit user relations as two disconnected users

- *Kun Yuan is with the Hefei University of Technology, China, and Key Laboratory of Process Optimization and Intelligent Decision Making, Ministry of Education,China. Email: yuankun@hfut.edu.cn*
- *Guannan Liu and Junjie Wu are with the Beihang University, China. Email: {liugn, wujj}@buaa.edu.cn. Corresponding author: Guannan Liu.*
- *Hui Xiong is with the Hong Kong University of Science and Technology (Guangzhou), China, and Guangzhou HKUST Fok Ying Tung Research Institute. Email: xionghui@ust.hk*

Manuscript received April 19, 2005; revised August 26, 2015.

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230972

who have same followers because of preference equivalence from their shared followers. Besides, implicit item relations refer to items rated by the same users with closer ratings because they tend to be more similar than any two randomly selected ones. Then, combined with the explicit social relations, we construct a heterogeneous information network (HIN) in which comprehensive users and items preferences can be captured by defining different types of semantic metapaths. For example, to model diverse user preferences, the metapath *user-items*, *user-explicit users* and *user-implicit users* provide heterogeneously semantic-view signals for user representation, i.e. collaborative domain, social domain and hybrid domain.

However, the rich relations and metapaths are also noisy, which further incur the challenge of robust representation learning of users and items. To alleviate this problem, we propose a novel mechanism, *Preserving & DropNode*, to reduce the negative impact of noisy relations. Specifically, we randomly drop out the nodes and in each metapath, so that the remaining relations compose different copies of local structural view of the metapath. Subsequently, we design a metapath-guided deep graph model to aggregate information along all metapaths for robust representation modeling. Note that though the generalization strategy *DropNode* can prevent over-fitting for noisy relations, it may incur large prediction variance due to the heterogeneity aroused in the random sampling, especially for the users and items with extremely unbalanced rating records. To strike a balance between bias and variance, we further propose to *Preserve* the metapath structure from global view with developing a wide and shallow attentive SVD model, to obtain more stable prediction. Then, both the predictions from local and the global structural view can be adaptively fused to achieve the final accurate and stable prediction. Considering both the semantic and structural view are utilized, we name our proposed model *Semantic and Structural view fuSion framework for Social Recommendation (S4Rec)*.

Furthermore, to address the challenge of the positional bias of the ratings, we propose to generalize the *relative position difference*. In particular, the relative position difference refers to the fact that the high-rating items should be closer to the user than low-rating items in embedding space, which however is merely addressed in prior studies. To address this, we regard each rating behavior as a positional triplet (*User*, *Rating*, *Item*) in Knowledge Graph [16], and utilize a graph embedding method to preserve the relative position difference as a constraint to the main rating model so as to improve the overall generalization ability. Finally, extensive experiments on three public datasets demonstrate that our model can achieve both more accurate and stable performance compared with state-of-the-art methods. Particularly, the proposed model can outperform others under different sparsity conditions, which further demonstrates its effectiveness and generalization on cold-start users.

The rest of the paper is organized as follows. Section 2 introduces related work about social recommendation. In Section 3 we present preliminaries and give the formal problem definition. Then we illustrate the details of the proposed framework in Section 4, and report the experimental results in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

In this section, we briefly introduce related work about classical social recommendation, graph based social recommendation, and multi-view modeling in recommendation.

Classical Social Recommendation. Recent years has witnessed the significant progress of exploiting social relations for recommendation for various online social platforms [17]. Inspired by the studies in social theories such as homophily [2] and social influence [3], users' preferences tend to be similar or impacted by their social friends. Based on above theories, prior work focuses on leveraging Matrix Factorization technique for social recommendation, which can be divided into three groups [1], *i.e.*, co-factorization methods [4], [5], ensemble methods [6], [7], and regularization methods [8], [9]. As the representative work of co-factorization methods, SoRec [4] and TrustMF [5] co-factorize user-item rating matrix and social network structure with probabilistic matrix factorization through sharing user latent feature space. Based on the idea that the user and her social friends should have similar rating preferences, RSTE [6] predicts the missing rating as a linear combination of ratings from herself and her social friends. SocialMF [8] and SoReg [18], as typical regularization models, force the preference of the user to be close to her social friends and act as a regularization term for the matrix factorization framework. However, this type of methods only capture linear information.

As developed of DNNs, more and more deep models are proposed to learn complex representation from user-item interactions and social domain [19], [20]. NSCR [21] leverage users with social friends as bridges to propagate user representations modeled from attribute-aware deep collaborative filtering model. SGL [22] explores self-supervised learning on user-item graph to improve accuracy and robustness of GCNs for recommendation. However, data sparsity and heterogeneity always hinder the further improvement of social recommendation. Therefore, besides the explicit social relations, research efforts also shift attention to undermine implicit relations for social recommendation [14], [15]. TrustSVD [23] incorporates both the explicit and implicit influence of trusted users and rated items. CUNE [24] identifies top implicit friends with random walks and utilizes them to regularize matrix factorization prediction model. In summary, most of these studies only partially tap the potential of interaction data and social relation since more high-quality implicit relations have not been mined.

Graph based Models for Recommendation. Recently, GNNs have achieved great success in recommendation task for their strong capability in modeling graph data [25], [26], [27]. Taking user-item interaction graph and social graph as input, many GNN-based deep models are proposed for social recommendation problem [13], [28], [29]. DiffNet [30] and its extension DiffNet++ [31] model the recursive social diffusion process of social influence and user interest. GraphRec [10] applies GAT to capture and fuse information from user-item graph and social graph. FBNE [12] folds the user-item bipartite graph to construct homogeneous node sequences and utilizes graph attention network to aggregate implicit high-order relations. MHCN [32] propose to leverage a hypergraph convolutional network to depict high-order user relations which further enhances social recommendation. Although various GNN-based models have been proposed, they don't fully unlock the potential of observed and implicit relations in social recommendation. Besides, they neglect the relative position difference problem in embedding space hidden in user rating behaviors, and fail to generalize this inductive knowledge to other social friends to further enhance recommendation performance.

Multi-view Modeling for Recommendation. Multi-view learning aims at undermining information from diverse views [33]. From embedding view, DELF [34] introduces inherent and dynamic embeddings for users and items, respectively, and four user-item interactions are fused for preference prediction. From social effect view, DANSER [11] models the homophily and social influence for both users and items, and fuse them with a policy-based network. DiffNetLG [27] concerns on two kinds of implicit influence: local implicit influence of unobserved users and global implicit influence of popular items, and further combine them with explicit influence through a GCN for social recommendation. From features combination view, NeuMF [35] is an ensemble of Generalized Matrix Factorization (GMF) and Multilayer Perceptron (MLP) which models linear and nonlinear interaction between user and item features. Similarly, Wide & Deep model [36] adopts the wide component to memorize some typical feature interactions and the deep component to generalize unseen feature combinations. Different from above models, we explore the social recommendation from both semantic and structural view, and we argue that different interaction structures contribute differently to rating prediction. Based on this, we propose a structural and semantic fusion framework, *i.e.*, generating local structure with deep graph model for accuracy modeling and preserving global structure with shallow model for stability modeling. Therefore, although our model may have superficial resemblances with other models, they are essentially different.

3 Preliminaries and Problem Definition

3.1 Implicit Relations in Social Recommendations

Most social recommender systems utilize explicit social relations, *e.g.*, the friendship relation between users, to enhance the recommendation effectiveness because it is generally assumed that connected users might have similar preferences driven by homophily effect [2]. However, except for the explicit relations, there also exist some other implicit relations between users or items, which can provide adequate clues to reveal the diverse preferences in similar to the explicit relations, which enhances the representation learning for users and items in social recommendation.

3.1.1 Implicit user relations

The observable relations explicitly disclose the similar preferences between connected users. However, in most cases the explicit social relations are sparse and biased, limiting the utilization of the social information. Therefore, we resort to undermining *implicit user relations* from first-order to higher-order relations, where users sharing more mutual followers are assumed to establish implicit relations with similar preferences. More specifically, for user i, we define the implicit relations $H_U(i)$ as follows:

$$H_U(i) = \{k| \|s_{ji} = 1 \wedge s_{jk} = 1, j \in U\| \geq \tau\}, \quad (1)$$

where $s_{ji} = 1$ represents user j follows i explicitly, $\|\cdot\|$ represents the set size; τ denotes the cutoff threshold, where a larger τ indicates that the implicit relations require more shared followers. We further examine the effectiveness of the defined implicit relations in capturing user preferences by conducting comparative statistical analysis with explicit and random selected user relations on three publicly available datasets (the details of the datasets are introduced in Section 5). As shown in Table 1, the average rating difference between the user and her explicit and implicit friends are close, and they are both significantly smaller than that between any two random users. This demonstrates the auxiliary implicit user relations can provide clues to disclose the shared preferences.

3.1.2 Implicit item relations

In traditional item-based CF methods [12], [37] , item similarity is exploited to estimate the scores of a user for an item. However, traditional similarity measure, such as Pearson correlation coefficient (*PCC*) [38] usually neglects the sizes of rating users. In reality, two items can be more similar if more users rate them with closer ratings, which however, cannot be manifested in traditional measures. Thus, in order to uncover the *implicit item relations*, we construct a similarity measurement based on the sizes of the rating users and their corresponding rating scales to further establish implicit item relations. More formally, given that user i gives rating r_{ij} and r_{ik} to items j and k, respectively, we firstly define the similarity s_{jk} between item j and k in view of the user i:

$$s_{jk}^i = \frac{1}{1 + |r_{ij} - r_{ik}|}. \quad (2)$$

Then, we reformulate the similarity between any two items in view of all rated users, *i.e.*, $s_{jk} = \sum_{i \in U_{jk}} s_{jk}^i$, where U_{jk} denotes the set of users who give ratings to both item j and k. Obviously, the closer ratings two items receive from more users, the more similar they are. Furthermore, in order to obtain more compact implicit item relations, we require that $s_{jk} > 1$ and we only retain the Top-N items according to the descending values of $s_{j,\cdot}$, which are referred to as implicit item relations of item j.

$$H_V(j) = \{k, \cdots | s_{jk} \geq \cdots > 1\}. \quad (3)$$

We set $N = 20$ to obtain the implicit item relations for each item, and we compute the average rating differences between the item pairs in comparison with that derived by *PCC*. As can be observed from Table 2, the average rating difference of implicit item relations is close to that of explicit user relations calculated in Table 1 and significantly smaller than that of *PCC*, which demonstrates the effectiveness of the proposed measurement in seeking similar item relations.

3.2 Metapaths for Social Recommendation

The ultimate goal of social recommender system is to match users with their preferred items, *i.e.*, predicts rating between users and items. Except for the explicit social relations

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230972

TABLE 1
Statistical comparison results of explicit, implicit and random user relations.

	Explicit User Relation			Implicit User Relation			Random User		
Dataset	# users	# links	avg. rating diff.	# users	# links	avg. rating diff.	# users	# links	avg. rating diff.
Epinions	17,885	374,039	0.4413	21,600	595,953	0.4822	23,251	600,000	0.5484
Ciao	6,788	111,606	0.3894	7,205	201,261	0.4091	7,371	201,000	0.4508
yelp	21,266	497,206	0.4832	21,260	613,393	0.5332	21,461	610,000	0.6135

TABLE 2
Statistical comparison results of implicit similar items.

	Collective intelligence-based				Pearson correlation coefficient-based			
Dataset	# items	# links	# avg. links	avg. rating diff.	# items	# links	# avg. links	avg. rating diff.
Epinions	118,672	1,944,963	16.38	0.4598	120,711	2,265,571	18.77	0.7488
Ciao	89,544	1,513,785	16.91	0.3704	90,913	1,724,164	19.12	0.6667
yelp	101,478	1,841,815	18.15	0.5874	102,433	1,977,026	19.30	0.7503

available for recommendation modeling, we also have defined implicit relations that can enhance the preferences learning of users and items. Thus, for the sake of clarity, we map all the meaningful relations to a Heterogeneous Information Network (HIN) as shown in Figure 1, where the users and items are regarded as nodes, and the interaction between users and items, explicit social links as well as implicit relations denote different types of edges. Then to capture comprehensive user and item preferences, we define a series of metapaths listed in Table 3, which are widely used in HIN to capture the underlying semantic relations between different types of entities. Specifically, for each user node, we define *user-items*, *user-users* and *user-implicit users* based on the one-hop neighbors which directly reflect the inherent behaviors of users. Besides, we also define two-hop neighbors, *i.e.*, *user-users-items* and *user-implicit users-items*, to capture rating behaviors similarity along the higher-order metapaths. Similarly, for each item node, we define a series of mathpaths including *item-user*, *item-implicit items*, and *item-implicit items-users* for comprehensive representation learning. Note that compared with ego-network utilized in prior research [10], [30], these predefined metapaths explicitly contain user preferences and item attributes based on homophily effect and influence effect, and are conducive to better user and item representations.

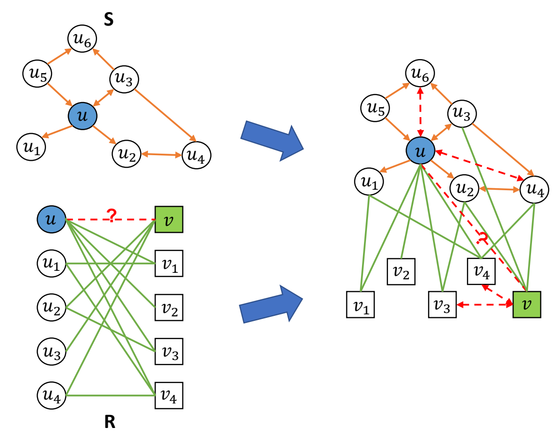


Fig. 1. Heterogeneous information network consists of user-item interaction graph and social graph. Red dot line denotes the implicit relations.

3.3 Metapath Guided Graph Attention Network

Graph Attention Network (GAT) [39] is widely used to encode the neighbor structure information of each node in graph and output a new low-dimensional dense node embedding. Along this line, many GAT-based recommendation methods have been proposed [10], [11], where both users and items are represented with embedding vectors via the message passing guided by GAT. Therefore, given a defined metapath, we can also derive the semantic embeddings of users and items with GAT to model the preference reflected by this metapath. Specifically, without loss of generality, assume that given node i and its local neighbors $N(i)$ in a metapath, the aggregation process can be defined as follows:

$$\hat{\boldsymbol{h}}_i = \sigma(\boldsymbol{W} \cdot \sum_{j \in N(i)} a_{ij} \boldsymbol{p}_j), \tag{4}$$

where σ and $\boldsymbol{W} \in \mathbb{R}^{d\times d}$ denote the activation function and the weight matrix, respectively. a_{ij} represents the importance of each $j \in N(i)$ in representation learning, which is learned by an attentive neural network with the inputs of embedding $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$, given by:

$$\boldsymbol{a}_{ij} = softmax\left[\boldsymbol{W}_2^T \sigma\left(\boldsymbol{W}_1\left[\boldsymbol{p}_i \oplus \boldsymbol{p}_j\right] + \boldsymbol{b}_1\right)\right], \tag{5}$$

where $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d\times d}$, and $\boldsymbol{b}_1, \boldsymbol{b}_2 \in \mathbb{R}^d$, are weights matrix and bias vectors, respectively. $\oplus$ denotes the concatenation operation and the softmax function is used to normalize the weight values. In this way, we can obtain the semantic embedding of any metapath through metapath-guided GAT.

3.4 Problem Definition

Let $U = \{u_1, u_2, \cdots, u_M\}$ and $V = \{v_1, v_2, \cdots, v_N\}$ be the sets of users and items respectively, where M is the number of users and N is the number of items, and the user-item rating matrix can be denoted by $\boldsymbol{R} = \{r_{ij}\}_{M\times N}$. Moreover, users can establish trust or friendship relations to other users in online platforms, and construct a user-user social graph $\boldsymbol{S} \in \boldsymbol{T}^{M\times M}$. In the social graph $\boldsymbol{S}$, if user i trusts or follow another user k, $s_{ik} = 1$, otherwise $s_{ik} = 0$.

TABLE 3
Metapaths designed for social recommendation.

	Metapaths	Example	Description
user	user-items	$u \leftarrow \{v_1, v_2, v_4\}$	rated items reflect the preference of the user
	user-users	$u \leftarrow \{u_3, u_5\}$	the user's preferences are the same with her friends
	user-implicit users	$u \leftarrow \{u_4, u_6\}$	extended implicit uesrs who share similar preferences with the user
	user-users-items	$u \leftarrow u_3 \leftarrow \{v\}$	the user has similar tastes with her friends
	user-implicit users-items	$u \leftarrow u_4 \leftarrow \{v, v_4\}$	the user also has similar tastes with implicit users
item	item-users	$v \leftarrow \{u_2, u_3, u_4\}$	rated users reflect the preference of the item
	item-implicit items	$v \leftarrow \{v_3, v_4\}$	extended implicit similar items with the item
	item-implicit items-users	$v \leftarrow v_4 \leftarrow \{u, u_1, u_4\}$	users who rated the implicit similar items may have similar taste on the item

Analogously, we use $F_U(i)$ to denote the set of social friends of user i in social graph, i.e., $F_U(i) = \{k|s_{ik} = 1\}$. In addition, as introduced previously, we further extract implicit user and itme relations H_U and H_V respectively, and then various metapaths can be defined to reflect diverse users and items preferences. Therefore, we extend the general social recommendation problem by incorporating the implicit relations and the metapaths, which is defined as follows:

Given the observed user-item rating graph $\boldsymbol{R}$ and user-user social graph $\boldsymbol{S}$, as well as the extended implicit relations H_U and H_V, we aim to predict the unknown rating values in $\boldsymbol{R}$ and recommend items to users with higher predicted ratings.

4 The Proposed Model

4.1 Model Framework

Given the multiple metapaths defined for users and items, an intuitive idea is to obtain user/item representations with all the interacted relations via GNNs [10], [11]. However, though the implicit relations and corresponding metapaths can extend the representation capacity of users and items, they also contain noises and irrelevant information such that the learned representations are over-fitting and lose of generalization if we simply apply GNNs to all metapaths without distinctions. For example, active users and popular items which have numerous interactions would generate plenty of metapaths, in which even the commonly used attention mechanism cannot guarantee to aggregate accurate users and items representations.

Therefore, to tackle with this dilemma, a filtering mechanism is necessary to diminish the negative impact of noises in the metapaths. In particular, prior research has showed that *dropout* mechanism improves the model performance and robustness by randomly dropping a partial nodes in the original deep neural network, which can reduce the effect of over-fitting [40], [41]. In spirit of this, we propose *DropNode* to enhance robustness and prevent over-fitting of metapath aggregation, which randomly drops out the nodes and the connected edges in the metapaths. In this regard, multiple copies of local view of the original metapath can be generated by the DropNode mechanism, which can be fitted with GAT to obtain robust and generalized representations meanwhile reduce the negative impact of noises.

However, if we only focus on the local view generated by *DropNode*, the model will be biased against a set of popular users and items, while the predictions on items with fewer and lower ratings incur greater loss. Thus, in order to correct this bias, we leverage all the first-order relations of a node with a shallow model to depict users and items preferences from the global structure view, which dedicate to stable rating prediction.

In summary, incorporating with both explicit and implicit semantic relations, we propose a local and global structure view fusion framework for social recommendation, which corresponds to deep and wide model, respectively. The overall framework is shown in Fig. 2.

4.2 Deep Graph Model for Local View

Given the multiple predefined metapaths constructed from the explicit and implicit relations, we firstly conduct *DropNode* by randomly dropping out the number of users or items to a fixed size, *e.g.*, 30. As a matter of fact, node sampling is a universal sub-graph sampling strategy to aggregate information from neighbor nodes [42]. Traditionally, a suitable size of node samples in homogeneous network can represent the overall distribution of neighbors. While in heterogeneous network, due to the rating bias of each node, a fixed-size node samples most likely leads to biased information aggregation. However, although the sampled nodes may generate biased metapaths, it can also provide different local views in capturing invariant rating patterns of each user/item such that the model will be more robust to noises and we obtain biased but more accurate prediction. Therefore, with the multiple local views of metapaths, we propose a deep graph model for representation learning of users and items, and further predict the ratings based on the learned ones.

Embedding Layer. It encodes users and items with corresponding latent representations. Let $\boldsymbol{P} = \{\boldsymbol{p}_i\}_{M\times d}$ be the user embeddings, where $\boldsymbol{p}_i$ denotes the latent embedding of user u_i and d is the embedding dimension. Correspondingly, another $\boldsymbol{Q} = \{\boldsymbol{q}_j\}_{N\times d}$ is proposed, where $\boldsymbol{q}_j$ denotes the latent embedding of item v_j.

User modeling. In Section 3.2, we have defined multiple semantic metapaths including *user-item*, *user-users* and *user-implicit users* from one-hop perspective, and *user-users-items* and *user-implicit users-items* from cross-domain perspective, which can help represent users and items from different domains. Then for each metapath, we utilize the metapath-guided GAT which aggregates different hop neighbors along the metapath to obtain the semantic embedding of the

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230972

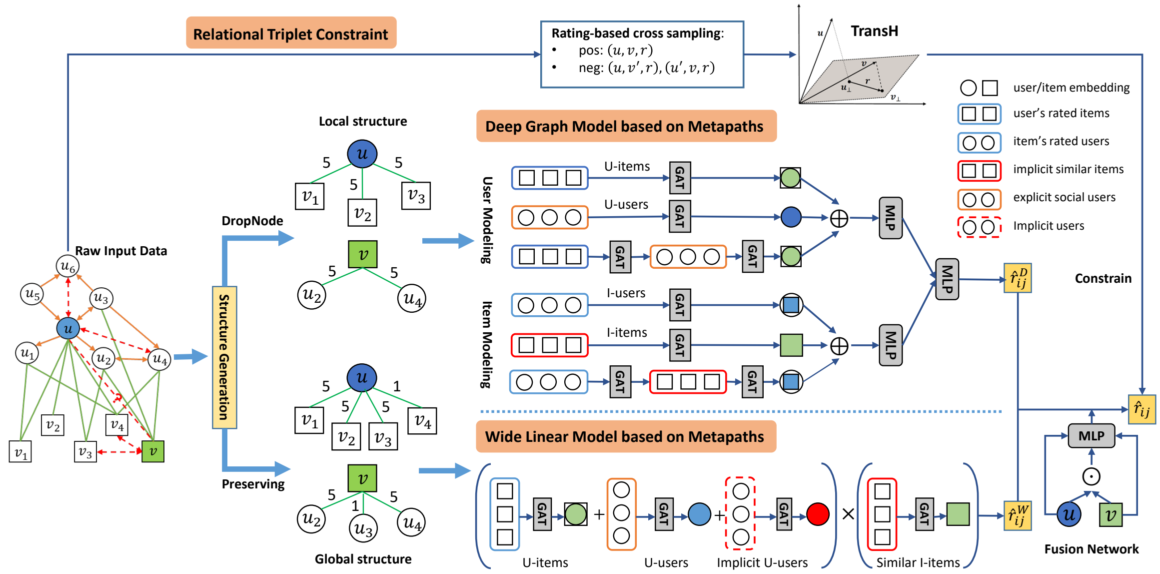


Fig. 2. Architecture of the proposed S4Rec model. GAT denotes the proposed metapath guided GAT for aggregation along predefined metapaths.

matapath. For example, for the meatapath *user-item*, we take the local structure of interacted items as input to Eq. (4) and output the item-based user embedding $\boldsymbol{p}_i^V \in \mathbb{R}^d$. In similar way, we can also obtain different metapath embeddings $\boldsymbol{p}_i^S$, $\boldsymbol{p}_i^{SV} \in \mathbb{R}^d$ from social graph, and $\boldsymbol{p}_i^H$, $\boldsymbol{p}_i^{HV} \in \mathbb{R}^d$ from implicit user relations. Subsequently, it is intuitive to merge the five types of user embeddings into a comprehensive one. Different fusion methods have been proposed, such as direct concatenation or weighted sum of these embeddings. In this paper we utilize a MLP to combine the multiple user embeddings and learn their relative importance automatically. Accordingly, the final user representation $\hat{\boldsymbol{p}}_i$ for user u_i is obtained by:

$$\hat{\boldsymbol{p}}_i = MLP_{user}([\boldsymbol{p}_i^V \oplus \boldsymbol{p}_i^S \oplus \boldsymbol{p}_i^{SV} \oplus \boldsymbol{p}_i^H \oplus \boldsymbol{p}_i^{HV}]). \quad (6)$$

Item modeling. Similarly, item modeling aims at learning a comprehensive item embedding $\hat{\boldsymbol{q}}_j$ from user-item interaction graph and implicit item relations. Starting from the centered item node, we select three metapaths to reflect different item attributes, i.e., *item-users*, *item-implicit items*, and *item-implicit items-users*, and further obtain the embedding of each metapath denoting as $\boldsymbol{q}_j^U$, $\boldsymbol{q}_j^H$, $\boldsymbol{q}_j^{HU} \in \mathbb{R}^d$, respectively. Then, we adopt another MLP to fuse the three types of embeddings. Correspondingly, the item representation $\hat{\boldsymbol{q}}_j$ for item v_j is computed as:

$$\hat{\boldsymbol{q}}_j = MLP_{item}([\boldsymbol{q}_j^U \oplus \boldsymbol{q}_j^H \oplus \boldsymbol{q}_j^{HU}]). \quad (7)$$

Rating Prediction. Then the learned user and item representations are concatenated and fed into a two-layer MLP to get the predicted ratings based on local structure:

$$\hat{r}_{ij}^D = MLP_{Deep}([\hat{\boldsymbol{p}}_i \oplus \hat{\boldsymbol{q}}_j]). \quad (8)$$

4.3 Wide Linear Attentive Model for Global View

Although DropNode mechanism help to improve robustness to noises, experiments show that it also incurs biased prediction results. We attempt to interpret this interesting phenomenon in the view of data distribution. Consider the imbalanced fact of data that high-rating items dominate the user-item rating matrix and only a small portion belongs to low-rating ones. Then the sampled items of local structure generated by our proposed *DropNode* are prior to popular and high-rating ones which are further intensified with deep graph model, resulting in high-accuracy prediction results to the majority but high-variance to the minor low-rating items. In order to correct the biased results, we revisit prior research and find that shallow linear model can output low-variance results, which is consistent with the conclusion that parameterizing users according to all item they rated shows more stability to the number of user interactions [43]. This finding provides the opportunity for achieving both high-accuracy and low-variance prediction results with combining both local and global structure of the metapath.

Therefore, in order to strike a balance between accuracy and stability, we propose a wide and shallow model for stability modeling from the global view of the one-hop relations. In particular, rather than modeling the users and items with neural-based models, we propose an Attentive SVD based on TrustSVD [23], in which both the explicit and implicit influence of the users and items are involved for prediction through linear combinations.

User Embedding. We firstly introduce $\boldsymbol{p}_i', \boldsymbol{q}_j' \in \mathbb{R}^{d'}$ to represent the latent factors of user u_i and item v_j, and we also define latent factors $\boldsymbol{x}_i, \boldsymbol{y}_i \in \mathbb{R}^{d'}$ to reflect implicit influence of u_i and v_j, respectively. Then integrating implicit influence from all rated items and user relations, the user embedding of u_i in the global view can be represented as:

$$\hat{\boldsymbol{p}}_i' = \boldsymbol{p}_i' + \sum_{k \in F_U(i)} \alpha_{ik}\boldsymbol{x}_k + \sum_{k \in H_U(i)} \beta_{ik}\boldsymbol{x}_k + \sum_{k \in R_V(i)} \gamma_{ik}\boldsymbol{y}_k, \quad (9)$$

where the three components correspond to implicit influence from social friends $F_U(i)$, implicit user relations $H_U(i)$ and rated items $R_V(i)$, respectively. Furthermore, we also introduce an attention mechanism to discriminate different importance of the relations with the attention weight $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$

and $\boldsymbol{\gamma}$, respectively. Taking $\boldsymbol{\alpha}_{ik}$ as an example, the attention weight can be derived as follows:

$$\boldsymbol{\alpha}_{ik} = softmax(\boldsymbol{W}_2^T \sigma(\boldsymbol{W}_1[\boldsymbol{x}_i \oplus \boldsymbol{x}_k] + \boldsymbol{b}_1)), \tag{10}$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2 \in \mathbb{R}^{d' \times d'}$ and $\boldsymbol{b}_1 \in \mathbb{R}^{d'}$ are weight parameters and σ denotes activation function.

Item Embedding. Likewise, we can represent item from implicit item relations with attention weight $\boldsymbol{\eta}$:

$$\hat{\boldsymbol{q}}'_j = \sum_{k \in \{H_V(j) \cup j\}} \eta_{jk} \boldsymbol{q}'_k \tag{11}$$

Rating Prediction. Afterwards, the rating $\hat{r}_{ij}^W$ from global structure is predicted by:

$$\hat{r}_{ij}^W = b_i + b_j + \mu + \hat{\boldsymbol{q}}_j'^T \hat{\boldsymbol{p}}'_i, \tag{12}$$

where b_i and b_j denotes the user and item biases, and μ is the global average rating.

4.4 Wide and Deep Fusion Module

With both local and global views captured with deep and wide prediction models, we can derive the rating prediction from the two sub-models, *i.e.*, $\hat{r}_{ij}^D$ and $\hat{r}_{ij}^W$. Then, we further fuse the two via a weight parameter λ_{ij} to obtain the final prediction:

$$\hat{r}_{ij} = \lambda_{ij} \hat{r}_{ij}^D + (1 - \lambda_{ij}) \hat{r}_{ij}^W. \tag{13}$$

The weight λ_{ij} represents the relative importance of the two views, which can be a hyper-parameter, or estimated by a MLP, which takes the concatenation of inherent embeddings of user $\boldsymbol{p}_i$ and $\boldsymbol{q}_j$ as input:

$$\lambda_{ij} = MLP_{fusion}([\boldsymbol{p}_i \oplus \boldsymbol{q}_j]). \tag{14}$$

4.5 Relational Triplet Constraint

Besides the predefined multiple metapaths for comprehensive users and items representation learning, the user rating behavior also contains valuable information for modeling. In fact, an item with higher rating by a user should better reflect the user preference and hence the user and item should be closer than that of a low-rating item in the embedding space. If this relative position is memorized by the model and further promoted to other similar users and items, it will greatly improve the performance and generalization ability of the model. However, this relative position difference has merely been addressed in previous approaches. Therefore, in addition to the rating prediction, we also conduct the relative position difference modeling to further improve the overall recommendation performance.

To enable the learning of the positional difference, we firstly transfer the user rating behavior to a triplet with the form of (*User, Rating, Item*). For example, u_1 rates item v_1 with 1 can be denoted as $(u_1, 1, v_1)$. If we regard the user, item and rating score as head, tail entity and relation respectively, a knowledge graph can be obtained. In this regard, we can apply the widely used *knowledge graph* (KG) embedding technique *TransH* [16] to constrain the relative position of users and items. More specifically, given a triplet (u, r, v), we learn the embeddings of each entity and relation by optimizing the translation principle $\boldsymbol{p}_u^r + \boldsymbol{e}_r \approx \boldsymbol{q}_v^r$. Herein, $\boldsymbol{p}_u$, $\boldsymbol{q}_v$, $\boldsymbol{e}_r \in \mathbb{R}^d$ denotes the embedding for u, v, r, respectively; and $\boldsymbol{p}_u^r$, $\boldsymbol{q}_v^r$ are the projected representations of $\boldsymbol{p}_u$ and $\boldsymbol{q}_v$ on relation r's hyperlane with $\boldsymbol{w}_r \in \mathbb{R}^d$ as the normal vector. Thus the scoring function is defined as:

$$f(u, r, v) = \|\boldsymbol{p}_u^r + \boldsymbol{e}_r - \boldsymbol{q}_v^r\|_2^2, \tag{15}$$

where $\boldsymbol{p}_u^r = \boldsymbol{p}_u - \boldsymbol{w}_r^\top \boldsymbol{p}_u \boldsymbol{w}_r$, $\boldsymbol{q}_v^r = \boldsymbol{q}_v - \boldsymbol{w}_r^\top \boldsymbol{q}_v \boldsymbol{w}_r$. The lower the score of $f(u, r, v)$ is, the more likely the triplet is true, and vice versa. Note that relation-specific hyperplanes in TransH increase the embedding flexibility and diversity with modeling different users give same ratings to a item.

Given the scoring function on triplets, we further propose a margin-based loss by requiring that the real triplets should yield lower scores than the broken triplets, which help preserve the relative position between users and items:

$$\begin{aligned} L_{KG} = &\sum_{(u,r,v,v')} [f(u,r,v) - f(u,r,v') + \gamma]_+ \\ &+ \sum_{(u,r,v,u')} [f(u,r,v) - f(u',r,v) + \gamma]_+, \end{aligned} \tag{16}$$

where the triplet (u, r, v') is derived from the true triplet (u, r, v) by replacing the tail entity, which can be regarded as a broken triplet. Similarly, the head entity can be replaced by u' to generate another broken triplet (u', r, v). $[f(\cdot)]_+$ denotes $\max(0, f(\cdot))$.

It is worth noting that conventional methods adopt random sampling and replacement strategy to generate broken triplets. However, for rating prediction task, the broken triplets generated from such strategy are not guaranteed to be real broken triplets since they may occasionally to be true in testing data, which will mislead the L_{KG} in discriminating the true and broken triplets. Therefore, based on the above concerns, we propose a *rating-based cross sampling and replacement* strategy to generate broken triplets. Specifically, all the faked items/users in broken triplets should be sampled from the rated items/users whose ratings are smaller than the rating in the true triplets. For example, for user u_i, let $R_{V4}(i)$ denotes the item set rated by u_i with rating 4. Then given a triplet $(u_i, 4, v_j)$, we randomly sample $v_k \in \{R_{V3}(i) \cup R_{V2}(i) \cup R_{V1}(i)\}$ to obtain a new broken triplet $(u_i, 4, v_k)$. Meanwhile, we can also process the user u_k with the same strategy from $\{R_{U3}(j) \cup R_{U2}(j) \cup R_{U1}(j)\}$. In this way, we obtain broken triplet list for each true triplet and further feed them into Eq.(16) to constrain the model.

4.6 Model Training

Considering to enhance the flexibility of the wide and deep fusion framework, we allow the embedding parameters in the two models to be different, and train them with individual loss function separately as follows:

$$L_D = \sum_{(u,i)} (r_{ij} - \hat{r}_{ij}^D)^2 + \lambda_1 L_{KG} + \lambda_2 (\|\boldsymbol{U}\| + \|\boldsymbol{V}\| + \|\boldsymbol{e}_r\|) \tag{17}$$

$$L_W = \sum_{(u,i)} (r_{ij} - \hat{r}_{ij}^W)^2 + \lambda_3 (\|\boldsymbol{U}'\| + \|\boldsymbol{V}'\| + \|\boldsymbol{X}\| + \|\boldsymbol{Y}\|), \tag{18}$$

where λ_1, λ_2 and λ_3 are regularization parameters. After $\hat{r}_{ij}^D$ and $\hat{r}_{ij}^W$ are predicted from each view, the parameter of

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230972

last layer (a constant hyper-parameter or MLP) is trained through minimizing the total rating loss:

$$L_{fusion} = (r_{ij} - \lambda_{ij}\hat{r}_{ij}^D - (1-\lambda_{ij})\hat{r}_{ij}^W)^2. \tag{19}$$

For training wide and deep models from scratch, we adopt RMSProp and Adam, respectively. After feeding the predicted ratings $\hat{r}_{ij}^D$ and $\hat{r}_{ij}^W$ to the last fusion layer, we optimize it with linear search for optimal hyper-parameter or vanilla SGD for the parameters in MLP. More details about training process are given in the experimental part. As a matter of fact, the separate training schemes can provide more flexibility and representation capacity in comparison with traditional multi-view model training which requires different views share the embedding parameters.

Furthermore, in training the relational triplet constraint, we may obtain large-scale triplets, *e.g.*, the size of triplets in yelp can reach to 70 million. In order to train the relational triplets more efficiently, we propose a pre-training and initialization strategy. Specifically, we first pre-train the TransH model with all the rating relational triplets to obtain the embedding parameters of users, items and ratings. Then when formally turns to train the deep model, we use the pre-trained parameters to initialize the corresponding embeddings in the deep model. Moreover, we randomly select a smaller fixed size of triplets for each user-item pair sample, *e.g.*, $L_e = 30$, to constrain the deep model instead of using all the triplets in the training process.

5 Experiments

In this section, we conduct extensive experiments to evaluate the performance of the proposed model S4Rec. Specifically, we aim to answer the following questions:

- **RQ1.** How does S4Rec perform compared with other state-of-the-art social recommendation methods?
- **RQ2.** Does S4Rec outperform others under different data sparsity settings?
- **RQ3.** How do different modeling modules, *e.g.*, two-view modeling, relational triplet constraint affect the performance of S4Rec?
- **RQ4.** How do hyper-parameters impact the recommendation performance?

5.1 Experimental Settings

5.1.1 Datasets

We evaluate the performance of all the methods on three publicly representative datasets Epinions, Ciao[1] and Yelp[2], which all have social links along with rating data.

Epinions: Epinions is a consumer review website, where users can rate items from 1 to 5 as well as trust other users by adding them in trust list. This dataset provides user-item rating data and user-user directed trust relation.

Ciao: This dataset contains users' ratings to the items they have purchased and the social connections between them. The ratings are also in the range of 1 to 5.

Yelp: Yelp is a popular online review platform, where users can make friends and write reviews for restaurants as well as give ratings in the range of [1,5] . Here we extract a subset data of user-business rating data and social relations from the original dataset for social recommendation.

To alleviate the sparsity issues of the datasets, users who have less than 5 ratings are filtered out. We randomly split the rating dataset with 80%, 10%, 10% for training, validation and testing set, respectively. The statistics of these datasets are presented in Table 4 .

TABLE 4
The statistics of the datasets.

Statistics	Epinions	Ciao	Yelp
# users	23,251	7,371	21,461
# items	120,711	90,913	102,433
# ratings	613,574	268,174	894,435
rating density	0.0219%	0.0400%	0.0407%
# social relations	374,039	111,749	497,206
link density	0.0692%	0.2056%	0.1079%

5.1.2 Baselines

To verify the effectiveness of our proposed model, we compare it with three groups of methods, including DNN-based recommender systems, MF-based social recommender systems, and GNN-based social recommender systems. For each group, we select several representative methods as baselines and detail them in below. These baselines are implemented by official released code on GitHub or LibRec[3]. The optimal hyper-parameters in each method are set either by our experiments or suggested by original papers.

The first group mainly leverages user-item interaction data for rating prediction. We can use them to evaluate the effectiveness of other models which use the additional social information:

NeuMF [35]. This method is a state-of-the-art deep learning based framework that combines matrix factorization with a multi-layer perceptron model for ranking task. We replace the loss with mean square error to match our rating prediction task.

CUNE [24]. This method identifies top-k semantic friends for each user from user feedbacks through random walk, and use the embedded relations as social regularization term of rating matrix factorization.

IF-BPR [14]. This model identifies implicit friends with embedding representation learning on predefined metapaths and further ranks social items based on different types of friends through BPR approach. To adapt to our task, we add another MSE loss for rating prediction.

The second group includes three MF-based social recommendation models, which leverage both user-item interactions and social relations as input:

SocialMF [8]. This method incorporates the mechanism of trust propagation with matrix factorization technique for social recommendation.

TrustMF [5]. This method is another matrix factorization based model, factorizing both rating and trust data with sharing user latent vector.

1. https://www.cse.msu.edu/tangjili/datasetcode/truststudy.htm
2. https://www.yelp.com/dataset
3. https://github.com/guoguibing/librec

TABLE 5
Performance comparison of two metrics MAE and RMSE on three datasets for all methods. Best baselines are underlined and the best performance among all methods are in boldface. The proposed S4Rec model achieves the best performance in all cases.

Model	Epinions		Ciao		yelp	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
NeuMF	0.7903(±0.0039)	1.0707(±0.0113)	0.7168(±0.0110)	0.9832(±0.0135)	0.7885(±0.0027)	1.0281(±0.0037)
CUNE	0.8172(±0.0041)	1.0826(±0.0145)	0.7353(±0.0049)	1.0005(±0.0072)	0.7947(±0.0037)	1.0409(±0.0049)
IF-BPR	0.8045(±0.0016)	1.0707(±0.0026)	0.7302(±0.0054)	0.9957(±0.0075)	0.7867(±0.0025)	1.0346(±0.0032)
SocialMF	0.8254(±0.0032)	1.0789(±0.0056)	0.7482(±0.0052)	0.9890(±0.0076)	0.8037(±0.0034)	1.0368(±0.0045)
TrustMF	0.8298(±0.0028)	1.0954(±0.0062)	0.7416(±0.0055)	0.9917(±0.0081)	0.8149(±0.0048)	1.0680(±0.0052)
TrustSVD	0.8035(±0.0028)	1.0557(±0.0045)	0.7216(±0.0041)	<u>0.9575(±0.0065)</u>	0.7802(±0.0029)	<u>1.0085(±0.0040)</u>
ConsisRec	0.8029(±0.0023)	1.0541(±0.0036)	0.7252(±0.0043)	0.9581(±0.0069)	0.7829(±0.0023)	1.0145(±0.0037)
GraphRec	0.8029(±0.0019)	1.0656(±0.0037)	0.7253(±0.0069)	0.9850(±0.0081)	0.7900(±0.0028)	1.0254(±0.0035)
Danser	<u>0.7866(±0.0016)</u>	<u>1.0454(±0.0046)</u>	<u>0.7117(±0.0048)</u>	0.9641(±0.0055)	<u>0.7766(±0.0041)</u>	1.0118(±0.0031)
S4Rec-SocialMF	0.7838(±0.0031)	1.0441(0.0021)	0.7138(±0.0067)	0.9584(±0.0089)	0.7710(±0.0023)	1.0074(±0.0036)
S4Rec-TrustMF	0.7813(±0.0023)	1.0583(±0.0055)	0.7093(±0.0075)	0.9717(±0.0140)	0.7726(±0.0075)	1.0251(±0.0048)
S4Rec	**0.7754(±0.0028)****	**1.0396(±0.0046)****	**0.7007(±0.0052)****	**0.9498(±0.0063)****	**0.7643(±0.0023)****	**1.0026(±0.0028)****

** indicates the p-values of t-tests between S4Rec and the best baselines are smaller than 0.05.

TrustSVD [23] . This method is a state-of-the-art matrix factorization based social recommendation model, which takes the explicit and implicit influence of trusted users and rated items into consideration.

The third group contains three GNN-based social recommendation models:

ConsisRec [13]. This method empowers the GNN-based recommendation model by sampling consistent neighbors and employing relation attention in aggregation process.

GraphRec [10]. This method applies two GATs in user-item graph and social graph to simultaneously aggregates and fuse information for user embedding.

DANSER [11]. This method is a state-of-the-art GNN-based model, which proposes dual GATs to collaboratively learn representations for static and dynamic social effects in both user and item domains.

5.1.3 *Evaluation Metrics*

To evaluate the performance of all recommendation methods, we adopt the widely used Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) as metrics. The smaller values of MAE and RMSE indicate better performance. We repetitively conduct the experiments for 10 times and report the average performance in the testing dataset with the best epoch in the validation dataset. Note that although the relative improvements for some metrics or datasets are small, previous research [44] has pointed out that small improvements in MAE or RMSE can have a significant impact on the quality of top-K recommendation.

5.1.4 *Parameter Settings*

We implement our proposed model with PyTorch and deploy it on Titan RTX with 24G memory. The hyper-parameter settings for both three datasets are as follows: batch size $B = 256$, embedding sizes for deep and wide model are 80 and 10, dropout ratio $p = 0.5$, the regularization parameter of relational triplet constraint $\lambda_1 = 2$, other regularization terms $\lambda_2 = 0.0001$, $\lambda_3 = 0.05$. Sample size of DropNode in user-item bipartite graph $L_b = 30$, friends size in social graph $L_f = 20$, and we select top-20 users and items for implicit user and item relations, respectively. The deep and wide models are separately trained with learning rate 0.001 and 0.003. The Leaky ReLu slope is 0.2 in attention mechanism module. Furthermore, for simplicity we directly utilize the TrustSVD as the wide model in the experiment.

5.2 Comparative Results (RQ1)

The overall performance of all models on the three datasets are reported in Table 5. As can be seen from the table, NeuMF which only leverage the observed user-item rating matrix for modeling, and IF-BPR which incorporates implicit friends with BPR approach, perform much better than SocialMF and TrustMF. This phenomenon shows the necessity of sufficient and adequate mining of social relations for social recommendation. Besides, shallow model TrustSVD also performs better than ConsisRec and GraphRec with achieving balanced performance of the two metrics. Among all the baselines, DANSER performs the best via learning the multifaceted social effects for users and items. From the results of TrustSVD and DANSER, we can conclude that multifaceted user embedding from multi-views benefits to capture comprehensive user preferences and further improves rating prediction performance. Our propsoed S4Rec outperforms all the baselines on all the three datasets, which demonstrates the effectiveness of our model for rating prediction. In particular, S4Rec achieves 1.42%, 1.55% and 1.58% improvement on MAE compared with the second best one. Meanwhile, we also perform t-test on the absolute improvements and conclude that S4Rec yields significant improvements. What's more, when combining the proposed deep graph model with shallow models like SocialMF and TrustMF, the fusion models S4Rec-SocialMF and S4Rec-TrustMF beat the original MF-based model by a large margin on both two metrics, showing the generalization ability of the S4Rec framework over different shallow models. These comparative results illustrate that social relations indeed contains valuable information to improve recommendation accuracy, but inadequate fusion and insufficient mining of social relations deteriorate the recommendation performance instead.

5.3 Performance Under Different Data Sparsity (RQ2)

The data sparsity and imbalanced distribution greatly affect the training process and final recommendation performance. In this section, we would like to explore the model performance with respect to different sparsity conditions.

5.3.1 Rating Sparsity

Users have different number of ratings and hence can yield different recommendation performance. Thus, we examine the recommendation performance for users who have different number of historical ratings. Specifically, we divide users into different groups according to the number of ratings they give in the training set. For instance, [20,40) denotes users who have more than 20 ratings and less than 40 ratings. Then we calculate the average recommendation metrics within each group. Due to the space limit, we only show the results on MAE in Figure 3 and the trend is nearly the same on RMSE. The horizontal axis represents group size, and vertical axis shows the corresponding evaluation metric. Overall speaking, we observe that as more ratings data are used for modeling, the performance increases for most models on Epinions and yelp, while a little bit of fluctuations exist on Ciao. Among them, our proposed model still consistently outperforms all the baselines, no matter for cold-start or active user group. For example, for the cold-start users who have ratings less than 20, S4Rec improves 0.9%, 0.17% and 2.43% over the best baseline in Epinions, Ciao and yelp, respectively. When it comes to active users with more than 160 ratings, the improvement reaches to 2.6%, 0.81% and 2.25%. This demonstrates that our model is robust against the rating sparsity and the results are stable for different types of users.

5.3.2 Social Sparsity

Social sparsity explores the recommendation performance on users who have different number of social links. Analogously, we also group users according to the size of their social friends and further calculate the average evaluation metrics for each group. The results are shown in Figure 4, in which the horizontal axis depicts the size of the user group, *e.g.*, [5,10) means the group users have more than 5 social friends and less than 10. Similar trend can be observed that the recommendation performance increases with the growing number of social friends on Epinions and yelp, while a little bit of fluctuation on Ciao. On the whole, our proposed model improves other baselines with significant margin in various situations, showing its wide adaptability. For example, for the cold-start users in group [0,5), our model exceeds the best baseline by 1.35%, 1.97%, and 2.02% on Epinions, Ciao and yelp, respectively. While for the dense group [80,) with more than 80 social relations, the improvement comes to 2.57%, 0.07% and 2.41%, respectively.

5.4 Ablation Study (RQ3)

To validate the effectiveness of different modules and components in our proposed S4Rec, we conduct ablation studies in the following aspects.

TABLE 6
Ablation Study on different implicit relations and components.

	Epinions		**Ciao**		**yelp**	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
S4Rec	0.7724	1.0384	0.7007	0.9477	0.7643	1.0026
w/o implicit user	0.7770	1.0510	0.7074	0.9529	0.7654	1.0085
w/o implicit item	0.7855	1.0402	0.7089	0.9506	0.7714	1.0073
w/o Triplet	0.7890	1.0388	0.7094	0.9511	0.7706	1.0067
w/o Pretrain	0.7800	1.0385	0.7029	0.9495	0.7674	1.0076
w/o Cross	0.7923	1.0398	0.7193	0.9498	0.7654	1.0085

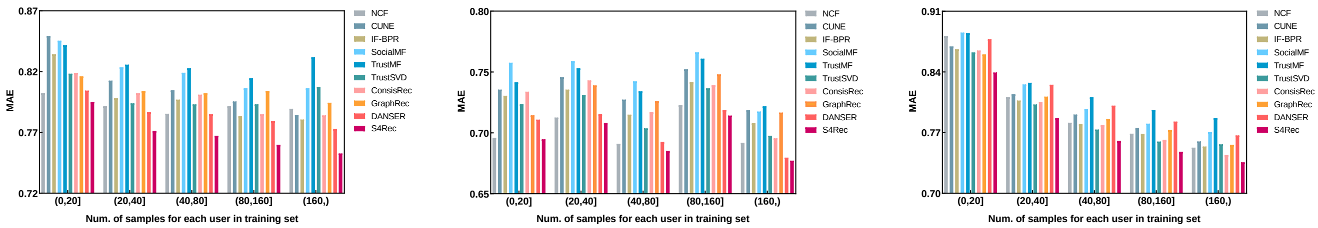
Effect of the Implicit Relations. Statistical analysis in Table 1 and 2 show that the implicit relations can provide as valuable information as the explicit social relations. In this subsection, we further conduct ablation experiments to check how different implicit relations affect the performance. Specifically, we remove the implicit relations individually from the model, e.g., *w/o implicit user* denotes removing extended implicit user relations for S4Rec, and the results are shown in Table 6. We can conclude from the table that both the implicit user and item relations contribute to improve the recommendation performance. Interestingly, the two types of implicit relations can have different impact on the MAE and RMSE. For example, the user and item relations further improve the base S4Rec model over 1.21% and 1.7% for RMSE and MAE on Epinions dataset, respectively. These results show that the augmented implicit relations are effective to capture diverse users and items preferences, and thus benefit for comprehensive representation learning.

Effect of the Relational Triplet Constraint. To better understand the role of the relational triplet constraint in S4Rec, we compare the performance of S4Rec with its three variants: *S4Rec w/o Triplet* eliminates the relational triplet constraint in the loss function; *S4Rec w/o Pretrain* removes the pre-training strategy by directly training the deep model with a fixed size of triplets; *S4Rec w/o Cross* replaces the proposed cross-sampling strategy with uniformly random sampling to generate relational triplets.

The experimental results are shown in Table 6. As can be seem from the table that when the relational triplet constraint is removed, the performance drops by -2.15%, -1.24% and -0.82% of MAE in the three datasets, which illustrates the necessity of the relational triplet constraint. Furthermore, we can also find that the pre-training strategy of the relational triplet and the initialization strategy can also help to improve the model performance, but the improvement magnitude depends on the data source. In addition, *S4Rec w/o Cross* performs the worst among all the variants in Epinions and Ciao, demonstrating the effectiveness of the cross-sampling strategy compared with the uniform sampling one. These findings shed light on applying the relational triplet constraint to other recommendation methods for further performance improvement.

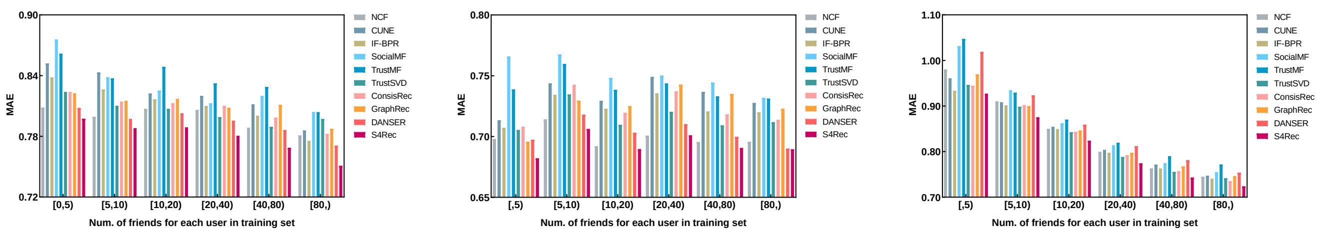
Effect of Two-view Fusion. To better reveal how each view can contribute to the performance, we compare S4Rec with its variant by modeling the single view respectively,

This article has been accepted for publication in IEEE Transactions on Knowledge and Data Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2022.3230972



(a) Epinions (b) Ciao (c) yelp

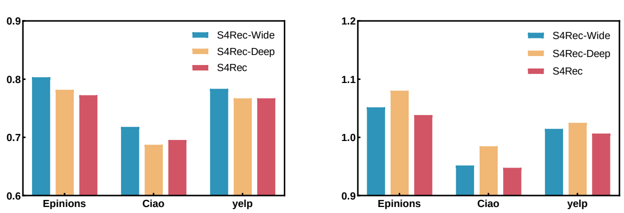
Fig. 3. Test performance of MAE under different rating sparsity on three datasets.



(a) Epinions (b) Ciao (c) yelp

Fig. 4. Test performance of MAE under different social sparsity on three datasets.

i.e., *S4Rec-Deep* (local-view) that only retains the deep graph model and *S4Rec-Wide* (global-view) which only considers the wide liner model. Note that we directly apply TrustSVD as a proxy for the *S4Rec-Wide* for simplicity. The comparative results are listed in Figure 5. We observe that the Wide-view model benefits stability modeling with smaller RMSE, while the Deep-view model delicates to accuracy modeling with smaller MAE. Taking the Epinions dataset as an example, Deep-view has +2.7% gain than Wide-view on MAE, and -2.72% gain on RMSE. Interestingly, when the two views are combined, the fusion model perfectly maintains the advantages of them two and achieves balanced performance in both MAE and RMSE. In sum, as a core contribution of our proposed model, the experiments have demonstrated that the fusion modeling strategy is capable of combining the strengths of both two views to build a more accurate and stable recommender system.



(a) MAE (b) RMSE

Fig. 5. Test performance of two views of S4Rec on MAE and RMSE.

5.5 Hyper-parameters Sensitivity (RQ4)

In this subsection, we investigate how the performance of our proposed model varies *w.r.t* some hyper-parameters including the embedding size d of deep model, DropNode size L_b in user-item interaction graph and L_f in social graph, and regularization parameter l_2. In addition, we only show the performance on the Epinions dataset due to the space limit. We omit the analysis on other hyper-parameters because they have less impact on model performance.

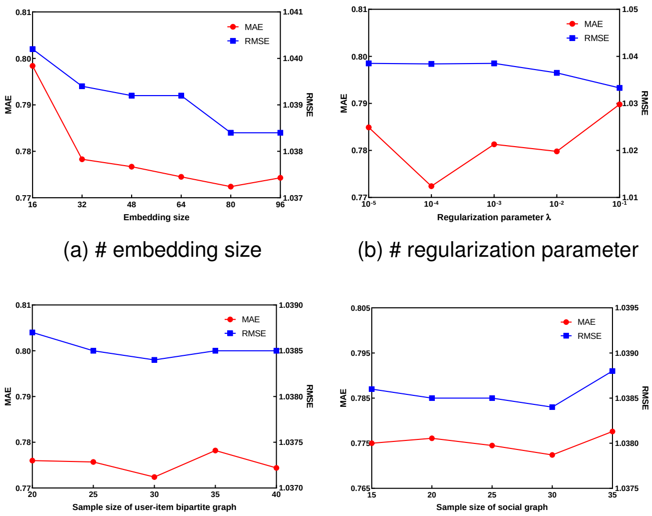
Impact of Embedding Size of the Deep Graph Model. Embedding size is associated with model's capacity. Figure 6a shows that with the increase of embedding size of the deep model, the overall performance improves due to the better expressiveness. However, when d reaches a certain scale (80 in our case), the marginal improvement is less significant and the computation complexity increases remarkably instead. Therefore, a proper embedding size is needed to guarantee both model capacity and complexity.

Impact of the Regularization Parameter. The regularization parameter plays an important role to adjust the performance, where either large or small regularization term cannot ensure an optimial recommendation performance. As shown in Figure 6b, a suitable setting, *i.e.*, $l_2 = 1e-4$ helps to regularize the model to achieve a satisfactory result.

Impact of Sample Size in DropNode. We also examine the impact of the sample size of DropNode applied in user-item interaction graph and social graph, and the performance is shown in Figure 6c and 6d, respectively. As can be seen, with the sample size increases, the performance becomes better since more diverse samples are involved to learn user and item representations. However, when it comes to a large size, the learned representations become more biased and mixed, which instead deteriorates the performance and meanwhile incurs heavy computational cost. Hence an appropriate sample size ensures both the prediction accuracy and training efficiency.

6 Conclusion

In this paper, we present a semantic and structural view fusion framework, i.e., S4Rec, for social recommendation,



(a) # embedding size (b) # regularization parameter

(c) # rated users/items (d) # friends

Fig. 6. Performance on Epinions w.r.t different hyper-parameters.

which deeply undermines and integrates information from user-item interaction graph and social graph. The reasons why the framework gets a good performance can be attributed to: 1) the proposed collective intelligence-based implicit user and item relations are in favor of constructing multiple semantic metapaths for comprehensive user and item representation learning; 2) a fusion of deep graph and wide linear models corresponding to local and global-view modeling on the graph structures of two domains benefits to achieve both accurate and stable rating prediction; 3) the designed (User, Rating, Item) triplet on the newly raised relative position difference problem provides a good generalization strategy to improve the overall performance. Extensive comparative experiments and ablation studies demonstrate the effectiveness of our proposed framework, and outperforms under different data sparsity settings.

For the future work, we plan to further investigate the proposed relative position difference problem with more experiments in order to make it clear that when and how the triplet constraint enhances the model performance, and whether it can be applied in other scenarios. Besides, we also intend to explore the explainable social recommendation by explicitly claiming which interacted items and social users dominate the current recommendation through the proposed two-view fusion framework.

Acknowledgments

This work was partially supported by the National Key R&D Program of China (2019YFB2101804). Dr. Kun Yuan's work was supported by National Natural Science Foundation of China (NSFC) under Grant 72101076, 72171071, and the Fundamental Research Funds for the Central Universities under Grant JZ2021HGQA0206. Dr. Guannan Liu's work was supported by NSFC under Grant 71701007, 92046025. Dr. Junjie Wu's work was supported in part by NSFC under Grant 71725002, 72031001. Dr. Hui Xiong's work was supported in part by Foshan HKUST Projects (FSUST21-FYTRI01A, FSUST21-FYTRI02A).

References

[1] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social network analysis and mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[2] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[3] P. V. MARSDEN and N. E. FRIEDKIN, "Network studies of social influence," *Sociological Methods & Research*, vol. 22, no. 1, pp. 127–151, 1993.

[4] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: Social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008, p. 931–940.

[5] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1633–1647, 2017.

[6] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, p. 203–210.

[7] H. Li, D. Wu, W. Tang, and N. Mamoulis, "Overlapping community regularization for rating rrediction in social recommender systems," in *Proceedings of the 9th ACM Conference on Recommender Systems*, 2015, p. 27–34.

[8] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the 4th ACM conference on Recommender systems*, 2010, pp. 135–142.

[9] H. Fang, Y. Bao, and J. Zhang, "Leveraging decomposed trust in probabilistic matrix factorization for effective recommendation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, Jun. 2014.

[10] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, 2019, pp. 417–426.

[11] Q. Wu, H. Zhang, X. Gao, P. He, P. Weng, H. Gao, and G. Chen, "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *The World Wide Web Conference*, 2019, pp. 2091–2102.

[12] H. Chen, H. Yin, T. Chen, W. Wang, X. Li, and X. Hu, "Social boosted recommendation with folded bipartite network embedding," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[13] L. Yang, Z. Liu, Y. Dou, J. Ma, and P. S. Yu, "Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, p. 2141–2145.

[14] J. Yu, M. Gao, J. Li, H. Yin, and H. Liu, "Adaptive implicit friends identification over heterogeneous network for social recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 357–366.

[15] H. Liu, L. Jing, J. Yu, and M. K.-P. Ng, "Social recommendation with learning personal and social latent factors," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[16] Z. Wang, J. Zhang, J. Feng, and Z. Chen, "Knowledge graph embedding by translating on hyperplanes," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.

[17] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2022.

[18] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, 2011, p. 287–296.

[19] A. Krishnan, H. Cheruvu, C. Tao, and H. Sundaram, "A modular adversarial approach to social recommendation," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1753–1762.

[20] C. Chen, M. Zhang, C. Wang, W. Ma, M. Li, Y. Liu, and S. Ma, "An efficient adaptive transfer neural network for social-aware recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 225–234.

[21] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users,"

in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2017, pp. 185–194.

[22] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, p. 726–735.

[23] G. Guo, J. Zhang, and N. Yorke-Smith, "Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015.

[24] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, "Collaborative user network embedding for social recommender systems," in *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*, 2017, pp. 381–389.

[25] R. v. d. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.

[26] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, p. 974–983.

[27] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 27–34, Apr. 2020.

[28] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui, "Enhance social recommendation with adversarial graph convolutional networks," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[29] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 413–424.

[30] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 235–244.

[31] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, and M. Wang, "Diffnet++: A neural influence and interest diffusion network for social recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[32] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *Proceedings of the Web Conference 2021*, ser. WWW '21, 2021, p. 413–424.

[33] Y. Li, M. Yang, and Z. Zhang, "A survey of multi-view representation learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 10, pp. 1863–1883, 2018.

[34] W. Cheng, Y. Shen, Y. Zhu, and L. Huang, "Delf: A dual-embedding based deep latent factor model for recommendation," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 7 2018, pp. 3329–3335.

[35] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 173–182.

[36] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 7–10.

[37] C.-C. Hsu, M.-Y. Yeh *et al.*, "A general framework for implicit and explicit social recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 12, pp. 2228–2241, 2018.

[38] M. Jamali and M. Ester, "Trustwalker: A random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, p. 397–406.

[39] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[40] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[41] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," in *International Conference on Learning Representations*, 2020.

[42] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1025–1035.

[43] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of KDD cup and Workshop*, 2007, pp. 5–8.

[44] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 426–434.

Kun Yuan is currently an assistant professor at the Hefei University of Technology (HFUT), Hefei, Anhui, China. He received Ph.D. degree from the Beihang University, China. His general area of research interests is data mining, recommender systems and social network analysis. His work has published in the journal of ACM TKDD, ACM TIST, IP&M, etc.

Guannan Liu is currently an Associate Professor in the Department of Information Systems with Beihang University, Beijing, China. He received the Ph.D. degree from Tsinghua University, China. His research interests include data mining, business intelligence, and recommender systems. His work has been published in the journal of IEEE TKDE, ACM TKDD, ACM TIST, Decision Support Systems, etc., and also in the conference proceedings such as KDD, ICDM etc.

Junjie Wu received his Ph.D. degree in Management Science and Engineering from Tsinghua University. He is currently a full Professor in Information Systems Department of Beihang University, the director of the Research Center for Data Intelligence (DIG), and the director of the Institute of Artificial Intelligence for Management. His general area of research is data mining and complex networks. He is the recipient of NSFC Distinguished Young Scholars award and MOE Changjiang Young Scholars award in China.

Hui Xiong (F'20) is currently a Chair Professor at the Hong Kong University of Science and Technology (Guangzhou). Dr. Xiong's research interests include data mining, mobile computing, and their applications in business. Dr. Xiong received his PhD in Computer Science from University of Minnesota, USA. He has served regularly on the organization and program committees of numerous conferences, including as a Program Co-Chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a Program Co-Chair for the IEEE 2013 International Conference on Data Mining (ICDM), a General Co-Chair for the 2015 IEEE International Conference on Data Mining (ICDM), and a Program Co-Chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He received the 2021 AAAI Best Paper Award and the 2011 IEEE ICDM Best Research Paper award. For his outstanding contributions to data mining and mobile computing, he was elected an AAAS Fellow and an IEEE Fellow in 2020.