Enhancing Social Recommendation with Adversarial Graph Convolutional Networks

Junliang Yu, Hongzhi Yin*, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui

Abstract—Social recommender systems are expected to improve recommendation quality by incorporating social information when there is little user-item interaction data. However, recent reports from industry show that social recommender systems consistently fail in practice. According to the negative findings, the failure is attributed to: (1) A majority of users only have a very limited number of neighbors in social networks and can hardly benefit from social relations; (2) Social relations are noisy but they are indiscriminately used; (3) Social relations are assumed to be universally applicable to multiple scenarios while they are actually multi-faceted and show heterogeneous strengths in different scenarios. Most existing social recommendation models only consider the homophily in social networks and neglect these drawbacks. In this paper we propose a deep adversarial framework based on graph convolutional networks (GCN) to address these problems. Concretely, for (1) and (2), a GCN-based autoencoder is developed to augment the relation data by encoding high-order and complex connectivity patterns, and meanwhile is optimized subject to the constraint of reconstructing the social profile to guarantee the validity of the identified neighborhood. After obtaining enough purified social relations for each user, a GCN-based attentive social recommendation module is designed to address (3) by capturing the heterogeneous strengths of social relations. Finally, we adopt adversarial training to unify all the components by playing a Minimax game and ensure a coordinated effort to enhance recommendation performance. Extensive experiments on multiple open datasets demonstrate the superiority of our framework and the ablation study confirms the importance and effectiveness of each component.

Index Terms—Recommender Systems, Social Recommendation, Adversarial Training, Graph Neural Networks.

1 Introduction

Recommender systems are often compromised by the sparsity of data as users can only consume a tiny fraction of items, which leads to a frustrating result that in many cases recommender systems fail to fulfill their intrinsic capacity. With the popularity of social platforms, social relations are often used to in mitigate the problem of data sparsity since people's decisions are often influenced by their friends, according to the principle of *homophily* [1]. Suggested by this theory, a large number of social recommendation models [2], [3], [4], [5], [6], [7], [8], [9], which exploit explicit social information to improve general recommendation, have been developed, showing decent improvements under ideal circumstances. However, the real situations are rather complex and follow-up investigations [10] from the industry revealed that the effectiveness of applying explicit social information into recommender systems had been overestimated.

In summary, the failure of social recommendation can be attributed to: (1) The observed social relations are extremely sparse, and users with few ratings/purchase records are also likely to have few social relationships [11]. Therefore it is questionable to claim that the majority of users who have limited purchase data can benefit from limited social relations. (2) The low cost of building social connections and the openness nature of social networks enable malicious accounts to easily infiltrate, making social networks noisy [12]. But social relations are often indiscriminately used in most social recommendation models. (3) Social relations have multi-facets and show heterogeneous strengths in different situations [13] and are not universally applicable to any contexts. But most prior studies assume that users can reach a consensus with their friends in all the aspects of preferences. In view of these findings, it is less controversial that researchers should change their way of leveraging social relations and develop new approaches to better exploit the imperfect social network.

Having been aware of the adverse factors of social recommendation, a few studies had made attempts to overcome these drawbacks. For example, to alleviate the influence of relation noises, a series of work [14] aiming to distill the observable social networks was proposed in quick succession. Instead of using all the relations, they propose to extract useful explicit social links or divide social connections into fine-grained classes and only make use of the filtered information. However, in consideration of the high sparsity of original social networks, the extracted relations are too sparse to effectively improve recommendation performance. Meanwhile, an opposite way focusing on augmenting available relations was also explored. Researchers who follow this way paid attention to discovering potential but reliable relations for each user and identify those users who are not directly connected to the current user but

- J. Yu, H. Yin, and Z. Huang are with the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Queensland, Australia.
E-mail: jl.yu@uq.edu.au, h.yin1@uq.edu.au, huang@itee.uq.edu.au
- J. Li is with Department of Electrical and Computer Engineering, Department of Computer Science, and School of Data Science, University of Virginia, Charlottesville, Virginia, US.
E-mail: jundong@virginia.edu
- M. Gao is with the School of Big Data & Software Engineering, Chongqing University, Chongqing, China.
E-mail: gaomin@cqu.edu.cn
- L. Cui is with the School of Software, Shandong University, Jinan, Shandong, China.
E-mail: clz@sdu.edu.cn

*Corresponding author and having equal contribution with the first author.

share the similar interests with her as the *implicit* friends [15], [16], [17]. Following their ways, the sparsity of explicit social relations can be alleviated but on the other hand the augmentation also inevitably introduces noises. Specifically, in these methods, the friend-searching strategies are not coupled with the recommendation process, which questions the validity of the implicit friends for recommendation. Besides, none of these augmentation-based methods pay attention to the multi-facets problem of social relations

To overcome the limitations of prior research on social recommender systems. We propose to exploit the implicit social relations to enhance social recommendation. Technically, we present a deep adversarial framework (shown in Fig, 1) based on graph convolutional networks (GCN) [18] to achieve our goals. We unfold our research by answering the following three questions:

(1) How does our framework address the problem of relation sparsity? The studies focusing on the identification of implicit relations have set a good example for us that searching for unobserved but helpful connections upon the known social networks/bipartite networks is practicable [11], [16], [19]. However, we claim that such models struggle to capture higher-order and complex connectivity patterns among nodes, because they only draw on information from immediate neighborhood or linear contexts (e.g. random walk based neighbors). Actually, in social networks, it has been shown to be beneficial to consider the higher-order structure [20]. In light of this, we follow their findings and further design some *motifs* [21] which are local structures involving multiple nodes to help uncover the high-order social information. To fully exploit the motifs, a tailored GCN is employed to aggregate information from motif-induced neighborhood and outputs a certain number of new relations, which are called the alternative neighborhood in this paper. By doing so, each user, especially those who have a very small number of relations, can benefit.

(2) How can our framework guarantee the validity of the alternative neighborhood? The identification process of the new neighbors is unsupervised and hence it is hard to assess the validity of the alternative neighborhood before used for recommendation. Thus we adopt the idea of autoencoder and investigate if we can use these new neighbors to reconstruct the explicit social profiles. As shown in Fig. 1, we concatenate the motif-based GCN with a multilayer perception (MLP) and make them act as a concrete autoencoder [22] for relation denoising. Technically, the GCN works as an encoder while the MLP acts as an decoder. By mapping the alternative neighborhood to part of the explicit social profiles, which are considered to be reliable, we believe that the informative and important information of the social network has been encoded into these the new neighbors. Intuitively, this design allows users to filter irrelevant relations or noises and guarantees the validity of the alternative neighborhood.

(3) How does our framework cope with the multi-facets problem of social relations? After obtaining a certain number of helpful alternative neighbors for each user, we can replace the explicit relations with the new neighborhood. Then the multi-facets problem is the next to be handled. As the social relation may show heterogeneous strengths in different contexts [13], it is necessary to develop a recommendation module that can selectively exploit the social information. To this end, we introduce an attentive social embedding propagation layer to our GCN based recommendation module in which a social attention mechanism is employed to weigh the contribution of the new neighbors and selectively aggregate information. Specifically, the attention mechanism is coupled with the context and then the importance of the social relations varies from context to context, enabling our framework to capture the heterogeneous strengths of trust to improve recommendation performance.

In summary, our proposed framework includes three main stages: alternative neighborhood generation, neighborhood denoising, and attention-aware social recommendation, which are corresponding solutions to the above three questions. To unify all the tasks and components and ensure a coordinated effort against all the challenges, we finally adopt adversarial training [23] to enhance our framework. In our adversarial framework, the motif-based GCN acts as a generator that produces the alternative neighborhood, while the attentive social recommendation module, works as the discriminator to recognize the informative connections and enforces the generator to learn the real distribution of the relations that can improve recommendation performance. With the competition between the generator and the discriminator, they mutually enhance each other. As a result, the whole framework will show better recommendation performance. Overall, the major contributions of this paper are summarized as follows:

- We design three different components to systematically and effectively deal with the three tough problems in social recommender systems.
- We unify and intensify these three components by integrating them into a deep adversarial framework based on GCNs. To the best of our knowledge, we are the first to combine adversarial training and graph neural networks for social recommendation.
- We conduct extensive experiments on multiple real-world datasets to demonstrate the superiority of the proposed framework and show the effectiveness of each component.

The remainder of this paper is organized as follows. In Section 2, the related work is presented. Section 3 illustrates the proposed adversarial social recommendation framework. The experimental studies are presented in Section 4. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Social Recommendation

The principle of homophily [1] lays the theoretical basis for the majority of social recommendation models. Among the early attempts of social recommendation, the leading ideas can be categorized into three groups including co-factorization methods, ensemble methods, and regularization methods [10]. As the representative work of co-factorization methods, SoRec [24] and TrustMF [25] jointly co-factorize the rating and social matrices and project rating and social contexts into the same latent space. STE [26] and mTrust [13], as the typical ensemble methods, regard user's primary preference as the linear combination of ratings from the user and her social network. Another line of work, social regularization model [4], proposes to narrow the preference gap between users and their friends by using weighted social regularization terms. Besides, follow-up studies also explore social impact on users' preferences from other perspectives. The model of [27] use social connections to capture users' exposure instead of preference. Yin *et al.* [6], [28] integrated social relations into point-of-interest (POI) recommendation. SBPR [29] leverages social connections to model the item ranks in personalized ranking. Group recommendation models [30], [31] also borrow the strength of social relations to overcome the data sparsity in their scenarios.

After the frustrating experience of applying social recommender systems in industry has been noticed [10], a few research efforts shift attention to addressing the practical problems of social recommendation. Based on the weak tie theory, Wang *et al.* [14] proposed fine-grained models to distinguish strong ties and weak ties. Inspired by the wide use of network embedding techniques, CUNE [11] and IF-BPR [16] adopted homogeneous and heterogeneous network embeddings to capture reliable implicit social relations, respectively. InSRMF [32] takes advantage of the indirect social relations detection and collaborative filtering on social network and rating behavior information to mitigate the relation sparsity. However, most of these studies only partially overcome the challenges of social recommendation.

2.2 Graph Neural Networks in Recommender Systems

Recently, graph neural networks (GNNs) [18] have received increasing attention due to their great capacity in various tasks. Inspired by the success in other fields such as node classification and link prediction, researchers also investigated the suitability of GNNs for the task of recommendation. Representative work like NGCF [33] and LightGCN [34] have proven that GNN can significantly enhance general recommendation. As the message passing among nodes in GNNs is well aligned with the diffusion of social information, it is natural to adopt GNNs to social recommendation. Wu *et al.* [8] developed a deep influence propagation model called DiffNet to simulate the recursive social diffusion process. Later, DiffNet++ [35], the extension of DiffNet, was proposed to model both the neural influence diffusion and interest diffusion in a unified framework. Besides, Fan *et al.* proposed GraphRec [7] to learn user representations by fusing first-order social and first-order item neighbors. Wu *et al.* [36] developed a dual graph attention networks to collaboratively learn representations for two-fold social effects. Song *et al.* proposed DGRec [37] to model both users' session-based interests as well as dynamic social influences. Kim *et al.* [38] proposed a graph-based social recommendation method that works on a group-user-item tripartite attributed multiplex heterogeneous graph to reduce noise and meanwhile paid attention to the oversmoothing problem of GNNs. Despite the effectiveness reported in these work, they either only address one of the problems discussed in Section 1, or neglect the limitations of directly using social information, resulting in a suboptimal performance.

2.3 Adversarial Training in Recommender Systems

Generative adversarial networks (GANs) [23] have led a revolution in many fields including recommender systems. The most popular paradigm of applying adversarial training to recommender systems is that, by playing a Minimax game, the discriminator guides the generator towards fitting the underlying relevance distribution over items of the given user, while the generator generates difficult examples to confuse and then improve the discriminator [39], [40], [41], [42], [43]. IRGAN [39] is the first GAN-based recommendation model which unifies the discriminative models and generative models. GraphGAN [40] designs a graph softmax function to accelerate training and improve computing efficiency. Wang *et al.* [41] employed an adaptive negative sampling framework to optimize the proposed streaming recommendation model. Besides, instead of generating samples, Chae *et al.* [42] proposed to learn user profiles with GAN. He *et al.* [44] made the first attempt to add adversarial perturbations to the latent factor to avoid overfitting. Additionally, [45], [46] explore a new application of GAN by augmenting user-item interactions to improve collaborative filtering.

There are also a few studies combining adversarial training and social recommendation [47], [48], [19]. Krishnan *et al* [47] proposed a modular adversarial framework to disentangle the architectural choices for the recommender and social representation models. Fan *et al.* [48] adopt a bidirectional mapping method to transfer users' information between social domain and item domain using adversarial training. Among these work, RSGAN [19] is the most similar to our proposed model in terms of the motivation. This work also focuses on identifying reliable relations with adversarial training to improve social recommendation. However, it only covers the relation sparsity problem and the scarcity of labeled reliable friends for supervised learning is its bottleneck which limits the model's capacity.

3 Proposed Framework

In this paper we present our GCNs-based deep adversarial social recommendation framework. Figure 1 shows the schematic overview of our framework. We first present how each component in our framework deals with the aforementioned problems of social recommendation and then show how adversarial training unifies and intensifies all the components by playing a Minimax game.

3.1 Notations and definitions

Let $U = \{u_1, u_2, ..., u_m\}$ and $I = \{i_1, i_2, ..., i_n\}$ respectively be the sets of users and items in recommender systems, where m is the number of users, and n is the number of items. $\mathcal{N}(u)$ is the set of items which were clicked/consumed by user u and $\mathcal{N}(i)$ is the user set including the users who clicked/purchased item i. $\mathbf{Y} \in \mathbb{R}^{m \times n}$ is the feedback matrix. For each pair (u, i), y_{ui} denotes the user's feedback on the item and $\hat{y}_{ui}$ denotes the predicted score. In this paper, we focus on Top-N recommendation. Following the convention, y_{ui} is either 1 (positive) or 0 (negative or unknown). For each user pair (u_1, u_2), $s_{u_1,u_2} = 1$ indicates that u_1 follows u_2 in the social network. It should

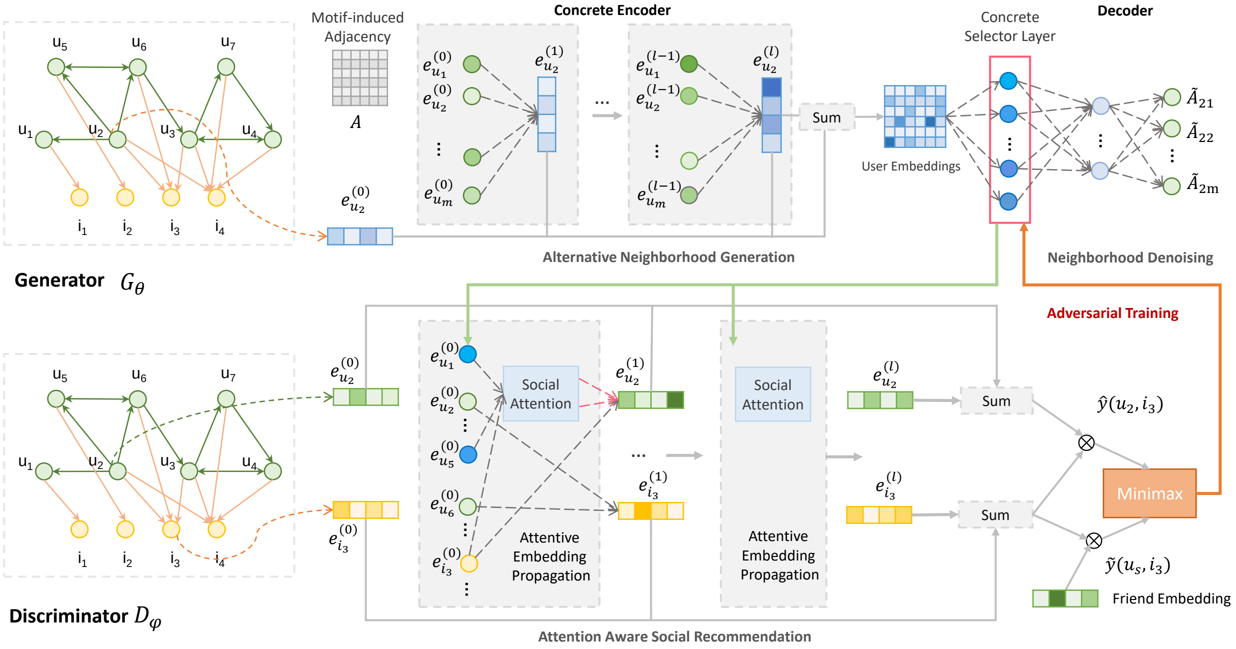


Fig. 1. The schematic overview of our framework. The arrows with solid line indicate the flow of information and the arrows with dashed line represent the computational operations of the neural networks.

be noted that we work on directed graphs and the relation matrix $\mathbf{S} \in \mathbb{R}^{m\times m}$ is asymmetric. The low dimensional vector $\mathbf{e}_{u_1}^{(l)}$ of dimension d represents the node embedding where the superscript l denotes the l-th layer and the subscript denotes the node. In this paper, we use bold capital letters to denote matrices and bold lowercase letters to denote vectors.

3.2 Alternative Neighborhood Generation

In this paper, alternative neighborhood refers to a group of users who share similar preferences with the specified user. Conceptually, it is literally like the *implicit friends* proposed by [15], [16], but actually a superset of them. For users with few social connections, the alternative neighborhood may be totally new neighbors, while partially overlaps with the neighbors of the users with a large number of social relations. We interchangeably call it alternative neighborhood/alternative neighbors/alternative relations in the remainder of this paper. Technically, prior studies [15], [16] identify implicit friends by means of rating similarity computation and network embedding techniques. But they struggle to capture the high-order and complex connectivity pattern among users because they only explore linear contexts (e.g. random walk based neighbors). In light of this, in our framework, we exploit the *motifs* for alternative neighborhood identification. It should be noted that, the high-order information in existing GCN-based models such as NGCF [33] and DiffNet++ [35] refers to the distant or multi-hop neighbors which are usually collected by Depth First Search. Being different from these existing literatures, the high-order in our work means relations beyond pairwise which cannot be captured with the conventional methods such as DFS.

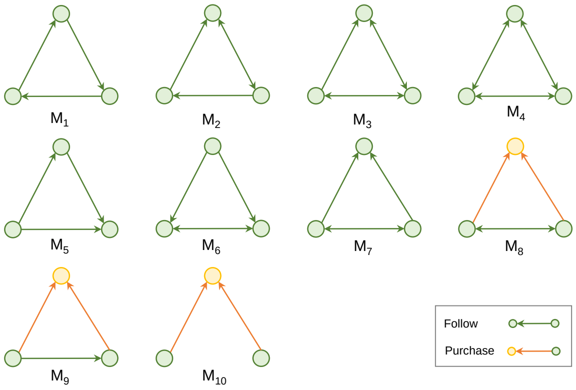


Fig. 2. Motifs used in our work. The green circles denote users and the yellow circles denote items.

3.2.1 High-order Social Information Exploitation

Motif, which was first introduced in [21], is the specific local structure involving multiple nodes [20], which are shown to be useful in many applications such as social computing [21]. In this paper, we only focus on triangular motifs because of the widespread triadic closure in social networks. Fig. 2 shows the used triangular motifs in which $\mathcal{M}_1 - \mathcal{M}_7$ are homogeneous and $\mathcal{M}_8 - \mathcal{M}_{10}$ are heterogeneous. According to [20], $\mathcal{M}_1 - \mathcal{M}_7$ are crucial in social computing. To leverage the shared purchase preferences, we further design $\mathcal{M}_8 - \mathcal{M}_{10}$ for our application. Specifically, the common purchases in $\mathcal{M}_8$ and $\mathcal{M}_9$ are used to strengthen the established relations, and $\mathcal{M}_{10}$ helps link users who are not directly connected via social relations. Obviously, introducing purchase information to social domain facilitates similarity measuring when social relations are very sparse.

The motif-induced adjacency matrix $\mathbf{A}_{M_k}$ represents the frequency of two nodes appearing in a given motif $\mathcal{M}_k$, which is computed by:

$$(\mathbf{A}_{M_k})_{i,j} = \sum_{i \in U, j \in U} 1\ (i, j\ occur\ in\ \mathcal{M}_k). \tag{1}$$

For example, given the motif $\mathcal{M}_8$, we can observe that, in the graph in Fig. 1, (u_3, u_4, i_4) forms one instance of $\mathcal{M}_8$, thus $(\mathbf{A}_{M_8})_{u_3,u_4} = 1$. Counting motifs in a network may be time-consuming. But as we only exploit triangular motifs, following [49], we show that motif-induced adjacency matrix can be computed based on simple matrix computation. Let $\mathbf{B} = \mathbf{S} \odot \mathbf{S}^T$ and $\mathbf{U} = \mathbf{S} - \mathbf{B}$ be the adjacency matrix of the bidirectional and unidirectional links of the explicit social network respectively, where $\odot$ denotes element-wise product. The computation of all the motif-induced adjacency matrices is summarized in Table 1.

TABLE 1
Computation of motif-induced adjacency matrices.

Motif	Matrix Computation
$\mathcal{M}_1$	$\mathbf{C} = (\mathbf{UU}) \odot \mathbf{U}^T$
$\mathcal{M}_2$	$\mathbf{C} = (\mathbf{BU}) \odot \mathbf{U}^T + (\mathbf{UB}) \odot \mathbf{U}^T + (\mathbf{UU}) \odot \mathbf{B}$
$\mathcal{M}_3$	$\mathbf{C} = (\mathbf{BB}) \odot \mathbf{U} + (\mathbf{BU}) \odot \mathbf{B} + (\mathbf{U} \cdot \mathbf{B}) \odot \mathbf{B}$
$\mathcal{M}_4$	$\mathbf{C} = (\mathbf{BB}) \odot \mathbf{B}$
$\mathcal{M}_5$	$\mathbf{C} = (\mathbf{UU}) \odot \mathbf{U} + \left(\mathbf{UU}^T\right) \odot \mathbf{U} + \left(\mathbf{U}^T\mathbf{U}\right) \odot \mathbf{U}$
$\mathcal{M}_6$	$\mathbf{C} = (\mathbf{UB}) \odot \mathbf{U} + \left(\mathbf{BU}^T\right) \odot \mathbf{U}^T + \left(\mathbf{U}^T\mathbf{U}\right) \odot \mathbf{B}$
$\mathcal{M}_7$	$\mathbf{C} = \left(\mathbf{U}^T\mathbf{B}\right) \odot \mathbf{U}^T + (\mathbf{BU}) \odot \mathbf{U} + \left(\mathbf{UU}^T\right) \odot \mathbf{B}$
$\mathcal{M}_8$	$\mathbf{C} = \left(\mathbf{YY}^T\right) \odot \mathbf{B}$
$\mathcal{M}_9$	$\mathbf{C} = \left(\mathbf{YY}^T\right) \odot \mathbf{U}$
$\mathcal{M}_{10}$	$\mathbf{C} = \left(\mathbf{YY}^T\right)$

As can be seen, all the formulations are almost based on a general operation $(\mathbf{PQ}) \odot \mathbf{T}$ (Here $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{T}$ are general notations denoting different matrices) which can be efficiently calculated since all the involved matrices are very sparse. For symmetric motifs, $\mathbf{A}_M = \mathbf{C}$ and for the asymmetric ones, $\mathbf{A}_M = \mathbf{C} + \mathbf{C}^T$. It is easy to understand these formulations as the operation $\mathbf{PQ}$ constructs paths connecting the three vertices in a triangular motif and the operation $\odot\mathbf{T}$ complements the motif with the missing edge with a certain direction and closes the triangle. With the motif-induced adjacency matrices, we are able to leverage the high-order connectivity patterns in the social network and the user-item relations. Particularly, $\mathbf{A}_{M_{10}}$ is a complement that uses common purchase to establish implicit relations when users have few relations. It should be noted that we only preserve elements that are larger than 5 in $\mathbf{A}_{M_{10}}$ to avoid noises.

3.2.2 Neighborhood Generation with Motif-based GCN

Due to the great capacity to capture the dependence of nodes, we tailor the classical GCN [18] to aggregate information from the explicit and motif-induced neighborhoods for the generation of the alternative neighborhood. Formally, a general GCN is constructed using the following layer-wise propagation [18]:

$$\mathbf{E}^{(l+1)} = \sigma\left(\mathbf{Z}\mathbf{E}^{(l)}\mathbf{W}^{(l)}\right),\ \mathbf{Z} = \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}, \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ ($\mathbf{I}$ represents an identity matrix with size M) with added self-loop, $\mathbf{D}$ is the diagonal degree matrix of $\tilde{\mathbf{A}}$, and σ is the nonlinear activation function. However, it is reported in the latest studies [34], [50] that two most common designs in GCNs, feature transformation and nonlinear activation function, contribute little to the performance of collaborative filtering (CF). That is because GCN is originally proposed for node classification on attributed graph, where each node has rich attributes as input features, while in CF, each node (user or item) is only described by a one-hot ID and there is no concrete semantics behind the identifiers. In this paper, we following the design of LightGCN [34] and remove these two operations in the general GCN. As for the adjacency matrix, we defined it as follows:

$$\mathbf{A} = \mathbf{S} + \mathbf{A}_{M_1} + \cdots + \mathbf{A}_{M_{10}}. \tag{3}$$

$\mathbf{S}$ is used because that motif-induced social network might cause some isolated nodes. To involve these isolated nodes which cannot build any motifs with other nodes, we need to incorporate the original relation adjacency matrix. At each layer, each user's node representation is refined by a weighted sum of the features of its motif-induced neighborhood. Formally, the neighborhood aggregation is defined as:

$$\begin{aligned} \mathbf{e}_{u_i}^{(l+1)} &= \sum_{j \in \mathcal{N}_{(u_i)}} a_{ij}\mathbf{e}_{u_j}^{(l)}. \\ \mathbf{E}_\theta^{(l+1)} &= \mathbf{D}_A^{-1}\mathbf{A}\mathbf{E}_\theta^{(l)}. \end{aligned} \tag{4}$$

where $\mathbf{E}_\theta$ is the embeddings of all users. A more flexible solution may be allowing each node to choose the most beneficial motif-induced neighborhood to integrate information [51]. However, considering that our design has shown decent performance in the experiments, we leave the adaptive selection of motif-induced neighborhood as our future work.

After propagating the high-order adjacency information through multiple layers, we obtain multiple representations of the the same dimension d for each user, namely $\{\mathbf{e}_u^{(0)}, \mathbf{e}_u^{(1)} \cdots \mathbf{e}_u^{(l)}\}$. Each representation in this set captures different node interactions and semantics at different layers. To fully exploit the obtained user embeddings, we perform layer combination to get final representations $\mathbf{e}_u$ to predict the alternative neighborhood. Namely, $\mathbf{e}_u = \sum_{l=0}^{L} \frac{1}{L+1}\mathbf{e}_u^{(l)}$. As our goal is to ensure the framework can be trained end-to-end, the process of neighborhood generation must be differentiable rather than using non-differentiable probability-based sampling. In other words, the output of the prediction should be discrete indexes, i.e., integers, that can represent user ID. To this end, a concrete selector layer [22] is employed for discrete user selection. By using the concrete distribution [52] and the reparametrization trick [53], our framework is able to produce a relaxation of the one-hot vector to represent the selected new neighbor. The extent to which the one-hot vector is relaxed is controlled by a temperature parameter $\tau \in (0, \infty)$. In detail, the concrete selector layer contains k neurons, each with a parameter $\mathbf{h}_i$ of size m, where the subscript i denotes the index of the neuron. For each user, we conduct inner product between its final embedding and all users' final embeddings $(\mathbf{E}_\theta\mathbf{e}_{u_i}^T)$ and then feed the intermediate result to the concrete selector layer. The layer performs such an operation

$Softmax((\mathbf{E}_\theta \mathbf{e}_{u_i}^T) \odot \mathbf{h}_i) = \boldsymbol{\alpha}_i$ and then outputs a relaxation of one-hot vector, which is computed as:

$$\mathbf{v}_i = \frac{\exp\left(\left(\log \boldsymbol{\alpha}_i + \mathbf{g}\right)/\tau\right)}{\sum_{j=1}^{m} \exp\left(\left(\log \boldsymbol{\alpha}_{ij} + \mathbf{g}_j\right)/\tau\right)}, \tag{5}$$

where $\mathbf{g}$ is an m-dimensional vector of i.i.d. samples from $Gumbel(0, 1)$[1] [54], which is used to simulate the probability-based relation sampling. When $\tau \to 0$, the concrete random variable $\mathbf{v}_i$ smoothly approaches the discrete distribution, and v_{ij} would be 1 with the probability $\alpha_{ij}/\sum_{p=1}^{m} \alpha_{ip}$. The j^{th} user would then be recognized as one of the new neighbors. As there are k neurons in the concrete selector layer, we can total up all the output and then get a vector $\mathbf{v}$ with k (e.g., 20) positions being 1 that denotes the alternative neighborhood.

3.3 Neighborhood Denoising

Although a majority of users in social recommender systems have only a few friends, there are also a portion of users that have built a lot of connections with others. For these users, identifying alternative neighborhood is like relation denoising. The final identified alternative neighborhood should be those users which contribute most to the recommendation task. It is reasonable to think that, for users with a large number of social relations, if the alternative neighborhood can reconstruct the reliable part of their explicit social profiles, the essential pattern of connectivity may have been encoded into the alternative neighbors. In this paper, we use the motif $\mathcal{M}_8$-induced relations to serve as the reliable part because $\mathcal{M}_8$-induced relations are a subset of the explicit social relations while they are strengthened by common purchases. To impose this constraint, we concatenate the motif-based GCN with a fully-connected multi-layer perception (MLP) and make them nearly work as a concrete autoencoder [22]. AutoEncoders have been widely used in feature selection to filter the irrelevant features [55]. In our problem setting, filtering redundant and noisy relations for users with many social connections is equivalent to feature selection especially when we use the concrete autoencoder to enable discrete relation selection. Structurally, the GCN is the encoder while the MLP is the decoder. From the perspective of data flow, the input is the motif-induced adjacency matrix, the learned representation is the output of the concrete selector layer. Let $\tilde{\mathbf{A}} = f_g(\mathbf{A}_{M_8}, \Theta)$ be the reconstructed motif-induced social profiles, where $\mathbf{A}_{M_8}$ is defined in Eq. 1 and Θ denotes the parameters of both the motif-based GCN and the MLP. We can formally define the objective function of the concrete autoencoder as:

$$\mathcal{L}_s = \arg\min \|\mathbf{A}_{M_8} - \tilde{\mathbf{A}}\|^2. \tag{6}$$

By minimizing the above loss, the learned new neighborhood becomes less noisy even in the user preference domain due to the homophily across social and user-item networks. But it should be noted that, for the users who not only have few social relations but also few purchase records, the effect of the reconstruction constraint is limited as the supervision of a small number of ground truths is relatively trivial. In Section 3.5, we will continue discussing how to directly improve the validity of the alternative neighborhood in the user preference domain.

3.4 Attentive Social Recommendation

With the alternative neighborhood, we next replace the explicit neighborhood for boosted performance. Due to the successful experience of GCN in practice [7], [33], we are motivated to develop an attentive GCN-based recommendation model to perform social recommendation.

3.4.1 *Attentive Social Embedding Propagation Layer*

As social relations often show heterogeneous strengths in different situations, it is crucial to model the importance of social relations when propagating social embeddings. In our GCN-based recommendation model, we perform embedding propagation between both user pairs and user-item pairs. Given a user-item pair (u, i), following [34], we define the information passing from item i to user u at layer l as:

$$\mathbf{e}_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_{(u)}|}\sqrt{|\mathcal{N}_{(i)}|}} \mathbf{e}_i^{(l)} \tag{7}$$

Analogously, the representation $e_i^{(l)}$ for item i is obtained by aggregating information from users who purchased it.

To capture the information from the alternative neighborhood when propagating, we propose a novel attentive social embedding propagation layer. Specifically, given a user pair (u, v) where v is one of the alternative neighbors, the user-to-user information aggregation is constructed as follows:

$$\mathbf{e}_u^{(l+1)} = \sum_{v \in \mathcal{A}_u} \alpha_{u,v}^{(l)} \mathbf{e}_v^{(l)} \tag{8}$$

where $\mathcal{A}_u$ is user u's alternative neighborhood. The difference between Eq (7) and Eq (8) is that we do not use the node centrality, e.g., degree, to measure the neighbor importance or how much information the alternative neighbor v contributes. Instead, we use the learned weight $\alpha_{u,v}^{(l)}$ to selectively acquire information from the neighbor according to the historical records which act as the contexts when aggregating information. To calculate $\alpha_{u,v}^{(l)}$, a social attention mechanism is employed and it formally operates in this way:

$$\alpha_{u,v}^{(l)} = \frac{\exp(\mathbf{q}^{(l)}\sigma(\mathbf{W}_1^{(l)}(\mathbf{e}_u + \mathbf{e}_v)||\mathbf{W}_2^{(l)}\mathbf{e}_i))}{\sum_{v' \in \mathcal{A}_u} \exp(\mathbf{q}^{(l)}\sigma(\mathbf{W}_1^{(l)}(\mathbf{e}_u + \mathbf{e}_{v'})||\mathbf{W}_2^{(l)}\mathbf{e}_i))} \tag{9}$$

where i is the item sampled from the historical purchase data of user u, $\mathbf{q} \in \mathbb{R}^{2d}$, $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_2 \in \mathbb{R}^{d\times d}$ are the parameters of the attention layer where $\mathbf{W}_1$ distills the user embedding and $\mathbf{W}_2$ is for item embedding distillation, and σ is the sigmoid function here. Although we are not the pioneer to apply graph attention mechanism to social recommendation, we are the first to couple attention with contexts to deal with the multi-facets problem of social relations. Compared with existing models, our attention mechanism has the advantage and novelty that the item embedding also participates in the computation of attention scores as the contextual information, which makes our attention mechanism better represent and capture the heterogeneity

[1] $Gumbel(0, 1)$ can be sampled with $g = -log(-log(\mu))$, where $\mu \sim Uniform(0, 1)$.

of social relations, while existing models compute attention scores just with the concatenation of two transformed user embeddings. Additionally, it should be noted that, as this work focuses on social recommendation, we only apply attentive embedding propagation to user-user connections for clear evaluation in the experimental part.

Finally, we combine the social embedding and item embedding propagation to obtain the user representations of the next layer. The operation is defined as:

$$\mathbf{e}_u^{(l+1)} = \sum_{v \in \mathcal{A}_u} \alpha_{u,v}^{(l)} \mathbf{e}_v^{(l)} + \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_{(u)}|}\sqrt{|\mathcal{N}_{(i)}|}} \mathbf{e}_i^{(l)}. \quad (10)$$

3.4.2 Model Prediction and Optimization

By stacking L attentive social embedding propagation layers, users and items are capable of receiving the information propagated from their l-hop neighbors. Following the design of LightGCN [34], we combine different embeddings of different layers to constitute the final embeddings $\mathbf{e}_u^*$ and $\mathbf{e}_i^*$ for model prediction. Formally, $\mathbf{e}^* = \sum_{l=0}^{L} \frac{1}{L+1} \mathbf{e}^{(l)}$. Given an instance (u, i), the predicted score $\hat{y}(u, i)$ is computed by $\mathbf{e}_u^* \mathbf{e}_i^{*T}$.

As this paper focuses on Top-N social recommendation, we adopt the pairwise ranking to model the order of items for each user. Following the Bayesian Personalized Ranking loss [56], we define our optimization function as follows:

$$\mathcal{L}_r = \sum_{(u,i,j) \in \mathcal{O}} -\log \sigma \left(\hat{y}_{u,i}(\Phi) - \hat{y}_{u,j}(\Phi)\right) + \lambda \|\Phi\|_2^2, \quad (11)$$

where Φ denotes the parameters of the attentive recommendation module, and $\sigma(\cdot)$ here is the sigmoid function. Each time a triple including the current user u, the positive item i purchased by u, and the negative item j which is disliked by u or unknown to u sampled from the observed data $\mathcal{O}$, is fed to the model. The model is optimized towards ranking i higher than j in the recommendation list for u. In addition, L_2 regularization is imposed to reduce generalized errors.

3.5 Unifying All Modules with Adversarial Training

So far, we have built an end-to-end framework that can deal with the three tough problems in social recommender systems respectively. However, we notice that there is no distinct reward to stimulate the motif-based GCN to generate better alternative neighborhood that can further enhance recommendation performance. The neighborhood generation model and the recommendation model mainly focus on their own task rather than collaboration.

The theory of homophily refers to the tendency for people to have ties with others who are similar to themselves in socially significant ways. According to this theory, a good neighbor in social recommender systems is very likely to share the same preference with the current user. But for a given item only purchased by the current user, a reasonable assumption is that the neighbor would not show a higher level of interest on this item, which is analogous to the assumption of the social ranking model [29] that a user tends to rank the items purchased by her higher than the ones purchased by her friends. Hence, we came up with such an idea that we should on one hand generate alternative neighbors who are interested in what the current user has purchased, but on the other hand make their affection to the items a little restrained. These two demands of our aim seem to be at cross-purposes. However, our goal is consistent with the key idea of the generative adversarial networks (GANs) [23]. Inspired by the success of GANs in IR applications [39], [19], we are able to make the two demands of our aim compatible.

The key idea of GANs is to let two neural networks contest with each other by playing a Minimax game. In our situation, we need to select new neighbors that can minimize the gap between the scores of the neighbors and the current user on the items purchased by the current user, which is explained to find neighbors who have the similar preferences with the current user. Meanwhile, we also have to maximize this gap because the user herself should show more affection to the purchased item. Let G denote the motif-based GCN, which plays the role of the generator, and D denote the attentive recommendation module, which acts as the discriminator. The Minimax game under our framework can be formulated as:

$$\begin{aligned} \mathcal{L}_{adv} = \arg\min_{D_\Phi} \max_{G_\Theta} -\log \sigma(\hat{y}_{u,i}(\Phi) - \hat{y}_{u',i}(\Phi)), \\ i \in \mathcal{N}_u, u' \in \mathcal{A}_u \sim P_G(\Theta | u, \mathbf{A}). \end{aligned} \quad (12)$$

By fixing the parameters of G and minimizing the above loss, D is optimized towards recognizing the generated neighbor u' and making the gap $(\hat{y}_{u,i} - \hat{y}_{u',i})$ larger. Oppositely, by fixing the parameters of D and maximizing the loss, G evolves towards generating neighbors that can narrow the gap. With the competition between G and D, the optimization for $\mathcal{L}_{adv}$ will eventually reach an equilibrium where the framework shows the best performance.

The use of adversarial training gives our framework a distinct objective that forces G to generate new neighborhood which provides valuable information to D, and also empowers D to capture fine-grained use preferences. Additionally, the neighbors who get lower score $\hat{y}_{u',i}$ would have less probability to be selected again, which can also be regarded as a way to denoise selected relations in the preference domain. With adversarial training, the interactions between the two GCNs in our framework increase, making all the components in our framework couple closely and ensuring a coordinated effort to improve social recommendation. Finally, we unify all the objective functions:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_s + \mathcal{L}_r + \mathcal{L}_{adv} \\ &= \|\mathbf{A}_{M_8} - \tilde{\mathbf{A}}\|^2 - \sum_{(u,i,j) \in \mathcal{O}} \log \sigma \left(\hat{y}_{u,i}(\Phi) - \hat{y}_{u,j}(\Phi)\right) \\ &\quad - \beta \arg\min_{D_\Phi} \max_{G_\Theta} \log \sigma(\hat{y}_{u,i}(\Phi) - \hat{y}_{u',i}(\Phi)) \end{aligned} \quad (13)$$

where β is the coefficient to control the magnitude of the minimax game. For simplicity, we skip regularization terms here. The integrated objective function is optimized by Adam and all the parameters are jointly learned. As the training of GAN is highly unstable, we first pretrain the discriminator in a normal way (with $\mathcal{L}_r$) and pretrain the generator with $\mathcal{L}_s$ until the framework converges, then we conduct adversarial training to enhance the framework. Besides, it should be mentioned that, according to our experimental findings, strictly imposing $\mathcal{L}_s$ on the whole learning process may cause overfitting to explicit relations and also result in mode collapse of the generator. Therefore, we just use it to pretrain the generator and fine tune the framework at the end of each iteration.

3.6 Complexity Analysis

Model size. As we follow the setting of LightGCN, we only need to learn the 0^{th} layer user and item embeddings of size $(m+n)d$ in graph convolution. Beside, for the MLP of the generator G, two matrices of size $m \times t$ are required for neighborhood denoising (m is the number of users, and t is the size of hidden units). The concrete select layer has the parameter of size $m \times k$. For the discriminator D, the weight matrices $\mathbf{W}_1$ and $\mathbf{W}_2$ for the attention mechanism are of the same size $d \times d$, and the parameter of $\mathbf{q}$ is of size $2d$. The total model size approximates $m(2t+k)+2Ld^2+(m+n)d$. Considering that d, t and k are small numbers generally less than 200 and the number of layers L is usually not greater than 5, the model is fairly light.

Time complexity. The computation cost is mainly from layer-wise propagation. For the two GCNs, without the attention mechanism, the layer propagation consumptions are $O(|\mathcal{A}^+|d)$ and $O(|\mathcal{Y}^+|d + kmd)$, respectively, where $|\mathcal{A}^+|$ is the number of non-zero elements in $\mathbf{A}$, $|\mathcal{Y}^+|$ is the number of non-zero elements in $\mathbf{Y}$, and km denotes the number of total alternative neighbors. The time complexity for computation through fully connected layer in G is $O(m^2t)$, but the denoising operation is not compulsory in each epoch. Involving the attention mechanism increases a certain amount of computation, but it also brings decent performance improvement. Hence, we need to make a trade-off between the better performance and time consumption in some cases to determine if the attention mechanism should be employed. As can be seen, in contrast to other GCN-based social recommendation models which directly use explicit social relations, our framework has extra time expenses in searching for alternative neighbors. However, considering the sparsity of the relation and feedback matrices, the compromise is acceptable.

TABLE 2
Dataset Statistics

Dataset	#Users	#Items	#Feedbacks	#Relations
LastFM	1,892	17,632	92,834	25,434
Douban	2,848	39,586	894,887	35,770
Gowalla	18,737	32,510	1,278,274	86,985

4 Experimental Results

4.1 Experimental settings

Datasets. We evaluate our proposed model over three real-world datasets: LastFM[2], Douban[3], and Gowalla[4]. For the dataset of Douban, we only preserve the ratings larger than 3 as the user feedback because our aim is to generate Top-N recommendation. The details of the datasets are summarized in Table 2. Ten times of 5-fold cross validation is adopted in the experiments and we present the average results.

Baselines. We name our framework **ESRF** (**E**hanced **S**ocial **R**ecommendation **F**ramework) in the experiments. To evaluate the performance of ESRF, we compare it with the following methods: BPR [56], SBPR [29], IF-BPR [16], RSGAN [19], LightGCN [34] and DiffNet++ [35]. Among them, BPR and LightGCN are general recommendation models based on implicit feedback while the rest are social recommendation models. As SBPR and DiffNet++ directly leverage explicit social relations for item ranking, comparing ESRF with them can validate the effectiveness of the alternative neighborhood. IF-BPR and RSGAN are implicit relation-aware models which are highly relevant to ESRF, but they only partially overcome the intrinsic problems of social recommender systems. Choosing these two models as baselines can highlight the integrality of ESRF with regard to fully addressing the social recommendation issues. Specifically, LightGCN and DiffNet++ are GCN-based models as well.

Metrics. To evaluate the performance of all methods, two relevancy-based metrics *Precision@10* and *Recall@10* and one ranking-based metric *NDCG@10* are used. For a fair comparison, we rank all the candidate items to calculate these three metrics.

Settings. In our experiments, we use grid search to tune all the specific parameters of baselines and ESRF to ensure the best performance of the baselines for fair comparison. For ESRF, the numbers of the alternative neighbors k on LastFm, Douban, and Gowalla are set to 40, 30, and 40, respectively. The coefficients for controlling the magnitude of adversarial training β are 0.3, 0.6, and 0.5, respectively. The temperature τ that controls the degree of relaxation for neighborhood generation is set to 0.2. For the general settings of all the models, we empirically set the dimension of latent factor vectors to 50, the regularization coefficient λ to 0.005, and the batch size to 512. For all the GCN-based models, we choose the number of the layers according to their best performance on these datasets. The number of layers is set to 4, 3, and 4 on LastFM, Douban, and Gowalla, respectively. Section 4.5 further investigates the sensitivity of ESRF to the number of layers, and presents the results. For all the experiments, we use these settings.

4.2 Recommendation Performance

In this part, we validate if ESRF is as effective as expected and outperforms the recent baselines. We respectively conducted experiments on the whole training set and the selected training set in which only the data of the cold-start users with less than 20 historical purchase records are included. The results are shown in Table 3 and Table 4.

As can be observed, ESRF outperforms the baselines in both the general and cold-start cases on all the datasets. Particularly, the performance improvement in the cold-start case is remarkable, more than 9% on average over all the datasets. As for the baselines, we notice that GCN-based recommendation models: LightGCN and DiffNet++, outperform the corresponding shallow models: BPR and SBPR,

[2] http://files.grouplens.org/datasets/hetrec2011/
[3] http://smiles.xjtu.edu.cn/Download/download_Douban.html
[4] https://www.dropbox.com/sh/qy3s8rs66nirhl9/AAClmTnFO-rR-4ecEYO-jU4ba?dl=0

TABLE 3
General recommendation performance comparison.

Dataset	Metric	Random	BPR	SBPR	IF-BPR	DiffNet++	RSGAN	LightGCN	**ESRF**	Improv.
LastFM	Prec@10	8.113%	7.585%	8.276%	9.102%	10.083%	9.475%	10.139%	**10.723**%	5.848%
	Recall@10	11.992%	11.306%	12.764%	13.625%	14.801%	14.128%	14.960%	**16.076**%	7.459%
	NDCG@10	0.11826	0.10763	0.12188	0.13043	0.14316	0.13509	0.14371	**0.15341**	6.749%
Douban	Prec@10	15.133%	14.892%	15.497%	16.433%	17.375%	17.263%	17.268%	**18.231**%	4.926%
	Recall@10	5.225%	5.023%	5.422%	5.542%	6.136%	6.032%	6.095%	**6.546**%	6.681%
	NDCG@10	0.16406	0.15855	0.17245	0.18551	0.19641	0.19469	0.19488	**0.21033**	7.087%
Gowalla	Prec@10	4.810%	4.521%	3.972%	5.245%	5.098%	5.360%	5.187%	**5.842**%	8.992%
	Recall@10	5.755%	5.484%	4.807%	6.287%	5.993%	6.327%	6.102%	**6.907**%	6.906%
	NDCG@10	0.06623	0.06311	0.05459	0.07279	0.07077	0.07450	0.07197	**0.08026**	7.731%

TABLE 4
Cold-start recommendation performance comparison.

Dataset	Metric	Random	BPR	SBPR	IF-BPR	DiffNet++	RSGAN	LightGCN	**ESRF**	Improv.
LastFM	Prec@10	5.906%	5.903%	6.129%	6.857%	6.787%	7.142%	6.806%	**7.731**%	8.246%
	Recall@10	13.480%	13.924%	15.478%	16.736%	16.720%	16.667%	16.548%	**18.410**%	9.995%
	NDCG@10	0.10674	0.11577	0.12128	0.13855	0.13901	0.14017	0.13127	**0.15230**	8.653%
Douban	Prec@10	1.612%	1.722%	1.935%	2.152%	2.230%	2.178%	2.134%	**2.421**%	8.565%
	Recall@10	6.735%	7.178%	8.084%	8.441%	8.705%	8.583%	8.317%	**9.575**%	9.994%
	NDCG@10	0.04577	0.04784	0.05716	0.06246	0.06767	0.06447	0.06037	**0.07240**	6.989%
Gowalla	Prec@10	2.540%	3.017%	2.776%	3.431%	3.212%	3.560%	3.129%	**3.993**%	12.162%
	Recall@10	5.528%	6.693%	5.855%	7.285%	6.854%	7.427%	6.995%	**8.142**%	9.627%
	NDCG@10	0.04906	0.05782	0.05276	0.06262	0.05812	0.06487	0.05804	**0.07171**	9.538%

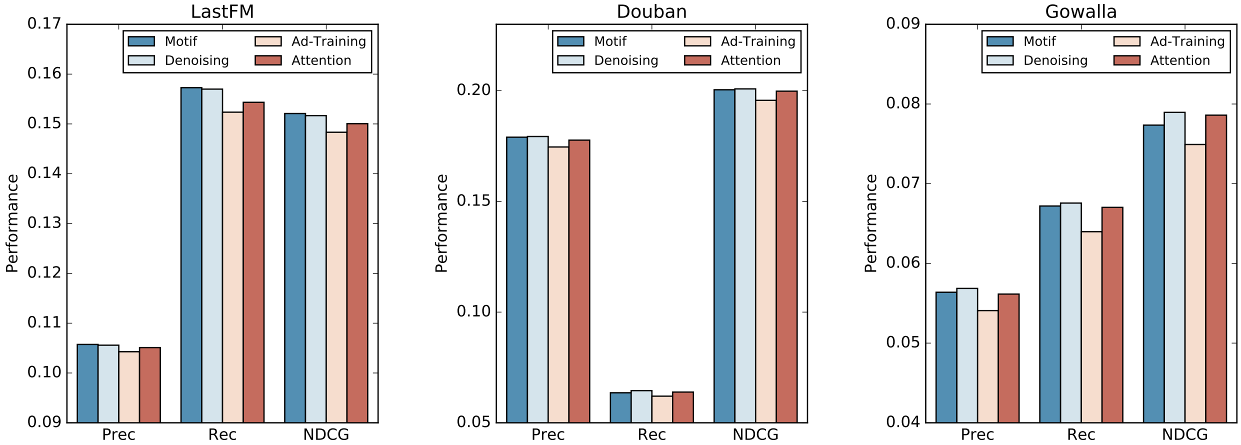
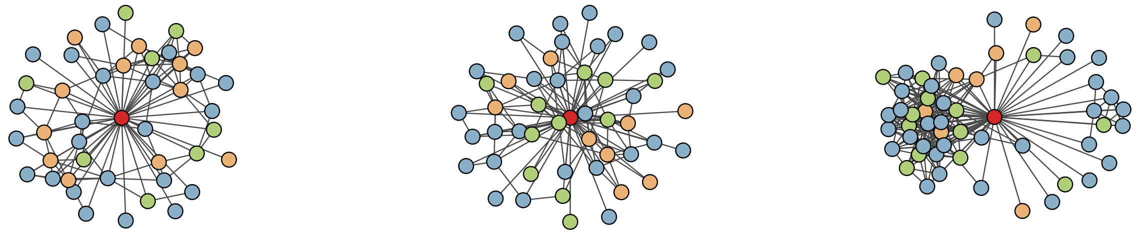


Fig. 3. Ablation study on the three datasets.

which shows the great capacity of GCNs. Social recommendation models using explicit social relations by and large outperform (at least are comparable to) the general recommendation models on two denser datasets: LastFM and Douban. But when it comes to the dataset of Gowalla which is much sparser in both the social and preference domains, these social recommendation models are less competitive. This situation deteriorates in the cold-start case with SBPR even failing to beat BPR. But meanwhile, the implicit relation aware models: IF-BPR and RSGAN show good performance on three datasets, it is in line with the motivation of this paper that explicit social relations sometimes may mislead the social recommendation models and it is necessary to explore and exploit implicit but helpful relations. Besides, RSGAN which is also based on adversarial training shows the second or third best performance in many cases. The success of RSGAN and ESRF confirms the effectiveness of applying adversarial training to social recommendation. Finally, to verify that the the performance improvement of ESRF is brought by the generated alternative neighborhood, we randomly sample the same number of relations for each user and replace the alternative neighborhood with them in the discriminator of ESRF while suspend the generator. The modified ESRF is named Random in Table 3 and Table 4 and we can observe that the performance of Random is much worse than that of other models in all the cases. Then we can draw a conclusion that the alternative neighborhood contributes a lot to the performance of ESRF.



(a).Ego-network sampled from LastFM (b).Ego-network sampled from Douban (c).Ego-network sampled from Gowalla

Fig. 4. Three nodes with their ego social networks sampled from three datasets. The red node denotes the user herself, the blue nodes denote the explicit social relations, and the orange and green nodes denote the alternative neighbors. The overlapped part of explicit social relations and alternative neighbors is denoted by green nodes.

4.3 Ablation Study

In ESRF, we design three components to address three common problems in social recommender systems, and then use adversarial training to unify them. Here we respectively decouple the three components and adversarial training from ESRF to verify the effectiveness of each module. In Fig. 3, we present the results of modified ESRFs in different cases. In each case, one of the four modules of ESRF is removed from ESRF for general recommendation. Each bar represents a case that the corresponding module is removed. The lower the bar is, the more helpful the corresponding module is. It should be noted that we directly perform convolutional operation on explicit social relations to generated the alternative neighborhood when motifs are not used. As can be seen, each module contributes to the overall performance and the importance of modules varies from dataset to dataset. Overall, adversarial training is the most helpful part which accounts for the largest improvement compared with the baselines. This finding corroborates that our intuition of using adversarial generative networks is reasonable. The second important module is the social attention mechanism which plays a crucial role in graph convolution. Relatively speaking, the motifs and the denoising module contribute less but also beneficial. On the sparsest dataset - Gowalla, motifs show more importance than the denoising module, which can be attributed to its function to discover implicit connectivity pattern when the explicit connections are very limited. Besides, since adversarial training and the attention module can partially undertake the job of the denoising module, the missing of the denoising mechanism would not heavily degenerate ESRF. In summary, according to the results, we can draw a conclusion that capturing high-order social information is more essential than denoising when the social network is sparse, while adversarial training and social attention mechanism are always effective because they determine how the alternative neighborhood is generated and used.

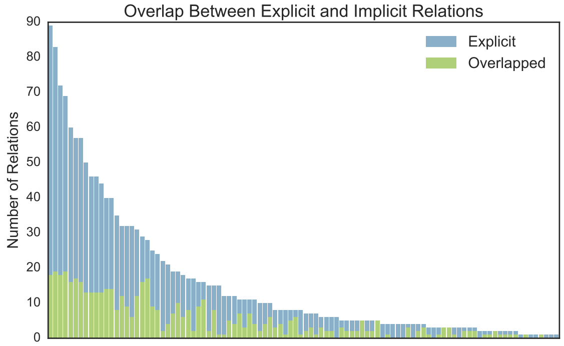


Fig. 5. The degree of the overlap between explicit and implicit relations on LastFM.

4.4 Comparison of Explicit Neighbors and Alternative Neighbors

As the focus of our paper, the alternative neighborhood is different from the explicit relations but has connections with them. To thoroughly investigate its characteristics, we first compare the connectivity patterns of explicit social neighbors and the alternative neighborhood. For each center user, we identify 20 alternative neighbors for her. Three ego social networks with both explicit and alternative neighbors are randomly sampled from three datasets, shown in Fig. 4. We can observe that: (1). More than half of alternative neighbors are also explicit neighbors when the center user has a large number of explicit social relations. (2). Compared with explicit neighbors, alternative neighbors are less isolated. Most of them are socially connected with other alternative neighbors or explicit neighbors. The second finding is very crucial. Intuitively, the strength of a tie becomes strong if the tie builders have common friends. According to the theory of homophily [1], they will also become similar in preferences. By contrast, the isolated explicit neighbors may have tenuous relationships with the center user and are less likely to share common interests with her. This can be a reason that explains why ESRF beats the social recommendation models using explicit social relations. Besides, we further investigate the degree of the overlap between the explicit relations and the alternative neighborhood. We randomly sample 100 users from LastFM and draw a distribution of the explicit relations. As can be seen in Fig. 5, more than 70% users have less than 20 explicit relations and the distribution is a power-law-like distribution. For those users who have many explicit relations, we can observe an obvious overlap between the explicit relations and most of the identified alternative neighbors, which illustrates the

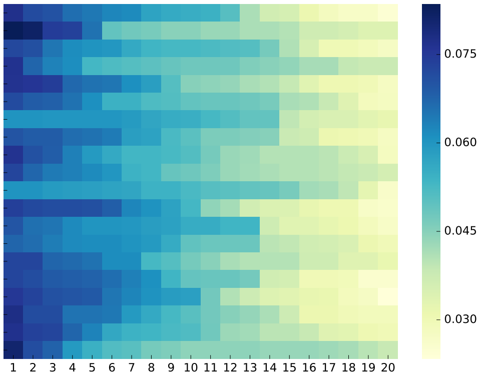


Fig. 6. The heatmap of learned attention weights on LastFM.

alternative neighborhood's connections to explicit relations. Statistically, in this case, 32.44% alternative relations overlap with the explicit relations. In addition, as we aim to capture the heterogeneous strengths of social relations, we propose a social attention mechanism to prevent relations being indiscriminately used. To validate its effectiveness, we collect the learned attention scores during training and randomly sample 20 users from LastFM and report the average results. In Fig. 6, each row in this heatmap represents a user and each column represents one of its identified 20 alternative neighbors. It is clear that, for each user, different alternative neighbors show different strengths in the convolution operation.

4.5 Parameter Sensitivity Analysis

We introduce two hyper-parameters k and β to ESRF. k denotes the number of identified alternative neighbors for each user and β controls the magnitude of adversarial training. In this section, we investigate the sensitivity of k and β. As can be seen in Fig. 7, with the increase of the number of alternative neighbors, the performance of ESRF on all the datasets becomes better. After reaching its peak when k is 40, 30, 50 on LastFM, Douban, and Gowalla, respectively, it steadily declines. Overall, the performance curve on Gowalla changes more dramatically, which could be attributed to Gowalla's high sparsity. These trends can be explained as that, augmenting the social information helps when data are sparse, but excessive augmentation may also introduce noises, which would mislead the model. From Fig. 8, we can observe similar trends that adversarial training improves ESRF, but a large magnitude of adversarial training will lower the performance of ESRF. Intuitively, we think it happens because maximizing the adversarial loss is opposite to our optimization goal. Excessive maximization would emphasize the goal of generating useful neighborhood while deprioritizing the recommendation task. Besides, as GCNs are sensitive to the number of layers, we also investigate the influence of the number of layers in this section. As the performance is mainly determined by the recommendation module and the effectiveness of motifs have been confirmed in Section 4.3, we only present the results on the discriminator in Fig. 9. As can be observed, with two layers, ESRF can almost outperform all the baselines (compare to Table 3/4). Increasing the number of layers can improve the performance, but the benefits diminish. The best performance is achieved when the layer number is 4, 3, and 4 in these three datasets, respectively.

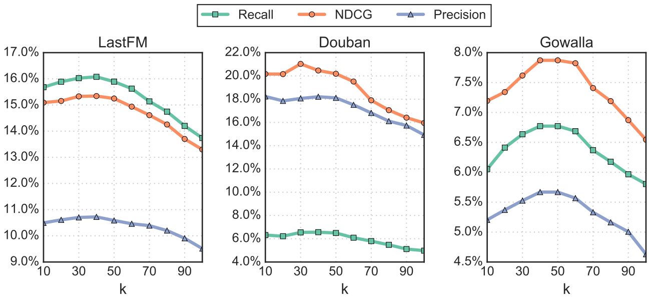


Fig. 7. The influence of the number of alternative neighbors.

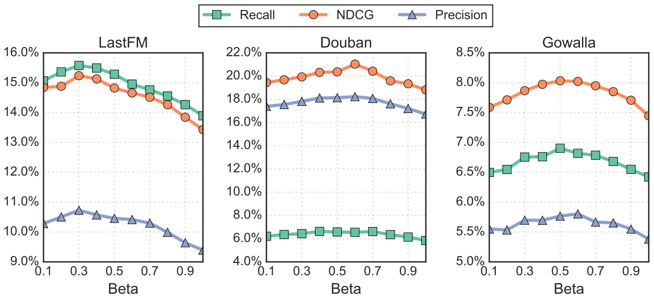


Fig. 8. The influence of the magnitude of adversarial training.

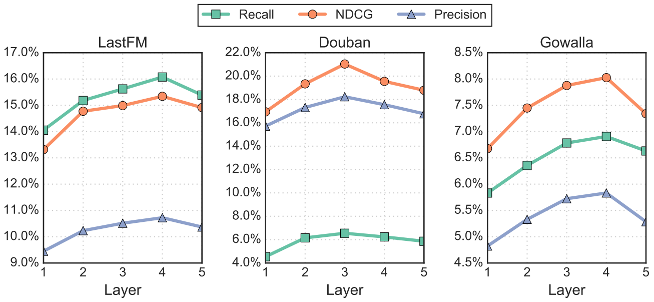


Fig. 9. The influence of the number of layers of the discriminator.

5 Conclusion

Social recommender systems have received attention these years due to the potential of social relations to improve traditional recommendation. However, the practice of social recommendation is not as successful as expected. Existing social recommender systems only pay attention to the homophily in social networks and neglect the intrinsic problems of social relations. In this paper, we develop a deep adversarial framework based on GCNs to address the challenges of social recommendation. To the best of our knowledge, this is the first work that combines adversarial training and GCNs to overcome the drawbacks of social recommender systems. Experimental results on three real-world datasets show that our recommendation model significantly outperforms other state-of-the-art social recommendation models.

6 Acknowledgement

This work was supported by ARC Discovery Project (Grant No.DP190101985, DP170103954).

References

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

[2] M. Jamali and M. Ester, "Trustwalker:a random walk model for combining trust-based and item-based recommendation," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 397–406.

[3] H. Ma, M. R. Lyu, and I. King, "Learning to recommend with trust and distrust relationships," in *ACM Conference on Recommender Systems*, 2009, pp. 189–196.

[4] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011, pp. 287–296.

[5] 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 *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 123–129.

[6] H. Yin, X. Zhou, B. Cui, H. Wang, K. Zheng, and Q. V. H. Nguyen, "Adapting to user interest drift for poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2566–2581, 2016.

[7] W. Fan, Y. Ma, Q. Li, Y. He, Y. E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, 2019, pp. 417–426.

[8] 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.

[9] 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.

[10] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1113–1133, 2013.

[11] 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*. SIAM, 2017, pp. 381–389.

[12] J. Yu, M. Gao, W. Rong, W. Li, Q. Xiong, and J. Wen, "Hybrid attacks on model-based social recommender systems," *Physica A: Statistical Mechanics and its Applications*, vol. 483, pp. 171–181, 2017.

[13] J. Tang, H. Gao, and H. Liu, "mtrust: discerning multi-faceted trust in a connected world," in *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 2012, pp. 93–102.

[14] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, "Social recommendation with strong and weak ties," in *ACM International on Conference on Information and Knowledge Management*, 2016, pp. 5–14.

[15] H. Ma, "An experimental study on implicit social recommendation," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2013, pp. 73–82.

[16] 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*. ACM, 2018, pp. 357–366.

[17] 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.

[18] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[19] J. Yu, M. Gao, H. Yin, J. Li, C. Gao, and Q. Wang, "Generating reliable friends via adversarial training to improve social recommendation," *arXiv preprint arXiv:1909.03529*, 2019.

[20] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[21] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[22] A. Abid, M. F. Balin, and J. Zou, "Concrete autoencoders for differentiable feature selection and reconstruction," *arXiv preprint arXiv:1901.09346*, 2019.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[24] 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*. ACM, 2008, pp. 931–940.

[25] 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.

[26] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 203–210.

[27] J. Chen, Y. Feng, M. Ester, S. Zhou, C. Chen, and C. Wang, "Modeling users' exposure with social knowledge influence and consumption influence for recommendation," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 953–962.

[28] H. Yin, W. Wang, H. Wang, L. Chen, and X. Zhou, "Spatial-aware hierarchical collaborative deep learning for poi recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2537–2551, 2017.

[29] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 261–270.

[30] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 566–577.

[31] H. Yin, Q. Wang, K. Zheng, Z. Li, and X. Zhou, "Overcoming data sparsity in group recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[32] 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.

[33] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[34] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," *arXiv preprint arXiv:2002.02126*, 2020.

[35] L. Wu, J. Li, P. Sun, Y. Ge, and M. Wang, "Diffnet++: A neural influence and interest diffusion network for social recommendation," *arXiv preprint arXiv:2002.00844*, 2020.

[36] 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.

[37] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 2019, pp. 555–563.

[38] K.-M. Kim, D. Kwak, H. Kwak, Y.-J. Park, S. Sim, J.-H. Cho, M. Kim, J. Kwon, N. Sung, and J.-W. Ha, "Tripartite heterogeneous graph propagation for large-scale social recommendation," *arXiv preprint arXiv:1908.02569*, 2019.

[39] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "Irgan: A minimax game for unifying generative and discriminative information retrieval models," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2017, pp. 515–524.

[40] H. Wang, J. Wang, J. Wang, M. Zhao, W. Zhang, F. Zhang, X. Xie, and M. Guo, "Graphgan: Graph representation learning with

generative adversarial nets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[41] Q. Wang, H. Yin, Z. Hu, D. Lian, H. Wang, and Z. Huang, "Neural memory streaming recommender networks with adversarial training," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2467–2475.

[42] D.-K. Chae, J.-S. Kang, S.-W. Kim, and J.-T. Lee, "Cfgan: A generic collaborative filtering framework based on generative adversarial networks," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2018, pp. 137–146.

[43] F. Zhou, R. Yin, K. Zhang, G. Trajcevski, T. Zhong, and J. Wu, "Adversarial point-of-interest recommendation," in *The World Wide Web Conference*, 2019, pp. 3462–34 618.

[44] X. He, Z. He, X. Du, and T.-S. Chua, "Adversarial personalized ranking for recommendation," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 355–364.

[45] Z. Wang, M. Gao, X. Wang, J. Yu, J. Wen, and Q. Xiong, "A minimax game for generative and discriminative sample models for recommendation," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2019, pp. 420–431.

[46] Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, "Enhancing collaborative filtering with generative augmentation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 548–556.

[47] 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.

[48] W. Fan, T. Derr, Y. Ma, J. Wang, J. Tang, and Q. Li, "Deep adversarial social recommendation," *arXiv preprint arXiv:1905.13160*, 2019.

[49] H. Zhao, X. Xu, Y. Song, D. L. Lee, Z. Chen, and H. Gao, "Ranking users in social networks with higher-order structures," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

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

[51] J. B. Lee, R. A. Rossi, X. Kong, S. Kim, E. Koh, and A. Rao, "Graph convolutional networks with motif-based attention," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 499–508.

[52] C. J. Maddison, A. Mnih, and Y. W. Teh, "The concrete distribution: A continuous relaxation of discrete random variables," *arXiv preprint arXiv:1611.00712*, 2016.

[53] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

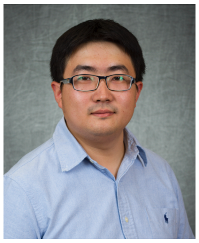
[54] E. J. Gumbel, *Statistical theory of extreme values and some practical applications: a series of lectures*. US Government Printing Office, 1948, vol. 33.

[55] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[56] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 452–461.

Junliang Yu received the B.S. and M.S degrees in Software Engineering from Chongqing University, Chongqing, China. Currently, he is a Ph.D. student with the school of Information Technology and Electrical Engineering at the University of Queensland, Queensland, Australia. His research interests include recommender systems and anomaly detection.

Hongzhi Yin received the PhD degree in computer science from Peking University, in 2014. He is a senior lecturer with the University of Queensland. He received the Australia Research Council Discovery Early-Career Researcher Award, in 2016 and UQ Foundation Research Excellence Award, in 2019. His research interests include recommendation system, user profiling, topic models, deep learning, social media mining, and location-based services.



Jundong Li is an Assistant Professor at the University of Virginia. Prior to joining UVA, he received his Ph.D. degree in Computer Science at Arizona State University in 2019, M.Sc. degree in Computer Science at University of Alberta in 2014, and B.Eng. degree in Software Engineering at Zhejiang University in 2012. His research interests are broadly in data mining and machine learning, with a particular focus on feature learning, graph mining, and social computing.

Min Gao received the MS and PhD degrees in computer science from Chongqing University in 2005 and 2010 respectively. She is an associate professor at the School of Big Data & Software Engineering, Chongqing University. Her research interests include recommendation systems, service computing, and data mining.

Zi Huang received the BSc degree from the Department of Computer Science, Tsinghua University, China, and the PhD degree in computer science from the School of ITEE, University of Queensland. She is an associate professor (reader) and ARC Future fellow in the School of ITEE, University of Queensland. Her research interests mainly include multimedia indexing and search, social data analysis, and knowledge discovery.

Lizhen Cui is a full professor with Shandong University. He is appointed dean and deputy party secretary for School of Software, co-director of Joint SDU-NTU Centre for Artificial Intelligence Research(C-FAIR), director of the Research Center of Software and Data Engineering, Shandong University. His main interests include big data intelligence theory, data mining, wisdom science, and medical health big data AI applications.