Social Influence Attentive Neural Network for Friend-Enhanced Recommendation

Yuanfu Lu[1,2], Ruobing Xie[2], Chuan Shi[1] ✉, Yuan Fang[3], Wei Wang[2], Xu Zhang[2], and Leyu Lin[2]

[1] Beijing University of Posts and Telecommunications
[2] WeChat Search Application Department, Tencent Inc.
[3] Singapore Management University
luyuanfu@bupt.edu.cn, xrbsnowing@163.com, shichuan@bupt.edu.cn, yfang@smu.edu.sg, {unoywang,xuonezhang,goshawklin}@tencent.com

Abstract. With the thriving of online social networks, there emerges a new recommendation scenario in many social apps, called Friend-Enhanced Recommendation (FER) in this paper. In FER, a user is recommended with items liked/shared by his/her friends (called a friend referral circle). These friend referrals are explicitly shown to users. Different from conventional social recommendation, the unique friend referral circle in FER may significantly change the recommendation paradigm, making users to pay more attention to enhanced social factors. In this paper, we first formulate the FER problem, and propose a novel Social Influence Attentive Neural network (SIAN) solution. In order to fuse rich heterogeneous information, the attentive feature aggregator in SIAN is designed to learn user and item representations at both node- and type-levels. More importantly, a social influence coupler is put forward to capture the influence of the friend referral circle in an attentive manner. Experimental results demonstrate that SIAN outperforms several state-of-the-art baselines on three real-world datasets. (Code and dataset are available at https://github.com/rootlu/SIAN.)

Keywords: Heterogeneous Graph · Friend-Enhanced Recommendation · Social Influence.

1 Introduction

Nowadays, with the thriving of online social networks, people are more willing to actively express their opinions and share information with friends on social platforms. Friends become essential information sources and high-quality information filters. Items that friends have interacted with (shared, liked, etc.) have great impacts on users, which are likely to become users future interests. There are lots of recommender systems that concentrate on social influences of friends (e.g., following feed in YouTube and Top Stories in WeChat). Some social recommendation algorithms also consider social factors for personalization [4, 16].

Impressed by the great successes of social influence in recommendation, we propose a novel scenario named **Friend-Enhanced Recommendation**

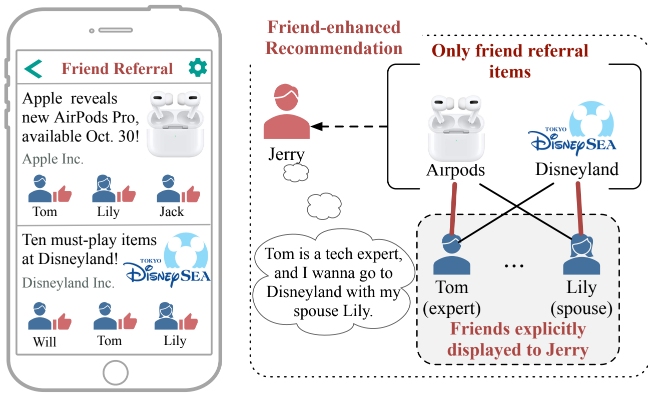


Fig. 1. A typical illustration of the friend-enhanced recommendation. The left shows the scenario that *Jerry* is recommended two articles, with friends (e.g., *Tom*) who have interacted with (shared, liked, etc.) them explicitly shown underneath. The right shows the formalization of the FER problem, where only friend referral items will be recommended and friends who interacted with the item are explicitly displayed to user.

(FER), which multiplies the influence of friends in social recommendation. FER has two major differences from the classical social recommendation: (1) FER only recommends to the user what his/her **friends have interacted with**, regarding friends as high-quality information filters to provide more high-quality items. (2) All friends who have interacted with the item are **explicitly displayed** to the user attached to the recommended item, which highlights the critical importance of explicit social factors and improves the interpretability for user behaviors.

In recent years, FER systems are blooming and have been widely-used by hundreds of millions of users. Fig. 1 gives a typical illustration of a real-world FER. For each user-item pair, FER explicitly shows the friend set having interacted with the item, which is defined as the **Friend Referral Circle (FRC)** of the user to the item. For instance, the FRC of *Jerry* to the article about *AirPods* is {*Tom*, *Lily*, *Jack*}. Such a FRC drastically highlights the social influence of friends and their roles, which makes FER more complicated and relevant. It has even changed the recommendation paradigm compared to classical social recommendation. Taking Fig. 1 as an example, in classical social recommendation, *Jerry* would have no idea about the FRC (which is not displayed to him), hence he may read an article based on his own interest. However, in our FER, in addition to the attractiveness of the item itself, the influence of friends may be the main reason for the click. Here the FRC is explicitly displayed to *Jerry*, so the more likely reason why he clicks the article about *AirPods* is because *Tom* (a tech-expert friend) has read it. It is also entirely possible that *Jerry* reads the article about *Disneyland* because his spouse *Lily* has read it. Furthermore, when the article is related to technology, the coupling between the expert and technology may have a greater impact on Jerry than that between his spouse and technology, whereas the opposite scenario may happen w.r.t. entertainment.

Hence, in FER, multiple factors contribute to user clicks. The reasons for a user clicking an article may come from (1) his interests in item contents (item), (2) the recommendation of an expert (item-friend combination), or even (3) the concerns on his friends themselves (friend). In FER, users have the tendency to see *what their friends have read*, rather than to merely see *what themselves are interested in*. It could even say that social recommendation focuses on bringing social information to better recommend items, while FER aims to recommend the combination of both items and friend referrals.

As the critical characteristic of FER, the explicit FRC brings in two challenges: (1) *How to extract key information from multifaceted heterogeneous factors?* FER involves multiple heterogeneous factors such as item contents, friend referrals and their interactions. The impacts of these factors vary in different scenarios with different combinations of users, items and friend referrals. FER is much more challenging since it is required not only to learn user preferences on items, but also to predict users' concerns towards different factors. (2) *How to exploit explicit friend referral information?* The explicit friend referrals greatly emphasize the importance of social information in recommendation, which are crucial in FER. However, there is few work that has explored the performances and characteristics of FRCs in real-world recommendation. A deliberate strategy is desired to make full use of the explicit friend referral information in FER.

To solve these issues, we propose a novel **S**ocial **I**nfluence **A**ttentive **N**eural network (**SIAN**). Specifically, we define the FER as a user-item interaction prediction task on a heterogeneous social graph, which flexibly integrates rich information in heterogeneous objects and their interactions. First, we design an attentive feature aggregator with both node- and type-level aggregations to learn user and item representations, without being restricted to pre-defined meta-paths in some previous efforts [19, 3]. Next, we implement a social influence coupler to model the coupled influence diffusing through the explicit friend referral circles, which combines the influences of multiple factors (e.g., friends and items) with an attentive mechanism. Overall, SIAN captures valuable multifaceted factors in FER, which successfully distills the most essential preferences of users from a heterogeneous graph and friend referral circles. In experiments, SIAN significantly outperforms all competitive baselines in multiple metrics on three large, real-world datasets. Further quantitative analyses on attentive aggregation and social influence also reveal impressive sociological discoveries. We summarize the contributions as follows:

- We are the first to study the widely-adopted recommendation scenario named friend-enhanced recommendation (FER), where friend referrals are attached to items and explicitly exposed to users.
- We propose a novel Social Influence Attentive Neural network (SIAN) for FER. It uses a novel attentive feature aggregator to extract useful multifaceted information, and leverages a social influence coupler to judge the significance of different friend referrals.
- Experiments on three real-world datasets verify the effectiveness and robustness of SIAN. Further quantitative analyses also reveal valuable sociological

patterns, reflecting the changes and interpretability of user behaviors when social influence becomes more significant.

2 Preliminaries

Definition 1. ***Heterogeneous Social Graph (HSG).*** *A heterogeneous social graph is denoted as* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, where* $\mathcal{V} = \mathcal{V}_U \cup \mathcal{V}_I$ *and* $\mathcal{E} = \mathcal{E}_F \cup \mathcal{E}_R$ *are the sets of nodes and edges. Here* $\mathcal{V}_U$ *and* $\mathcal{V}_I$ *are the sets of users and items. For* $u, v \in \mathcal{V}_U$*,* $\langle u, v\rangle \in \mathcal{E}_F$ *represents the friendship between users. For* $u \in \mathcal{V}_U$ *and* $i \in \mathcal{V}_I$*,* $\langle u, i\rangle \in \mathcal{E}_R$ *is the interaction relation between* u *and* i*.*

It it not difficult to extend the HSG by adding attribute features or link relations as a Heterogeneous Information Network (HIN) [14]. Fig. 1 shows an HSG containing three types of nodes, i.e., {*User*, *Article*, *Media*}, and multiple relations, e.g., {*User-User*, *User-Article*, *User-Media*, *Article-Media*}.

Definition 2. ***Friend Referral Circle (FRC).*** *Given an HSG* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$*, we define the friend referral circle of a user* u *w.r.t. a non-interacting item* i *(i.e.,* $\langle u, i\rangle \notin \mathcal{E}_R$*) as* $\mathcal{C}_u(i) = \{v|\langle u, v\rangle \in \mathcal{E}_F \cap \langle v, i\rangle \in \mathcal{E}_R\}$*. Here* v *is called an* ***influential friend*** *of user* u*.*

Taking Fig. 1 as an example, the friend referral circle of *Jerry* w.r.t. the non-interacting article about *AirPods* is {*Tom*, *Lily*, *Jack*}, while the FRC in terms of the article about *Disneyland* is $\mathcal{C}_{Jerry}(Disneyland) = \{Will, Tom, Lily\}$.

Definition 3. ***Friend-Enhanced Recommendation (FER).*** *Given an HSG* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *and the FRC* $\mathcal{C}_u(i)$ *of a user* u *w.r.t. a non-interacting item* i*, the FER aims to predict whether user* u *has a potential preference to item* i*. That is, a prediction function* $\hat{y}_{ui} = \mathcal{F}(\mathcal{G}, \mathcal{C}_u(i); \Theta)$ *is to be learned, where* $\hat{y}_{ui}$ *is the probability that user* u *will interact with item* i*, and* Θ *is the model parameters.*

3 The Proposed Model

3.1 Model Overview

As illustrated in Fig. 2, SIAN models the FER with an HSG. In addition to the user and item representations (e.g., $\mathbf{h}_u$ for *Jerry* and $\mathbf{h}_i$ for the *Disneyland* article), SIAN learns a social influence representation (e.g., $\mathbf{h}_{ui}$) by coupling each influential friend (e.g., *Tom*) with the item. They are jointly responsible for predicting the probability $\hat{y}_{ui}$ of interaction between user u and item i.

First, each user or item node is equipped with an attentive feature aggregator with node- and type-level aggregations, which is designed to exploit multifaceted information. At the node level, the features from the neighbours of the same type (e.g., articles that *Jerry* liked) will be aggregated in the current type space; at the type level, the representations from different type spaces will be further aggregated to encode multifaceted information. At each level, an attention

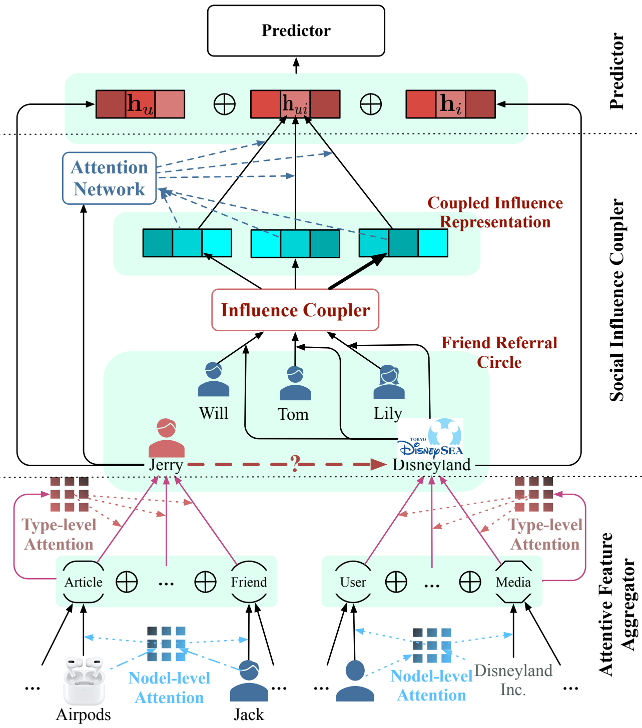


Fig. 2. The overall architecture of SIAN. The attentive feature aggregator hierarchically aggregates heterogeneous neighbour features with node- and type-level attention, and outputs the representations of users and items (i.e., $\mathbf{h}_u$ and $\mathbf{h}_i$). The social influence coupler couples the influence of each influential friends and the item, to encode the explicit social influence into the representation (i.e., $\mathbf{h}_{ui}$).

mechanism is employed to differentiate and capture the latent relevance of the neighbors and types, respectively. Such a hierarchical attentive design enables SIAN to encode the fine-grained relevance of multifaceted information, and the dual attention mechanism allows it to delicately capture the effect of different factors. Unlike some previous works [19, 3], SIAN does not require any manual selection of meta-paths, so that it is expected to yield a better performance.

Second, the influence from an influential friend (e.g., *Tom*) and an item (e.g., the *Disneyland* article) is jointly captured with a social influence coupler, which quantifies the degree of their coupled influence. Multiple coupled influences from the FRC are then combined through attentive propagation to derive the representation of the overall influence (i.e., $\mathbf{h}_c$). With the learned user, item and influence representations, SIAN predicts the probability $\hat{y}_{ui}$ that user u (e.g., *Jerry*) will interact with item i (e.g., the *Disneyland* article).

3.2 Attentive Feature Aggregator

Given an HSG $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, attentive feature aggregator aims to learn user and item representations (i.e., $\mathbf{h}_u$ and $\mathbf{h}_i$, $u, i \in \mathcal{V}$). Considering that different neighbours of the same type might not equally contribute to the feature aggregation, and different types entail multifaceted information, we design a hierarchical node- and type-level attentive aggregation. Node-level aggregation separately models user/item features in a fine-grained manner, while type-level aggregations capture heterogeneous information.

Node-level Attentive Aggregation. Formally, given a user u, let $\mathcal{N}_u = \mathcal{N}_u^{t_1} \cup \mathcal{N}_u^{t_2} \cup \cdots \cup \mathcal{N}_u^{t_{|\mathcal{T}|}}$ denotes his/her neighbours, which is a union of $|\mathcal{T}|$ types of neighbour sets. For neighbours of type $t \in \mathcal{T}$ (i.e., $\mathcal{N}_u^t$), we represent the aggregation in the t type space as the following function:

$$\mathbf{p}_u^t = \mathrm{ReLU}(\mathbf{W}_p(\sum\nolimits_{k \in \mathcal{N}_u^t} \alpha_{ku} \mathbf{x}_k) + \mathbf{b}_p), \tag{1}$$

where $\mathbf{p}_u^t \in \mathbb{R}^d$ is the aggregated embeddings of user u in t type space. $\mathbf{x}_k \in \mathbb{R}^d$ is the initial embedding of the neighbour k, which is randomly initialized. Here $\mathbf{W}_p \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_p \in \mathbb{R}^d$ are the weight and bias of a neural network. α_{ku} is the attentive contribution of neighbour k to the feature aggregation of u,

$$\alpha_{ku} = \frac{\exp(f([\mathbf{x}_k \oplus \mathbf{x}_u]))}{\sum_{k' \in \mathcal{N}_u^t} \exp(f([\mathbf{x}_{k'} \oplus \mathbf{x}_u]))}, \tag{2}$$

where $f(\cdot)$ is a two-layer neural network activated with ReLu function and $\oplus$ denotes the concatenation operation. Obviously, the larger α_{ku}, the greater contribution of neighbour k to the feature aggregation of user u.

Given multiple types of neighbours, we can get multiple embeddings for u in various type spaces, denoted as $\{\mathbf{p}_u^{t_1}, \cdots, \mathbf{p}_u^{t_{|\mathcal{T}|}}\}$.

Type-level Attentive Aggregation. Intuitively, different types of neighbours indicate various aspects of information and a node is likely to have different preferences for multiple aspects. Given a user u and his/her node-level aggregated embeddings in different type spaces, we aggregate them as follows:

$$\mathbf{h}_u = \mathrm{ReLU}(\mathbf{W}_h \sum\nolimits_{t \in \mathcal{T}} \beta_{tu} \mathbf{p}_u^t + \mathbf{b}_h), \tag{3}$$

where $\mathbf{h}_u \in \mathbb{R}^d$ is the latent representation of user u. $\{\mathbf{W}_h \in \mathbb{R}^{d \times d}, \mathbf{b}_h \in \mathbb{R}^d\}$ are parameters of a neural network. β_{tu} is the attentive preferences of type t w.r.t. the feature aggregation of user u, as various types of neighbours contain multifaceted information and are expected to collaborate with each other. For user u, we concatenate the aggregated representations of all neighbour types, and define the following weight:

$$\beta_{tu} = \frac{\exp(\mathbf{a}_t^\top [\mathbf{p}_u^{t_1} \oplus \mathbf{p}_u^{t_2} \oplus \cdots \oplus \mathbf{p}_u^{t_{|\mathcal{T}|}}])}{\sum_{t' \in \mathcal{T}} \exp(\mathbf{a}_{t'}^\top [\mathbf{p}_u^{t_1} \oplus \mathbf{p}_u^{t_2} \oplus \cdots \oplus \mathbf{p}_u^{t_{|\mathcal{T}|}}])}, \tag{4}$$

where $\mathbf{a}_t \in \mathbb{R}^{|\mathcal{T}|d}$ is a type-aware attention vector shared by all users. With Eq. (4), the concatenation of various neighbour types captures multifaceted information for a user, and $\mathbf{a}_t$ encodes the global preference of each type.

Similarly, for each item i, the attentive feature aggregator takes the neighbours of i as input, and outputs the latent representation of i, denoted as $\mathbf{h}_i$.

3.3 Social Influence Coupler

To exploit the FRCs and capture the effects of influential friends, we propose a social influence coupler. The differential impact of the influential friends and the item on social behaviors is first coupled together, and then we attentively represent the overall influence in the FRC.

Coupled Influence Representation. Following [7], human behaviors are affected by various factors. In FER, whether u interacts with i is not simply driven by only the item itself or only the friends. More likely, the co-occurrence of friends and the item have a significant impact. As in the previous example (Fig. 1), when it is technology-related, the coupling between the expert (e.g. *Tom*) and the item (e.g. *AirPods*) has a greater impact than the coupling between the spouse and a tech-item, but the opposite scenario may happen for entertainment-related items. Hence, given user u, item i, and the FRC $\mathcal{C}_u(i)$, we couple the influence of each friend $v \in \mathcal{C}_u(i)$ and item i as following:

$$\mathbf{c}_{\langle v,i \rangle} = \sigma(\mathbf{W}_c \phi(\mathbf{h}_v, \mathbf{h}_i) + \mathbf{b}_c), \tag{5}$$

where $\mathbf{h}_v$ and $\mathbf{h}_i$ are aggregated representations of user v and item i. $\phi(\cdot,\cdot)$ serves as a fusion function, which can be element-wise product or concatenation (here we adopt concatenation). σ is the ReLU function. Obviously, Eq. (5) couples the features of item i and the influential friend v, capturing the influence of both.

Attentive Influence Degree. With the coupled influence representation $\mathbf{c}_{\langle v,i \rangle}$, our next goal is to obtain the influence degree of $\mathbf{c}_{\langle v,i \rangle}$ on the user u. Since the influence score depends on user u, we incorporate the representation of user u (i.e., $\mathbf{h}_u$) into the influence score calculation with a two-layer neural network parameterized by $\{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, b_2\}$:

$$d'_{u \leftarrow \langle v,i \rangle} = \sigma(\mathbf{W}_2(\sigma(\mathbf{W}_1 \phi(\mathbf{c}_{v,i}, \mathbf{h}_u) + \mathbf{b}_1)) + b_2). \tag{6}$$

Then, the attentive influence degree is obtained by normalizing $d'_{u \leftarrow \langle v,i \rangle}$, which can be interpreted as the impact of the influential friend v on the user behavior:

$$d_{u \leftarrow \langle v,i \rangle} = \frac{\exp(d'_{u \leftarrow \langle v,i \rangle})}{\sum_{v' \in \mathcal{C}_u(i)} \exp(d'_{u \leftarrow \langle v',i \rangle})}. \tag{7}$$

Since the influences of friends propagate from the FRC, we attentively sum the coupled influences of the influential friends and item v on user u:

$$\mathbf{h}_{ui} = \sum\nolimits_{v \in \mathcal{C}_u(i)} d_{u \leftarrow \langle v,i \rangle} \mathbf{c}_{\langle v,i \rangle}. \tag{8}$$

As the coupled influence representation $\mathbf{c}_{\langle v,i\rangle}$ incorporates the latent factors of the influential friend and the item, Eq. (8) guarantees that the social influence propagating among them can be encoded into the latent representation $\mathbf{h}_{ui}$.

3.4 Behavior Prediction and Model Learning

With the representations of user, item and the coupled influence (i.e., $\mathbf{h}_u$, $\mathbf{h}_i$ and $\mathbf{h}_{ui}$), we concatenate them and then feed it into a two-layer neural network:

$$\mathbf{h}_o = \sigma(\mathbf{W}_{o_2}(\sigma(\mathbf{W}_{o_1}([\mathbf{h}_u \oplus \mathbf{h}_{ui} \oplus \mathbf{h}_i]) + \mathbf{b}_{o_1}) + \mathbf{b}_{o_2}). \tag{9}$$

Then, the predicted probability of a user-item pair is obtained via a regression layer with a weight vector $\mathbf{w}_y$ and bias b_y:

$$\hat{y}_{ui} = \text{sigmoid}(\mathbf{w}_y^\top \mathbf{h}_o + b_y). \tag{10}$$

Finally, to estimate model parameters Θ of SIAN, we optimize the following cross-entropy loss, where y_{ui} is the ground truth and λ is the L2-regularization parameter for reducing overfitting:

$$-\sum_{\langle u,i\rangle \in \mathcal{E}_R} (y_{ui} \log \hat{y}_{ui} + (1 - y_{ui}) \log (1 - \hat{y}_{ui})) + \lambda ||\Theta||_2^2. \tag{11}$$

4 Experiments

We conduct comprehensive experiments on three real-world datasets, demonstrating superior performance and revealing interesting sociological patterns.

4.1 Datasets

Yelp and Douban are classical open datasets widely used in recommendation, for which we build FRCs for each user-item pair to simulate the FER scenarios. FWD is extracted from a deployed live FER system with real FRCs displayed to users. The detailed statistics of datasets are shown in Table 1.

- **Yelp**[4] is a business review dataset containing both interactions and social relations. We first sample a set of users. For each user u, we construct a set of FRCs based on the given user-user relations and user-item interactions. Interactions with an empty FRC are filtered from the data. To get the initial feature vector of a node, we learn the word embeddings with word2vec using the review texts, and average the learned vectors for each user or item.
- **Douban**[5] is a social network related to sharing books, which including friendships between users and interaction records between users and items. As preprocesses done for Yelp, we construct a set of FRCs based on the given user-user relations and user-item interactions. We take book descriptions and user reviews as input of word2vec, and then output the feature vectors of books and users. We predict the interaction probability between users and books.

[4] https://www.yelp.com/dataset/challenge
[5] https://book.douban.com

Table 1. Statistics of datasets.

Datasets	Nodes	#Nodes	Relations	#Relations
Yelp	User (U)	8,163	User-User	92,248
	Item (I)	7,900	User-Item	36,571
Douban	User (U)	12,748	User-User	169,150
	Book (B)	13,342	User-Book	224,175
FWD	User (U)	72,371	User-User	8,639,884
	Article (A)	22,218	User-Article	2,465,675
	Media (M)	218,887	User-Media	1,368,868
			Article-Media	22,218

– **Friends Watching Data (FWD)** is extracted from a real-world live FER system named WeChat Top Stories, where FRCs are explicitly displayed. Based on FWD, we construct a HSG containing nearly 313 thousand nodes and 12 million edges. Each user or item is associated with some given features (e.g., age or content vectors). We predict the interaction probability between users and articles.

4.2 Experimental Settings

Baselines. We compare the proposed SIAN against four types of methods, including feature/structure-based methods (i.e., MLP, DeepWalk, node2vec and metapath2vec), fusion of feature/structure-based methods (i.e., DeepWalk+fea, node2vec+fea and metapath2vec+fea), graph neural network methods (i.e., GCN, GAT and HAN) and social recommendation methods (i.e., TrustMF and DiffNet).

– **MLP** [10] is the most simple baselines, which is implemented with the same architecture as the prediction layer in SIAN. It takes the concatenation of feature vectors of users and items as input, and output the prediction probability of the interaction. Here we vary the size of feature vector with $\{32, 64\}$.
– **DeepWalk, node2vec** and **metapath2vec**. DeepWalk [12] and node2vec [5] are two homogeneous network embedding methods. metapath2vec [3] is a heterogeneous network embedding method based on meta-paths [15]. Here we adopt meta-paths shorter than 4 and report the best performance. We feed the embeddings of users and items into a logistic regression classifier to predict the probability of interaction. The MLP as in SIAN is also be applied here, but the performance is worse. Thus, we use the logistic regression here.
– **DeepWalk+fea**, **node2vec+fea** and **metapath2vec+fea**. With the learned embeddings , we further respectively concatenate them with the features of users and items, and use the logistic regression to evaluate performances, which derives DeepWalk+fea, node2vec+fea and metapath2vec+fea.
– **GCN**, **GAT** and **HAN**. GCN [9] and GAT [17] are graph convolutional networks designed for homogeneous graphs, while HAN [19] is designed for

heterogeneous graphs. These methods take node features as input and output the node embeddings. We learn embeddings for users and items and then predict the probability of interactions as the above method. We test the same meta-paths used in metapath2vec for HAN and report the best performance.
- **TrustMF** and **DiffNet**. TrustMF [22] factorizes social trust networks and maps users into two spaces. Here we use it to learn embeddings for users and items. Then, we employ the aforementioned method to predict the interaction probability. DiffNet [20] is a social recommendation method, which takes social relations as input to enhance user embeddings. We learn the probability of the user-item interaction by modifying the output layer with the sigmoid function.

Parameters Settings. For each dataset, the ratio of training, validation and test set is 7:1:2. We adopt Adam optimizer [8] with the PyTorch implementation. The learning rate, batch size, and regularization parameter are set to 0.001, $1,024$ and 0.0005 using grid search [1], determined by optimizing AUC on the validation set. For random walk based baselines, we set the walk number, walk length and window size as 10, 50, and 5, respectively. For graph neural network based methods, the number of layers is set to 2. For DiffNet, we set the regularization parameter as 0.001. The depth parameter is set to 2 as recommended in [20]. For other parameters of baselines, we optimize them empirically under the guidance of literature. Finally, for all methods except MLP, we set the size of feature vector as 64 and report performances under different embedding dimensions $\{32, 64\}$.

4.3 Experimental Results

We adopt three widely used metrics AUC, F1 and Accuracy to evaluate performance. The results w.r.t. the dimension of latent representation are reported in Tables 2, from which we have the following findings.

(1) SIAN outperforms all baselines in all metrics on three datasets with statistical significance ($p < 0.01$) under paired t-test. It indicates that SIAN can well capture user core concerns from multifaceted factors in FER. The improvements derive from both high-quality node representations generated from node- and type-level attentive aggregations, and the social influence coupler that digs out what users are socially inclined to. Besides, the consistent improvements on different dimensions verify that SIAN is robust to the dimension.

(2) Compared with the graph neural network methods, the impressive improvements of SIAN proves the effectiveness of the node- and type-level attentive aggregations. Especially, SIAN achieves better performances than HAN which is also designed for heterogeneous graphs with a two-level aggregation. It is because that the type-level attentive aggregation in SIAN captures heterogeneous information in multiple aspects, without being limited by the predefined meta-paths used in HAN. Moreover, the improvements also indicate the significance of our social influence coupler in FER.

(3) Social recommendation baselines also achieve promising performances, which further substantiates the importance of social influence in FER. Compared with baselines which only treat social relations as side information, the

Table 2. Results on three datasets. The best method is bolded, and the second best is underlined. * indicate the significance level of 0.01.

Dataset	Model	AUC		F1		Accuracy	
		d=32	d=64	d=32	d=64	d=32	d=64
Yelp	MLP	0.6704	0.6876	0.6001	0.6209	0.6589	0.6795
	DeepWalk	0.7693	0.7964	0.6024	0.6393	0.7001	0.7264
	node2vec	0.7903	0.8026	0.6287	0.6531	0.7102	0.7342
	metapath2vec	0.8194	0.8346	0.6309	0.6539	0.7076	0.7399
	DeepWalk+fea	0.7899	0.8067	0.6096	0.6391	0.7493	0.7629
	node2vec+fea	0.8011	0.8116	0.6634	0.6871	0.7215	0.7442
	metapath2vec+fea	0.8301	0.8427	0.6621	0.6804	0.7611	0.7856
	GCN	0.8022	0.8251	0.6779	0.6922	0.7602	0.7882
	GAT	0.8076	0.8456	0.6735	0.6945	0.7783	0.7934
	HAN	0.8218	0.8476	0.7003	0.7312	0.7893	0.8102
	TrustMF	0.8183	0.8301	0.6823	0.7093	0.7931	0.8027
	DiffNet	0.8793	0.8929	0.8724	0.8923	0.8698	0.8905
	SIAN	**0.9486***	**0.9571***	**0.8976***	**0.9128***	**0.9096***	**0.9295***
Douban	MLP	0.7689	0.7945	0.7567	0.7732	0.7641	0.7894
	DeepWalk	0.8084	0.8301	0.7995	0.8054	0.8295	0.8464
	node2vec	0.8545	0.8623	0.8304	0.8416	0.8578	0.8594
	metapath2vec	0.8709	0.8901	0.8593	0.8648	0.8609	0.8783
	DeepWalk+fea	0.8535	0.8795	0.8347	0.8578	0.8548	0.8693
	node2vec+fea	0.8994	0.9045	0.8732	0.8958	0.8896	0.8935
	metapath2vec+fea	0.9248	0.9309	0.8998	0.9134	0.8975	0.9104
	GCN	0.9032	0.9098	0.8934	0.9123	0.9032	0.9112
	GAT	0.9214	0.9385	0.8987	0.9103	0.8998	0.9145
	HAN	0.9321	0.9523	0.9096	0.9221	0.9098	0.9205
	TrustMF	0.9034	0.9342	0.8798	0.9054	0.9002	0.9145
	DiffNet	0.9509	0.9634	0.9005	0.9259	0.9024	0.9301
	SIAN	**0.9742***	**0.9873***	**0.9139***	**0.9429***	**0.9171***	**0.9457***
FWD	MLP	0.5094	0.5182	0.1883	0.1932	0.2205	0.2302
	DeepWalk	0.5587	0.5636	0.2673	0.2781	0.1997	0.2056
	node2vec	0.5632	0.5712	0.2674	0.2715	0.2699	0.2767
	metapath2vec	0.5744	0.5834	0.2651	0.2724	0.4152	0.4244
	DeepWalk+fea	0.5301	0.5433	0.2689	0.2799	0.2377	0.2495
	node2vec+fea	0.5672	0.5715	0.2691	0.2744	0.3547	0.3603
	metapath2vec+fea	0.5685	0.5871	0.2511	0.2635	0.4698	0.4935
	GCN	0.5875	0.5986	0.2607	0.2789	0.4782	0.4853
	GAT	0.5944	0.6006	0.2867	0.2912	0.4812	0.4936
	HAN	0.5913	0.6025	0.2932	0.3011	0.4807	0.4937
	TrustMF	0.6001	0.6023	0.3013	0.3154	0.5298	0.5404
	DiffNet	0.6418	0.6594	0.3228	0.3379	0.6493	0.6576
	SIAN	**0.6845***	**0.6928***	**0.3517***	**0.3651***	**0.6933***	**0.7018***

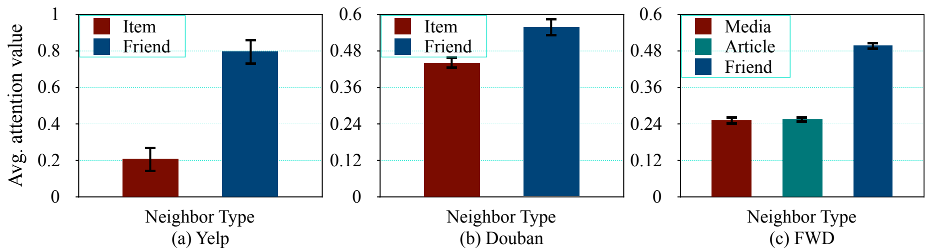


Fig. 3. Attentive aggregator analysis of *User*.

improvements imply that the friend referral factor may take the dominating position in FER, which should be carefully modeled. In particular, our SIAN achieves the best performance, reconfirming the capability of our social influence coupler in encoding diverse social factors for FER.

4.4 Impacts of Multifaceted Information

In attentive feature aggregator, each node embedding is aggregated from its neighbours of various types with different weights. We investigate the contribution of heterogeneous factors (e.g., friend, item, media), by finding the average type-level attention values (i.e., β in Eq. (4)) among all instances.

As shown in Fig. 3, the average attention value of the *Friend* type is significantly larger than that of other types. It is perhaps astonishing that the model pays more attention to users' social relationships, a notable departure from conventional recommendation where user-item interactions have thought to be more critical. This also justifies the proposed social influence coupler in SIAN, which plays an important role in extracting preferences from FRCs.

4.5 Analysis on Social Influence in FER

We have verified that FRC is the most essential factor in FER. However, a friend could impact user from different aspects (e.g., authority or similarity). Next, we show how different user attributes affect user behaviors in FER. Since we have detailed user attributes in FWD, here we conduct analysis on it.

Evaluation Protocol. The attention in social influence coupler reflects the importance of different friends. We assume that the friend v having the highest attention value (i.e., $d_{u \leftarrow \langle v,i \rangle}$ in Eq. (6)) is the most influential friend w.r.t. item i for user u, and all of v's attribute values are equally regarded as contributing to the influence. Given a user attribute and a user group, we define the *background distribution* by counting the attribute values of all friends in FRCs of users in this group, and also define the *influence distribution* by counting the attribute values of the most influential friends of users in the group. Thus, the background distribution represents the characteristics of general friends of this user group,

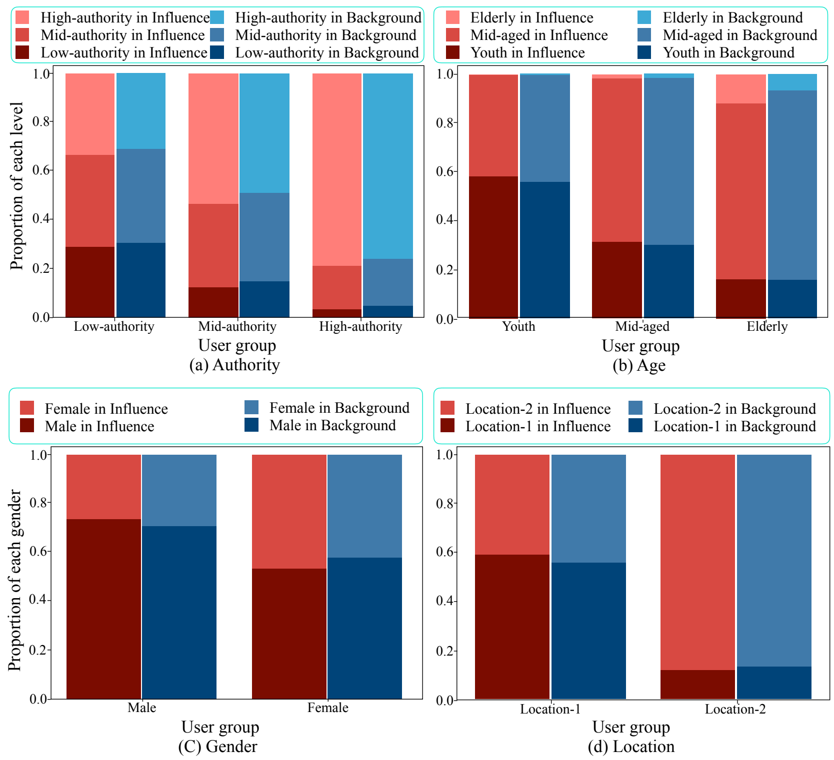
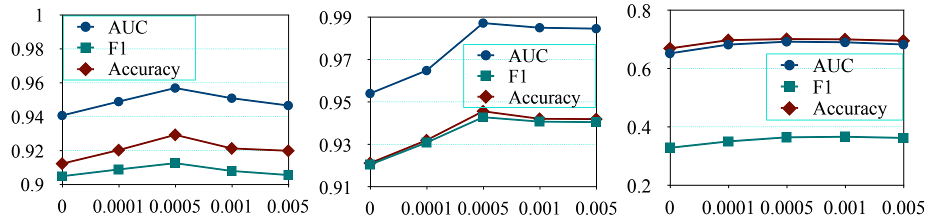


Fig. 4. Social influence analysis w.r.t user attributes. For each attribute and user group (e.g., the authority and the low-authority group in (a)), the left is the influence distribution while the right is the background distribution. In each bar, the height of each different-colored segment means the proportion of an attribute value in the influence or background distribution. Best read in color.

while the influence distribution represents the characteristics of the most influential friends of this user group. If the two distributions perfectly agree with each other, this attribute is not a key social factor in influencing this user group. In contrast, the differences between the two distributions imply how much this attribute is a key social factor, and how its different values affect user behaviors.

Results and Analysis. As shown in Fig. 4, we find out the following:

(1) In Fig. 4(a), we observe that *user behaviors are more influenced by their friends who are more authoritative, regardless of what authority the user him/herself has.* In all three user groups of varying authority, the proportion of high-authority in the influence distribution is larger than that in the background distribution. For instance, in the mid-authority user group, the top red block (high-authority influence) is larger than the top blue one (high-authority background), which implies that high-authority friends are more influential for mid-authority users. The result is not surprising as users are usually more suscep-



(a) Yelp (b) Douban (c) FWD

Fig. 5. Impact of λ in L2-regularization.

tible and easy to be affected by authoritative persons, which is consistent with common sense. It also reveals an interesting phenomenon in FER that sometimes users pay more attention to what their bosses or community authorities like, rather than what they actually like.

(2) We also conduct several analyses on influences w.r.t. other user attributes. We find that *users are easy to be influenced by their friends which are similar to themselves*. Specifically, Fig. 4(b) shows that people like items recommended by their peers, especially for the youth and the elderly; meanwhile, Fig. 4(c) and (d) show that users tend to watch articles recommended by their friends with the same gender or location. Recommendation with user similarity, which has been widely assumed in collaborative filtering, is still classical even in FER.

In conclusion, while different social factors have various influences on the target user, none of them is dominating, which further establishes the complexity of FER. In this case, the promising improvements by SIAN demonstrate that it could well capture multifaceted social factors in FER, which could potentially contribute to the understanding of interpretable recommendation.

4.6 Parameters Analysis

Our SIAN involves two parameters, i.e., the embedding dimension $d \in \{32, 64\}$ and the L2-regularization parameter λ in Eq. (11). As we have reported model performance w.r.t. d in Section 4.3, here we vary λ in the set of {0, 0.0001, 0.0005, 0.001, 0.005} to analyze its impact on model performance. As shown in Fig. 5, the optimal performance is obtained near $\lambda = 0.0005$, indicating that λ cannot be set too small or too large to prevent overfitting and underfitting.

5 Related Work

Social Recommendation. With the booming of social media, rich social information can be utilized for enhancing recommendation performance [2, 6, 11, 13, 21], which motivates the advent of social recommendation. Specifically, SoRec [11] integrates collaborative filtering with social information by proposing a probabilistic matrix factorization model. [6] incorporates the trust influence on top of SVD++, which takes the social neighbours' preferences as the side information. TrustMF [22] factorizes social trust networks and maps users into two low-dimensional spaces: truster space and trustee space. Distinct from these methods merely treating social neighbours as side information, SIAN models the social information as first-class citizens based on the unique FRC formulation.

GNN-based Social Recommendation. Recent advances in graph neural networks (GNN) have been crucial to modeling graph data [23]. Related to our work, HAN [19] embeds heterogeneous graphs with node- and semantic-level attentions, which heavily relies on the choice of predefined meta-paths. Besides, some works attempt to utilize GNNs to model user-item bipartite graphs or/and social networks. [18] integrates the knowledge graph into recommender systems, and [4] incorporates the social network into the learning of user and item latent factors. The recent DiffNet [20] models social influence with GCN. Although our SIAN also employs a GNN-based framework, it is tailored to capture multifaceted information diffusing from the FRCs through the novel node- and type-level attentive feature aggregator and social influence coupler.

6 Conclusion

In this paper, we first formulated a novel friend-enhanced recommendation problem, which is widely applicable to many social apps, and presented a social influence attentive neural network (SIAN). SIAN learns user and item representations with a two-level attentive aggregator and distills preferences from the unique friend referral circles with a social influence coupler. Experimental results demonstrate that SIAN significantly outperforms state-of-the-art baselines on three real-world datasets, and reveal interesting sociological patterns.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (No. 61772082, 61806020, 61702296), the National Key Research and Development Program of China (2018YFB1402600), and the Tencent WeChat Rhino-Bird Focused Research Program. This work is also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2018-001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. Yuanfu Lu is also supported by 2019 Tencent Rhino-Bird Elite Training Program.

References

1. Bergstra, J.S., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: NeurIPS. pp. 2546–2554 (2011)

2. Chen, C., Zhang, M., Ma, W., Zhang, Y., Liu, Y., Ma, S.: Efcient heterogeneous collaborative filtering without negative sampling for recommendation. In: AAAI (2020)
3. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: SIGKDD. pp. 135–144. ACM (2017)
4. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: WWW. pp. 417–426 (2019)
5. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: SIGKDD. pp. 855–864. ACM (2016)
6. Guo, G., Zhang, J., Yorke-Smith, N.: Trustsvd: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI (2015)
7. Jolly, A.: Lemur social behavior and primate intelligence. Science pp. 501–506 (1966)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (2017)
10. Kubat, M.: Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, ISBN 0-02-352781-7. Knowledge Eng. Review pp. 409–412 (1999)
11. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: CIKM. pp. 931–940. ACM (2008)
12. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD. pp. 701–710. ACM (2014)
13. Shen, T., Jia, J., Li, Y., Ma, Y., Bu, Y., Wang, H., Chen, B., Chua, T.S., Hall, W.: Efcient heterogeneous collaborative filtering without negative sampling for recommendation. In: AAAI (2020)
14. Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. IEEE TKDE pp. 357–370 (2018)
15. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. VLDB pp. 992–1003 (2011)
16. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Social Network Analysis and Mining pp. 1113–1133 (2013)
17. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (2018)
18. Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., Guo, M.: Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In: CIKM. pp. 417–426. ACM (2018)
19. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: WWW. pp. 2022–2032 (2019)
20. Wu, L., Sun, P., Fu, Y., Hong, R., Wang, X., Wang, M.: A neural influence diffusion model for social recommendation. In: SIGIR. pp. 235–244 (2019)
21. Xiao, W., Zhao, H., Pan, H., Song, Y., Zheng, V.W., Yang, Q.: Beyond personalization: Social content recommendation for creator equality and consumer satisfaction. In: SIGKDD. pp. 235–245 (2019)
22. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE transactions on pattern analysis and machine intelligence pp. 1633–1647 (2016)
23. Zhang, Z., Cui, P., Zhu, W.: Deep learning on graphs: A survey. IEEE TKDE (2020)