Neurocomputing 468 (2022) 469–481

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-perspective social recommendation method with graph representation learning

Hai Liu [a], Chao Zheng [b,*], Duantengchuan Li [c], Zhaoli Zhang [a], Ke Lin [d], Xiaoxuan Shen [b], Neal N. Xiong [a], Jiazhang Wang [e]

[a] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan, China
[b] National Engineering Laboratory For Educational Big Data, Central China Normal University, Wuhan, China
[c] School of Computer Science, Wuhan University, Wuhan, China
[d] Department of Control Science and Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen, China
[e] Northwestern University, Evanston, IL, USA

ARTICLE INFO

Article history:
Received 25 April 2021
Revised 23 September 2021
Accepted 19 October 2021
Available online 22 October 2021
Communicated by Zidong Wang

Keywords:
Recommender system
Diverse social relation
Graph convolutional network
Representation learning

ABSTRACT

Social recommender systems (SRS) aim to study how social relations influence users' choices and how to use them for better learning users embeddings. However, the diversity of social relationships, which is instructive to the propagation of social influence, has been rarely explored. In this paper, we propose a graph convolutional network based representation learning method, namely multi-perspective social recommendation (MPSR), to construct hierarchical user preferences and assign friends' influences with different levels of trust from varying perspectives. We further utilize the attributes of items to partition and excavate users' explicit preferences and employ complementary perspective modeling to learn implicit preferences of users. To measure the trust degree of friends from different perspectives, the statistical information of users' historical behavior is utilized to construct multi-perspective social networks. Experimental results on two public datasets of Yelp and Ciao demonstrate that the MPSR significantly outperforms the state-of-the-art methods. Further detailed analysis verifies the importance of mining explicit characteristics of users and the necessity for diverse social relationships, which show the rationality and effectiveness of the proposed model. The source Python code will be available upon request.

1. Introduction

Recommender systems have been receiving increasing attention with the surge of overload information. It is widely applied in various fields, such as online e-commerce [1], transportation [2], entertainment [3] and social media platforms [4]. As the information filtering tool, recommender systems learn and predict users' preferences through their historical behavior data. Collaborative filtering (CF) [5–8], which focuses on finding out similar users' (items') groups and collaboratively considers user feedback, is one of the most popular recommendation algorithms for modern recommender systems. Matrix factorization (MF) [6–8] can be considered a model-based CF methodology, which attempts to learn the precise feature embeddings of users and items. Despite its simplicity and efficiency, traditional CF based methods confront a major challenge: cold start [5,9]. The available rating data in commercial recommender systems is extremely sparse and its density is often less than 1%[5], which leads to the problem suffered by recommender systems–how to sift through the excessive products to find the most suitable for consumers. To this end, numerous literatures leverage extra information to tackle the above issue, such as side information [10], reviews [11,12], images [13,14] and social networks [15–17].

With the emergence of online social communities, many online platforms (e.g., Tiktok, Yelp) allow users to share preferences in their social circles, accelerating the spread of social influences. One widely accepted assumption is that friends tend to have similar preferences owing to their frequent communications. Social recommender systems (SRS) are proposed to incorporate useful information extracted from social relations [18–20], which explores the social influences on users' choices and takes advantage of friends' preferences for higher recommendation accuracy. The commonest method is to add additional prior constraints to the user representations by leveraging social relation data. For example, [15,17] proposed using L_2 regularization constraint on

* Corresponding author.
E-mail address: zhengchao9528@gmail.com (C. Zheng).

https://doi.org/10.1016/j.neucom.2021.10.050
0925-2312/

user representations to enforce users in social networks to be approximate in embedding space. This scheme, which incorporates the prior knowledge from social space, can place more reasonable priors on user feature vectors and further constrain the solution space of models. Besides, several other works [18,20] considered the priority between products and assumed that the products purchased by their friends should be more preferable to themselves than purchased by people outside their community circle. They defined a pairwise loss term and applied it to restrain the loss function.

Although above efforts had explored utilizing social information, most of them combined social relationships as constraints into the target function of the model. With the rapid development of representation learning technology, several works [16,19] directly encoded trust relations and item ratings, which are incorporated into user embeddings. Moreover, as a method of network representation learning, the graph convolutional network (GCN), which has shown its superiority in Non-Euclidean Space, is broadly applied in SRS. There are several GCN based works [21,22,23] that have been proposed to aggregate neighbors' (social neighbors and interacted items) embeddings in a unified way to acquire richer user embeddings and achieved great performance in social recommendation tasks, especially for sparse users with few purchase records.

Nevertheless, social relations should be considered from multiple perspectives, and the relationship between friends should be different from each perspective. As shown in Fig. 1(a), Bob will share "Flipped" with Lina because they have a common preference for romantic films. And "Avatar" will be shared between Bob and Tom because they are all fans of James Cameron. Therefore, Bob and Lina could be called close friends from the 'Genre' perspective, while they communicate less from the other 'Director' perspective. Actually, when buying a movie ticket, users will ask for opinions of different friends from different perspectives, and the trust degree from different friends should be considered discrepant from different perspectives (e.g., genre and director). Furthermore, from Fig. 1 (b), we can discover that Bob has a high preference for "Titanic" because it meets his needs in genre and director, though these actors were not his favorites. Based on the above analysis, we consider that Bob's preference on film could consist of the preferences from genre, actor, director and other perspectives, which is also in line with the reality that users will consider multiple dimensions (e.g., price and logistics service) based on their preferences when shopping online.

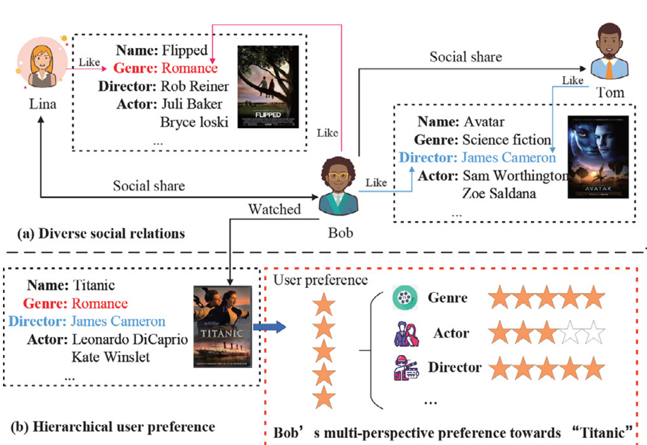

Fig. 1. An illustrative example of diverse social relations and hierarchical user preference: (a) Bob will share different films with different friends. (b) Bob's preference towards "Titanic" consists of multiple perspectives.

Thus, the diverse social relations and multi-perspective user preferences should not be ignored and motivates us to explore a hierarchical social preference model to improve SRS. Although this multi-perspective strategy is used in many areas [24,25], and some recent methods [26,27] adopt attention mechanisms to learn propagation vectors for achieving different weight aggregation. However, these approaches further increase the complexity of the model and is difficult to be applied to multi-perspective scenes. To address the above issues, we propose a multi-perspective social recommendation (MPSR) framework in this article. The major contributions of our study are summarized as follows:

- A hierarchical social recommendation framework MPSR is proposed to integrate users' multi-perspective preferences. Furthermore, MPSR efficiently captures the explicit preference from item attributes, which provides additional information for a more accurate prediction and improves model interpretability.
- To investigate the complex social relations, the trust degree between friends is reconsidered with variant values from different perspectives. The similarity of behavior between users is calculated as the trust degree by a statistical method to reconstruct the social graph network with different weights.
- Experiments are conducted on two real-world datasets to evaluate the proposed method. The results show that MPSR achieves state-of-the-art performance for social recommendation tasks, indicating its effectiveness and rationality.

The remainder of this article is organized as follows. In the next section, we introduce the related works. The detailed method and model optimization are presented in Section 3. In Section 4 experimental results and performance analysis are provided. Section 5 concludes this article.

2. Related work

2.1. Matrix factorization

As the most popular and effective method of CF algorithm, matrix factorization (MF) makes great use of "collective intelligence" through mapping the representations of users and items to a unified latent factor space. Mnih et al. [6] firstly modeled the user preference matrix as a product of two lower-rank user and item matrices from a probabilistic perspective. In [28], Koren et al. additionally accounted for the biases of individual user or item to explain the phenomenon of "nice guy" and popular products. Until today, in this booming era of artificial intelligence, MF can still be considered the basis of many effective recommender models [29].

Part of MF methods is designed for rating prediction, which aims to predict the definite scores that users give to products, and users need to show feedback on their preferences explicitly. However, in most recommended scenarios, only implicit feedback (e.g., clicks and browsers) is accessible, which leads to another recommendation task–top-N recommendation. Rendle et al. [8] presented a pair-wise personalized ranking method (Bayesian personalized ranking [BPR]) for the implicit feedback scenario, which is the first time to focus on the relative relations between the observed and unobserved items of a user. Since then, researchers [30–32] have made efforts on the top-N recommendation based on BPR strategy, which indicates the dominant role of BPR in the implicit feedback task. We also adopt the pair-wise learning method in our model because of its efficiency and effectiveness in exploring users' unobserved relative preferences.

2.2. Social recommendation

Considering the rapid spread of popular products in social circles, a large number of studies have attempted to leverage the relationships between friends for better recommendation accuracy. Depending on how social relations are used, the social recommendation methods can be grouped into three categories: prior constraint-based (PCB) approach, ranking-based (RAB) approach, and representation learning-based (RLB) approach. For PCB approaches, all consider the social network as an additional constraint. Following the traditional MF method, several authors [15,33] connected the social network structure and the user-item rating matrix and constrained a shared user latent factor vector. In [34,35], researchers incorporated heterogeneity of social relations and proposed characterized social regularization to model the characteristics of different social relations.

The RAB methods aim at comparing the ranking between items with different relations. SocialBPR [18] utilized the social influence to divide all the items into three parts: positive, social, and negative feedback, and defined the ranking between these three sets. Yu et al. [20] proposed a novel approach to identify adaptively implicit friends, and further categorized the item sets into five types: positive items, joint social items, positive social items, negative social items and non-consumed items.

For RLB approaches, user representation and social relations are encoded by embeddings vectors in the view of representation learning. Ma et al. [16] proposed a novel probabilistic factor analysis framework to fuse naturally the user tastes and the favors of his trusted friends together. In [19], Guo et al. proposed the TrustSVD method, which extended the model SVD++ [7] with social trust information and incorporated the explicit and implicit influence of item ratings as well as the user trust.

The above works could be summarized as follows. Some PCB approaches [34,36] are conscious of the heterogeneity of social relations, but cannot straightforwardly utilize the representation of one-order or higher-order social neighbors. RAB approaches intend to divide diverse explicit or implicit relations into different groups, and RLB approaches mainly focus on the construction of user or item embeddings, but neither of them considers inconsistency in the social relations. Despite these social recommendation methods make a great success, to the best of our knowledge, few studies have considered the diverse social relations as well as the different social influences, which are crucial to determining user preference.

2.3. Graph convolutional network

Convolutional neural network (CNN) has shown remarkable success in many domains, such as image process [37] and natural language processing (NLP) [38]. Given that the graph structure is naturally irregular, GCN is presented to generalize convolutions to graphs, which has shown its theoretical superiority and relatively higher performance in many graph-based tasks recently. As one of the most powerful tools in representation learning, GCN mainly leverages the information of neighbors in graphs to construct the representation of nodes. Given that nodes in a single layer of GCN can only obtain information from its first-order neighbors, information from higher-order and reachable nodes in the graph could be obtained by stacking multiple GCN layers. In Graph Convolutional Matrix Completion [39], Berg et al. presented an auto-encoder framework in graph structure from the view of link prediction to solve the problem of rating prediction. In [40], Hamilton et al. paid attention to learning a node embedding by sampling and aggregating representations of its neighborhoods. Previous studies [41,42] have attempted to consider high-order neighbor information to complement latent embedding representations of users and items, and achieved competitive performance in recommendation tasks.

Owing to the great performance of GCN, many studies [21,27,43–45] have applied GCN into social recommendations. Previous works [21,43] attempted to capture how user preferences are influenced by the social diffusion process in social networks. As the attention mechanism is adopted for graph structure data in graph attention networks [26], many attention models have been proposed to learn the social influence strength [27,44,45]. SocialGCN [21] learned the attentive weights for social neighbors and rated items for user modeling. In [27], Wu et al. applied the attention model to fuse the social network and interest network for the social recommendation. Tang et al. [45] modeled the high-order social relations and leveraged the attention mechanism to acquire information from different order neighborhoods. Although these methods had learned the weight of edges in the graph through the attention mechanism, the training cost is expensive. Our work is also inspired by the applications of attention modeling and applies its statistical characteristic to form the attentive weights and distinguish the disparate influences of friends.

3. Proposed MPSR model

For convenience, we first introduce the uniform definitions and notations used in this article. We let $\mathcal{U} = \{u_1, u_2, \ldots, u_T\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_M\}$ represent the sets of users and items, respectively. Here, T is the number of users and M is the number of items. Moreover, the matrix $\mathbf{R} \in \mathbb{R}^{T\times M}$ is utilized to denote user history behavior where each element $r_{i,j} = 1$ if an interaction (e.g., clicks and browses) is observed between user i and item j and zero otherwise. The social relationships between users are described as matrix $\mathbf{S} \in \mathbb{R}^{T\times T}$, where $s_{i,i'} = 1$ indicates that user i is connected with user i' and $s_{i,i'} = 0$ if no social connection exists. By default, all the diagonal elements in $\mathbf{S}$ can be set to one, which is considered as the user's own social connections. Moreover, the information of item attributes is also used to analyze the features of users and items. The recommendation task can be defined as follows:

Input: a user set $\mathcal{U}$, an item set $\mathcal{V}$, the user-item interaction matrix $\mathbf{R}$, the social relation matrix $\mathbf{S}$ and item attribute information.

Output: an item list $\mathcal{R}e_i$ for each user i where each item is ranked by a real value $p_{i,j} : \mathcal{V} \rightarrow \mathbb{R}$.

3.1. Overview of the framework

In Fig. 2, we illustrate the overall framework of the proposed MPSR method. First, the features of users and items from each perspective will be analyzed, which are crucial inputs for the single-perspective preferences module. Meanwhile, the social network will be also constructed based on user features from different perspectives. In each single-perspective module, the embeddings of users and items could be obtained through several submodules, which are iteratively updated in social and item space, respectively. The detailed description is presented in the next subsections.

In this article, user preferences are divided into different perspectives, aligned with the item attribute categories. As Fig. 2 shows, users will show their explicit preferences, such as genre preference, actor preference. Moreover, these explicit preferences sometimes dominate in terms of user preferences. For example, it sounds unrealistic for most people to recommend a restaurant in the city where the person has never been to, no matter how delicious the food tastes at the restaurant. Thus, the location property of the restaurant plays a crucial role in choosing one restaurant. Furthermore, an extra complementary perspective is introduced to represent some implicit preferences of users. As far as we know,

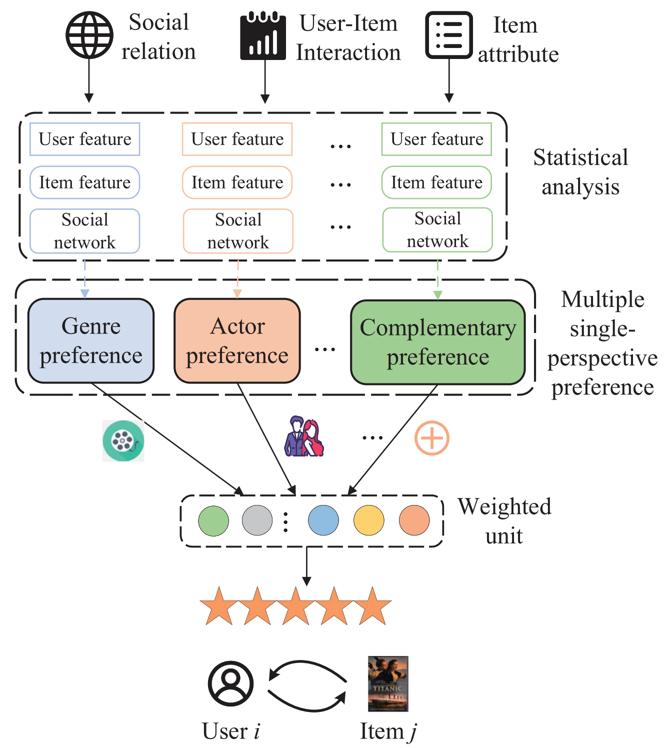

Fig. 2. Holistic framework of the proposed MPSR. User preferences comprise multiple single-perspective preference modules.

user preference can't be made up of a finite number of perspectives. Therefore, it is necessary to introduce the complementary perspective to represent some unknown perspectives. To construct a unified representation of users and items, both of them own a unique embedding under each perspective. In this study, the dot product of two embeddings is adopted to represent user preferences, accounting for its high efficiency and performance. Finally, a user preference towards an item could be obtained through a weighted unit by leveraging a group of single-perspective preferences.

3.2. Statistical analysis

For each single-perspective preference, we first acquire statistical features of users and items, and the generation process of the statistical features is shown in Fig. 3. The item feature matrix $\mathbf{I}^k \in \mathbb{R}^{M \times L_k}$ is on behalf of the attribute characteristic on the k-th perspective. The j-th row of $\mathbf{I}^k$ is depicted as,

$$\mathbf{I}_j^k = \left[I_{j,1}^k, I_{j,2}^k, \ldots, I_{j,c}^k, \ldots, I_{j,L_k}^k\right], \tag{1}$$

where L_k is the number of attribute categories in the k-th perspective and $I_{j,c}^k$ denotes an indicator, indicating whether item j belongs to category c. From the basic item feature matrix, the user feature matrix $\mathbf{F}^k \in \mathbb{R}^{T \times L_k}$, which represents statistics of user history behavior on each attribute category, is obtained through user-item interactions. The statistical feature of user i on the k-th perspective is depicted as,

$$\mathbf{F}_i^k = \left[F_{i,1}^k, F_{i,2}^k, \ldots, F_{i,c}^k, \ldots, F_{i,L_k}^k\right], \tag{2}$$

where $F_{i,c}^k$ represents the frequency of user i selection on this c category.

Then, for each user-pair $(u_i, u_{i'})$ in a social network (see in Fig. 3), their statistical characteristics are represented as $\mathbf{F}_i^k$ and

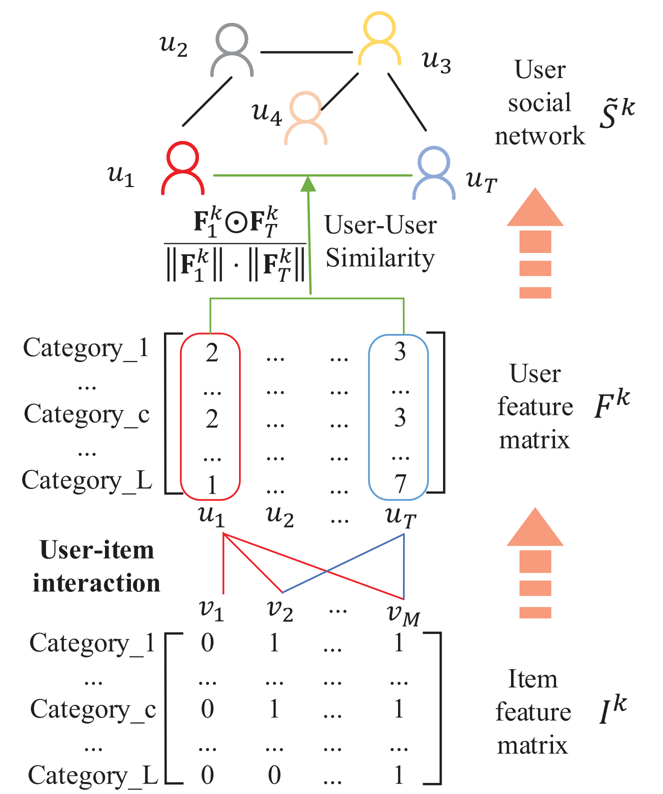

Fig. 3. Pipline of the item feature matrix and the user feature matrix, as well as social network reconfiguration in each single-perspective.

$\mathbf{F}_{i'}^k$ in the k-th perspective. Furthermore, the calculation of similarity between the above two connected users is formulated as,

$$\tilde{\mathbf{s}}_{i,i'}^k = \frac{\mathbf{F}_i^k \odot \mathbf{F}_{i'}^k}{\left\|\mathbf{F}_i^k\right\| \cdot \left\|\mathbf{F}_{i'}^k\right\|}, \tag{3}$$

where $\odot$ stands for the inner product between vectors, and $\|\cdot\|$ calculates the length of the vector. After calculating each $\tilde{\mathbf{s}}_{i,i'}^k$, every link in the social relations will be reassigned to form a social network $\tilde{\mathbf{S}}^k$ for the k-th perspective. This preprocessing does not increase the overhead of algorithm training, because all these steps can be prepared in advance.

As for the complementary perspective, we set the edge weights in its social network to one, which puts the social influence of this unknown perspective on an equal footing.

3.3. Single-perspective preference

In this subsection, we focus on modeling the single-perspective preference. Intuitively, user preference is generally believed to be influenced by their friends (the neighbors in a social network). However, most of the previous studies simply assume that users are equally influenced by their neighbors, and do not consider the differences between friends. Simply speaking, users are always affected to varying degrees by different friends for different points of interest (corresponding to the different perspectives mentioned above). The generation of single-perspective preference (see Fig. 4), which leverages distinct relationships from different perspectives, could be implemented in three steps: 1) single-perspective embedding 2) social space aggregation and 3) item space aggregation. From varying perspectives, single-perspective embedding aims to initialize representations of users and items in unified embedding space, and social space aggregation mainly aggregates the embeddings from neighbors of users from a specific

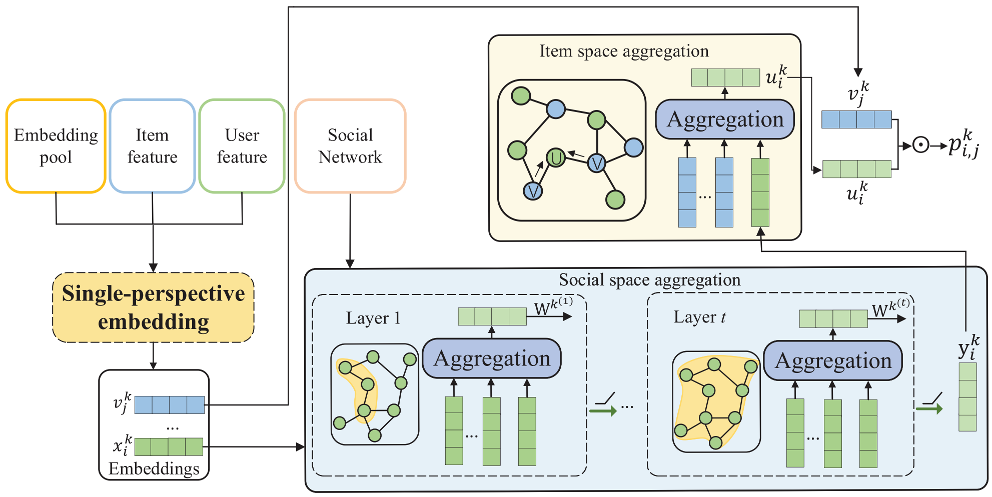

Fig. 4. Single-perspective preference module. The generation of user preference from each perspective comprises three submodules: single-perspective embedding, social space aggregation and item space aggregation.

perspective. In item space, user embedding representations further combine the information of connected items, which also relieve the sparsity problem in social network connections. Eventually, the dot product of two embeddings represents the user preference from the k-th perspective.

3.3.1. Single-perspective embedding

To construct a unified embedding representation, the item and user embeddings come from the same feature embedding pool. Suppose m explicit attributes exist, the feature embedding pool from the $k(k = 1, 2, \cdots, m)$-th perspective is represented as,

$$\mathbf{E}^k = \begin{bmatrix} & | & | & | & \\ \cdots & \mathbf{e}^k_{c-1} & \mathbf{e}^k_c & e^k_{c+1} & \cdots \\ & | & | & | & \end{bmatrix}, \tag{4}$$

where $\mathbf{E}^k \in \mathbb{R}^{d\times L_k}$, is a learnable and randomly initialized matrix. L_k is the number of attribute categories in the k-th perspective, and d denotes the size of the embedding dimension for each category embedding vector $\mathbf{e}^k_c$(here d remains the same for all k values). The category embedding vector $\mathbf{e}^k_c$ stands for the representation of a specific category (e.g., comedy, romance). In this way, the embeddings of users and items can correspond in the embedding space, and each dimension has the same hidden meaning. Moreover, because the number of attributes is limited, the number of user interactions doesn't need to be large to reflect the user's characteristics from this perspective. In general, this multi-perspective user model is more interpretable and the model coupling is significantly reduced. This preference division allows users to be close enough in embedding space once their behavior is similar from one perspective, regardless of how the user behaves from the other perspectives.

The initial representation of user i and item j from the k-th perspective becomes,

$$\mathbf{x}^k_i = \frac{\sum_{c=1,2,\ldots,L_k} F^k_{i,c} \cdot \mathbf{e^k_c}}{\sum_{c=1,2,\ldots,L_k} F^k_{i,c}}, \tag{5}$$

$$\mathbf{v}^k_j = \frac{\sum_{c=1,2,\ldots,L_k} I^k_{j,c} \cdot \mathbf{e^k_c}}{\sum_{c=1,2,\ldots,L_k} I^k_{j,c}}. \tag{6}$$

Here, $F^k_{i,c}$ and $I^k_{j,c}$ are considered as weighted average operations to normalize the data.

Although user explicit characteristics from their historical behavior often dominate user preferences, the implicit preference analysis coming from fully autonomous learning is essential, especially when the component of explicit preference in the system is one-sided or the user owns scarce behavioral data. Consequently, the complementary perspective is utilized to well represent users and items from other unknown perspectives, and directly construct pairwise learnable embeddings $\mathbf{x}^{m+1}_i \in \mathbb{R}^d$ and $\mathbf{v}^{m+1}_j \in \mathbb{R}^d$, which is also a common manner called free embeddings.

3.3.2. Social space aggregation

In social space, the social graph is connected with different weights, and the aggregation of user social embeddings is conducted from diverse perspectives. In this way, the connection strength between users from different perspectives is not consistent, and this strength is associated with the similarity between users. For each perspective (both explicit and complementary perspectives), we stack multiple GCN layers to obtain information from higher-order neighbors. After t-layer propagation, user embedding, which acquires knowledge from his t-order neighbors, could be formulated recursively as,

$$\mathbf{y}^{k^{(t)}}_i = relu\left(\sum_{i' \in \mathcal{N}_i \cup \{i\}} \mathbf{h}^k_{i,i'}(t)\right), \tag{7}$$

where $\mathcal{N}_i$ is the social neighbor set of user i, and $relu$ is the activation function. Here $\mathbf{h}^{k^{(t)}}_{i,i'}$ is the propagation embedding from user i' to user i from k-th perspective, defined as,

$$\mathbf{h}^{k^{(t)}}_{i,i'} = \frac{\widetilde{\mathbf{s}}^k_{i,i'}}{\left\|\widetilde{\mathbf{s}}^k_i\right\|_1 \cdot \left\|\widetilde{\mathbf{s}}^k_{i'}\right\|_1} \mathbf{y}^{k^{(t-1)}}_i \mathbf{W}^{k^{(t)}}, \tag{8}$$

where $\mathbf{W}^{k^{(t)}} \in \mathbb{R}^{d\times d}$ is a trainable feature transformation matrix of t-th GCN layer, and $\mathbf{y}^{k^{(t-1)}}_i$ denotes the user embedding after $(t-1)$-th GCN layer processing. The initial $\mathbf{y}^{k^{(0)}}_i$ is set to $\mathbf{x}^k_i$(in (5)). Given the nonnegativity of $\widetilde{\mathbf{s}}^k_{i,i'}$, $\left\|\widetilde{\mathbf{s}}^k_i\right\|_1 = \sum_{j=1}^{T}\left|\widetilde{\mathbf{s}}^k_{i,j}\right|$is adopted as a normalization factor. Elements on the diagonal of $\widetilde{\mathbf{S}}^k$ are equal to one, which also represents self-connection. Meanwhile, the user embedding matrix of the t-th layer in social space can be calculated by the following matrix form effectively,

$$\mathbf{Y}^{k^{(t)}} = relu\left(\mathbf{L}^k \mathbf{Y}^{k^{(t-1)}} \mathbf{W}^{k^{(t)}}\right), \tag{9}$$

and

$$\mathbf{L}^k = \mathbf{D}^{k^{-\frac{1}{2}}}\widetilde{\mathbf{S}}^k\mathbf{D}^{k^{-\frac{1}{2}}}, \tag{10}$$

where $\mathbf{L}^k \in \mathbb{R}^{T\times T}$ can be regarded as a normalized social aggregation operation. From each perspective, $\mathbf{D}^k$ represents the diagonal degree matrix where q-th diagonal element equals the sum of q-th row in $\widetilde{\mathbf{S}}^k$. Assuming the layer number of GCN layer is t, from the k-th perspective, the embedding output of user i in the social space is $\mathbf{y}_i^{k^{(t)}}$.

3.3.3. Item space aggregation

Accounting for the situation that the user seldom communicates with others on the Internet, we fuse the embeddings of connected items to the user representation. Different from social space, only a one-order neighbor is aggregated in item space because of its heterogeneity. The even-order neighborhood seems to be a user node, which may overlap with neighbors in the social space. Therefore, the information aggregation in the item space is formulated as,

$$\mathbf{u}_i^k = \mathbf{y}_i^{k^{(t)}} + \frac{\sum_{j\in\{j|r_{i,j}=1\}}\mathbf{v}_j^k}{\sum_{j=1}^{M} r_{i,j}}. \tag{11}$$

Similarly, the calculation of matrix form is illustrated as,

$$\mathbf{U}^k = \mathbf{Y}^{k^{(t)}} + \mathbf{ARV}^k, \tag{12}$$

where $\mathbf{A} \in \mathbb{R}^{T\times T}$ is a diagonal matrix, of which each q-th diagonal element represents the number of q-th user's connected items. And the j-th row of $\mathbf{V}^k$ is composed of $\mathbf{v}_j^k$.

Algorithm 1:Multi-perspective Social Recommender (MPSR) Model with Graph Representation Learning

---

Input: $\mathscr{U}$: User set, $\mathscr{V}$: Item set, $\mathbf{R}$: User-item interaction matrix, $\mathbf{S}$: Social relation matrix, Item attribute, Top-N size;

Set: dimensionality d, initial learning rate β, regularization parameter λ, dropout ratio φ, GCN layer number t;

1: Calculate each item feature matrix $\mathbf{I}^k(k = 1, 2, \cdots, m)$, each user feature matrix $\mathbf{F}^k(k = 1, 2, \cdots, m)$, each reconstructed social weight matrix $\widetilde{\mathbf{S}}^k(k = 1, 2, \cdots, m)$ as well as user degree matrix $\mathbf{A}$;
2: Initialize $\mathbf{\Theta}, \boldsymbol{\alpha}$, and other learnable parameters in our model randomly;
3: **while**not converge **do**
4: Sample negative examples D^- from training set D^+;
5: **for** $k = 1, 2, \cdots, m, m+1$ **do**
6: Initial embeddings $\mathbf{x}_i^k, \mathbf{v}_j^k$ as (5) and (6),
7: Model user representation $\mathbf{U}^k$ as (9) and (12)
8: **end for**
9: Predict user overall preference as (13);
10: Calculate the model loss and update Θ as (15);
11: **end while**
12: **for** $i = 1, 2, \cdots, T$ **do**
13: Get all unobserved preference $\left\{p_{i,j'}\middle|r_{i,j'} = 0\right\}$ after ranking;
14: **end for**

Output: Top-N recommendation list $\mathscr{R}e_i(i = 1, 2, \cdots, T) = \left\{j'\middle|p_{i,j'}\right\}, \ |\mathscr{R}e_i| = N$;

3.4. Model prediction and optimization

3.4.1. Model prediction

The illustration of Fig. 2 shows that user preference towards the specific item consists of two aspects: multiple explicit perspectives and a complementary perspective. However, simply summing up user preferences from all perspectives seems illogical. Thus, we assign different weights to different perspectives of user preferences. Through the weighted unit, the preferences of users can be depicted as,

$$p_{i,j} = \sum\nolimits_{k=1}^{m+1} att_{i,k}\mathbf{u}_i^{k^T}\mathbf{v}_j^k, \tag{13}$$

$$att_{i,k} = \frac{e^{\alpha_{i,k}}}{\sum_{p=1}^{m+1} e^{\alpha_{i,p}}}, \tag{14}$$

where each $\boldsymbol{\alpha}_i = \left[\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,m}, \alpha_{i,m+1}\right]^T$ is an $m + 1$ dimensional trainable vector, representing normalized influence of varying perspectives on user i after *softmax* operation.

3.4.2. Model optimization

Given that we merely consider user implicit feedback and focus on top-N item recommendation, we resort to BPR [8] strategy, which assumes that users' preferences towards observed interaction are higher than the unobserved ones. The loss function of the proposed model at the training stage is formulated as,

$$Loss = \sum_{i=1}^{T}\sum_{(i,j,j')\in D} -\ln\sigma\left(p_{i,j} - p_{i,j'}\right) + \lambda\|\mathbf{\Theta}\|_2^2, \tag{15}$$

where $\sigma(\cdot)$ is the sigmoid function. And the symbol $\mathbf{\Theta}$ represents the learnable embeddings of our model, containing each $\mathbf{E}^k(k = 1, 2, \cdots, m)$ and implicit embeddings $\mathbf{x}_i^{m+1}, \mathbf{v}_i^{m+1}$, which are controlled by the regularization parameter λ. $D = \left\{(i,j,j')\middle|(i,j) \in D^+, (i,j\prime) \in D^-\right\}$ denotes the training set where $D^+ = \left\{(i,j)|r_{i,j} = 1\right\}$ is the interacted user-item pair set and D^- is the unobserved record in the training step. We implement Adam [46] algorithm with a large initial learning rate 0.01 to optimize the model.

Furthermore, to prevent the overfitting problem because of the strong representation power of the GCN layer, we introduce the dropout technique [47] after a feature transformer in every GCN layer. Lastly, the detailed implementation is presented in **Algorithm 1**, which provides the top-N item list to each user.

4. Experiments

In this section, we implement experiments on two datasets in the real world to evaluate the performance of our proposed model. we emphasize these research questions:

RQ1: How does the MPSR outperform the baselines for social recommendations?.

RQ2: Why does the MPSR perform well with the multi-perspective framework and distinct social relations?.

RQ3: How does the hyperparameter of MPSR affect its performance?.

4.1. General settings

4.1.1. Datasets

The statistics of two real-world datasets are exhibited in Table 1. These two datasets provide both social relations and item attribute information.

Table 1
Statistics about the Yelp and Ciao Datasets. Relations(A-B) indicates the specific meaning of A and B, and # is a count symbol.

Datasets	Relations(A-B)	#A	#B	#(A-B)
Yelp	User-Business	16,239	14,284	198,397
	User-User	10,580	10,580	158,590
	Business-City	14,267	47	14,267
	Business-Category	14,180	511	40,009
Ciao	User-Product	6,792	103,408	273,747
	User-User	6,792	6,792	110,426
	Product-Category	103,408	28	103,408

- **Yelp.** Founded in 2004, Yelp is the largest review site in the US. This website allows users to rate restaurants, dentists and bars. More importantly, users' experiences can be shared with friends through photos and reviews, and the city and category attributes of the business are accessible.
- **Ciao.** As a popular product review site in the UK, Ciao allows people to rate products and make friends on the Internet. The category of products is also provided, and each product belongs to only one category.

The Yelp dataset [49] has dense rating relations but sparse social relations, whereas the Ciao dataset [24] has fewer users but a greater number of products. Given that the user preferences of these datasets are presented in the form of ratings, all historical records of users are transformed into positive samples considering that these interacted items satisfy at least a part of the user preferences. Furthermore, the items have only one or two attributes, which limits the ability of our model to evaluate with more perspectives.

4.1.2. Baselines and evaluation metrics

We compare the proposed MPSR methods with several representative algorithms, including the traditional and state-of-the-art social recommendation methods.

- **Most Popular (MP)**: This algorithm is non-learnable, and ranks item lists based on popularity.
- **BPR [8]**: This method is the most simple but effective learnable algorithm, which belongs to the classical MF algorithm.
- **NGCF [41]**: It is a famous GCN-based recommendation framework, which only utilizes the rating between users and items.
- **DHCF [42]**: Adopting the divide-and-conquer strategy, this method is presented recently to model different users and items flexibly with varying interactions and consider the high order relations.
- **SERec [48]**: This approach integrates social exposure into CF, which utilizes social information to capture user exposures rather than user preferences.
- **DiffNet [43]**: This method is a state-of-the-art social recommendation, which accounts for the influence of social circles, and models how user latent embeddings evolve as the social diffusion process continues.

While focusing on the top-N recommendation, to evaluate our algorithm thoroughly, three widely used ranking based metrics are chosen: Recall, Precision, and normalized discounted cumulative gain (NDCG). The Recall@N, Precision@N and NDCG@N are defined as follows:

$$Recall@N = \sum_{i=1}^{T} \frac{\left|\{j \in \mathscr{R}e_i | R_{i,j} = 1\}\right|}{T \cdot |\mathscr{R}e_i|}, \tag{16}$$

$$Precision@N = \sum_{i=1}^{T} \frac{\left|\{j \in \mathscr{R}e_i | R_{i,j} = 1\}\right|}{T \cdot N}, \tag{17}$$

$$NDCG@N = \frac{1}{T}\sum_{i=1}^{T} \frac{\sum_{n=1}^{N} R_{i,\mathscr{R}e_{i,n}} \cdot \frac{1}{\log_2(n+1)}}{\sum_{n=1}^{min(|\mathscr{R}e_i|,N)} \frac{1}{\log_2(n+1)}}. \tag{18}$$

where $|\mathscr{R}e_i|$ represent the length of the recommendation list of user i, and $\mathscr{R}e_{i,n}$ is the n-th item ID in $\mathscr{R}e_i$. Recall and Precision are the hit ratio of recommendation but the former measures how many positive items appear in the top-N list of test sets, whereas the latter reflects the proportion of the true positive sample in the recommendation list. NDCG evaluates the ranking accuracy of the recommendation list, and the more forward the position, the more important it is. For these three metrics, the larger value indicates better performance of the model.

4.1.3. Data preparation and parameter setting

In the process of data preparation, we filter out the users without social connections and ensure the users are present in both the social relation matrix and the interaction matrix. Besides, the items without attribute information are removed. The datasets are randomly partitioned, 80% are for training sets, and the rest are divided into test sets. Meanwhile, all users are chosen with at least one record in the training set to guarantee that no absolutely cold-start users appear in the test set.

Besides MP, all the baselines are based on the latent factor models. All these learnable latent vectors are initialized with small random values. Furthermore, the number of latent factors is fixed at 64, and we use Adam [46] as the optimizer to train all the models with an initial learning rate of 0.01. The learning rate can change dynamically along with the loss. In the proposed MPSR model, the grid search strategy is adopted to turn the regularization parameter λ in [0.001, 0.01, 0.1, 1, 10] and the GCN layer number t in [1, 2, 3]. Dropout ratio is searched in [0.1, 0.2, …, 0.8, 0.9, 1] to obtain a more generalized model. We experimentally set the layer number of GCN as 3 in NGCF, and as 2 in DiffNet. The layer number of GCN in DHCF is set as 2 in these two datasets. For SERec, λ_x, λ_t and λ_b are all set to one, and s equals 5 as recommended. To control the magnitude of learnable parameters, the regularization parameter is set as 0.001 in NGCF, DHCF and DiffNet on the two datasets.

4.2. Performance comparison (RQ1)

4.2.1. Overall comparison

The ranking metrics of our model and these baselines on two datasets are listed in Table 2. Given that dimension size d significantly affects the generalization ability of models, we evaluated different models in the same dimension size of 64 with varying top-N values. First, we observe that Recall and NDCG keep growing but Precision decreases as the N value increases. Given the data sparsity in these two datasets, the average number of items interacted by each user is approximately 12 in Yelp, and 40 in Ciao. Many preferred items of users are less than five, thus the length of the actual preference list in the test sets may remain while the recommendation list keeps growing as the N value becomes bigger. It may indicate the reason for the first discovery above. Second, Recall and NDCG on Yelp are always better than the metrics on Ciao which is because of the more intensive user implicit feedback on Yelp. However, for Precision, all the methods perform better on Ciao than Yelp, which may also be caused by the few connected items on the test set of Yelp.

Table 2
Comparisons on Recall@N, Precision@N and NDCG@N with Different Top-*N* Values. Bolded fonts are the best performance of each row, '_' denotes the best baseline compared with our method, and all numbers in this table are percentage numbers without '%'.

Datasets	Metrics	MP	BPR[8]	NGCF[41]	DHCF[42]	SERec[48]	DiffNet[43]	MPSR	Improv.
Yelp	Recall@5	2.0438	3.1444	3.03489	3.1259	2.9975	3.1329	**3.8428**	+22.21%
	Precision@5	1.4799	3.0837	3.1055	3.0764	3.3242	3.3789	**4.0678**	+20.39%
	NDCG@5	3.7169	4.9445	4.7122	5.1021	4.9825	5.0250	**5.2780**	+5.03%
	Recall@10	2.9469	5.0423	5.0618	4.9391	5.1861	5.4887	**6.2843**	+14.50%
	Precision@10	1.3031	2.6025	2.7100	2.5879	2.8759	2.8741	**3.5156**	+22.24%
	NDCG@10	4.1722	5.4872	5.2714	5.5460	5.5109	5.6226	**5.7830**	+2.85%
	Recall@50	9.7181	14.8036	15.8362	15.5648	15.1489	16.1003	**17.6520**	+9.64%
	Precision@50	1.0290	1.6599	1.7966	1.7204	1.7693	1.8086	**2.1072**	+16.51%
	NDCG@50	6.4456	8.6135	8.7086	8.8866	8.5616	8.9575	**8.9884**	+0.34%
Ciao	Recall@5	2.4856	2.6991	2.7389	2.5169	3.0260	2.9601	**3.3688**	+11.33%
	Precision@5	3.9180	4.6621	4.6080	4.2105	4.9897	4.9038	**5.4349**	+8.92%
	NDCG@5	4.6900	5.5707	5.3689	4.9694	5.8705	5.8224	**6.3760**	+8.61%
	Recall@10	4.0670	4.2175	4.4497	4.1387	4.8707	4.6627	**5.3784**	+10.42%
	Precision@10	3.2644	3.6190	3.7939	3.5395	4.1803	3.9291	**4.3552**	+4.18%
	NDCG@10	4.8079	5.3888	5.4110	5.0456	5.9427	5.7320	**6.3248**	+6.43%
	Recall@50	8.9784	10.3179	11.3006	10.4571	12.0186	10.9202	**12.4313**	+3.43%
	Precision@50	1.4613	1.7885	1.9587	1.8518	2.0770	1.8575	**2.1014**	+1.17%
	NDCG@50	6.0702	6.9206	7.2583	6.7220	7.8057	7.3045	**8.0247**	+2.81%

Among these baselines, the former four methods only consider the user-item interaction information whereas the latter two are further combined with social relations. Based on these experimental results, we have the following observations between the baselines.

- MP achieves the worst performances in all cases, which indicates only considering the popularity of the item can not provide users with personalized recommendations. BPR aims to model the representation of users and items, which can capture the relationships between users and items by utilizing existing interactive records. Despite its simplicity, BPR is an efficient and relatively powerful algorithm, which has more than 50% performance improvement compared with MP and even performs best at Recall@5 among all the baselines.
- As the representative recommendation framework based on graph representation learning, NGCF and DHCF achieve certain performance improvements especially when *N* is large. The different improvements show the necessity of explicitly taking connected items as a part of user characteristics when modeling users representations. DHCF performs better than NGCF on Yelp, but worse than BPR on Ciao, indicating that its divide-and-conquer strategy may not be suitable for the sparse datasets.
- Taking advantage of social relations, SERec and DiffNet perform better than the former four algorithms. It further reveals the value of social networks, especially for the recommended system, in which cold start is the main challenge. Meanwhile, we observe that the contribution of social relations and the social influence diffusion process is more obvious in Ciao than in Yelp. Among all the baselines, DiffNet performs best in most cases on Yelp, whereas SERec shows its superiority on Ciao. The reason may be that user exposures towards items need much more social information and sparse social relations hurt the recommendation accuracy. Furthermore, it also proves the important role of accounting for the user similarity in social networks.
- MPSR yields the best performance consistently on both datasets. In particular, our proposed method outperforms the most powerful baseline by over 10% improvements on the Yelp dataset in terms of Recall@N and Precision@N. Meanwhile the smaller the *N* value, the greater the performance improvement. It indicates that our model has a high accuracy rate at the head of the recommendation list, and recommendations with small *N* values are more practical in actual scenarios. In contrast, the improvement on Ciao is smaller than Yelp, though dense social relations are observed on Ciao. We attribute the greater improvement mainly to more aspects of explicit preference modeling on Yelp. Compared with SERec and DiffNet, despite both have considered the social influence, MPSR still outperforms. Thus, a more reasonable and distinguished spread of social trust in MPSR is beneficial to model user preferences.

4.2.2. Performance comparison w.r.t data sparsity

To explore the performance of our model on sparse data, the users of the test set are divided into five groups according to their historical interaction records in training sets. Figs. 5(a) and 5(e) show the proportion of users in each group. The range (0, 5] represents that the number of items connected in the training set for this group is between 0 and 5. Both datasets show the problem of data sparsity, that is, almost half of users have no more than five connected items and over 85% users have less than 50 connections with items while the number of items is 14,284 in Yelp (103,408 in Ciao). This observation highlights the sparsity challenge faced by the recommendation system.

Subsequently, we show the performance of each sparsity group with various models. In Fig. 5, $N = 10$ is chosen for a uniform comparison and the horizontal axis shows the user group information. As observed in Fig. 5, for both datasets, our model outperforms other methods in most cases, especially in the group (5, 20] and (20, 50]. It denotes the advantages of our multi-perspective framework in improving the accuracy of predicting preference on sparse users. With the increasing number of user rating records, the performance of Precision and NDCG increases quickly for all models, whereas Recall first rises and then falls when user interaction is larger than 50. Furthermore, our method surpasses other methods on Recall in all groups, and the improvement compared with the base MP reduced gradually as the interaction increases. When the connected record of each user is less than 5, MP performs better than BPR, NGCF and DHCF for Recall and NDCG on Yelp, and even obtains the best performance for NDCG@10 on Yelp. Under this sparsity group, SERec, DiffNet and MPSR leverage social relations to alleviate the influence of sparsity, and achieve comparable improvement. The improvement of Ciao is much more significant than that of Yelp, because Ciao has much denser social relations compared with Yelp. Comparing with social recommendations, our proposed model performs much better than SERec and DifferNet in most cases but obtain close and even worse performance on (100, 1500). In the densest group (100, 1500), differences between models are not obvious. NGCF, which lacks social information, surpasses SERec in Yelp. This indicates that the role of social relationships seems to be unimportant because the interactions are already enough.

4.3. Detailed discussion (RQ2)

4.3.1. Case study

To explore the existence of multiple perspectives in social relations, we choose the user (user_id = 1) and some of his friends in the Yelp dataset to conduct a case study. Fig. 6 show the users behaviors and the similarity between them from different perspectives.

In Fig. 6(a), each user is represented by a histogram, where the horizontal axis represents the different attribute categories in the city (e.g., Ahwatukee, Florence.) and the vertical axis represents the user's selection frequency in these cities. After analyzing the features of these users, we could find the similarities between the user (user_id = 1) and his different friends are significantly different, which also shows that the trust between friends of users should be considered different. Fig. 6(b) shows the users' behaviors and the similarity between them in category perspective. In this perspective, users can choose 512 different categories of activities. Different from Fig. 6(a), the similarities between users are very low, which shows these friends do not have much trust for the user (user_id = 1) in this perspective. In general, the user (user_id = 1) has different similarities with his friends in city and category perspective, which shows the social relations of the user (user_id = 1) are diverse in multiple perspectives.

4.3.2. Component analysis

To investigate the effectiveness of explicit characters in MPSR, we consider the variant of MPSR without the explicit character, named implicit aware social recommendation (IASR). IASR abandoned this weight distribution scheme in (13), and only complementary perspective preferences are preserved, compared with the base MPSR. It is equivalent to the basic GCN model with social information. For convenience, one GCN layer is adopted in MPSR and its variant. In IASR, we also adopt the grid search strategy for L_2 regularization term and dropout ratio. Experimental results show that 0.1 is the best L_2 regularization term coefficient, whereas 0.7 and 0.3 lead on Yelp and Ciao for dropout ratio, respectively.

Table 3 shows that MPSR outperforms IASR on both datasets, which further demonstrates the necessity and usefulness of introducing explicit features and multi-perspective modeling for users. This table shows that the longer the recommendation list, the greater the improvement after adding explicit features. One possible explanation might be that the multi-perspective modeling is better at exploring the associations between users and items, which leads to better accuracy in a wider range of recommendations. Furthermore, the increase in Precision is greater compared with the other two metrics, and the improvement on NDCG is minimal. It indicates our model is more concerned about the accuracy of the recommendation list and pays less attention to the priority relationship between items.

4.3.3. Effect of different social trust setting

To prove the effectiveness of statistics based social trust setting, we compare it with the other two settings in our MPSR framework. With other steps remaining the same, the Original Social MPSR (OSMPSR) abolishes the weighted social network configuration in explicit perspectives and replaces it with the original social relation network. The other Graph Attention MPSR (GATMPSR) adopts the attention mechanism in [26] to acquire learnable attentive weights and replaces the $\widetilde{\mathbf{s}}^k_{i,i'}$ in (3) with it. These two variants of MPSR are experimented and show that the L_2 coefficient and dropout ratio remain consistent with MPSR to obtain the best results. The comparisons with MPSR are shown in Fig. 7.

The experimental results show that MPSR outperforms OSMPSR and GATMPSR in all cases, indicating the effectiveness of statistics based social trust settings. Meanwhile, GATMPSR performs even worse than OSMPSR, although the difference gets smaller as N goes up. The reason might be that attentive weights are not suitable for the multi-perspective framework, and the increase of parameters makes it difficult to train the model effectively in limited data. Besides, the improvement of MPSR over OSMPSR on Ciao is much

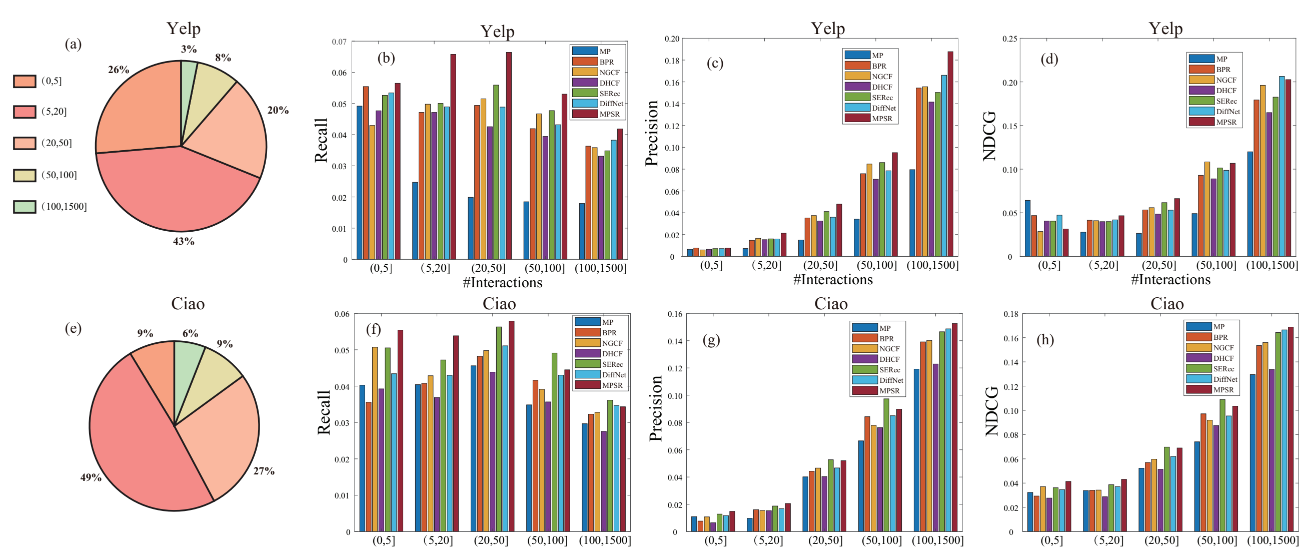

Fig. 5. Sparse distribution and experiments results of each group. (a) Sparse distribution of users on Yelp test sets. (b) Recall comparison under different sparsity group of Yelp. (c) Precision comparison under different sparsity group of Yelp. (d) NDCG comparison under different sparsity group of Yelp (e) Sparse distribution of users on Ciao test sets. (f) Recall comparison under different sparsity group of Ciao. (g) Precision comparison under different sparsity group of Ciao. (h) NDCG comparison under different sparsity group of Ciao. All performance comparison is on Top-*N* recommendation.

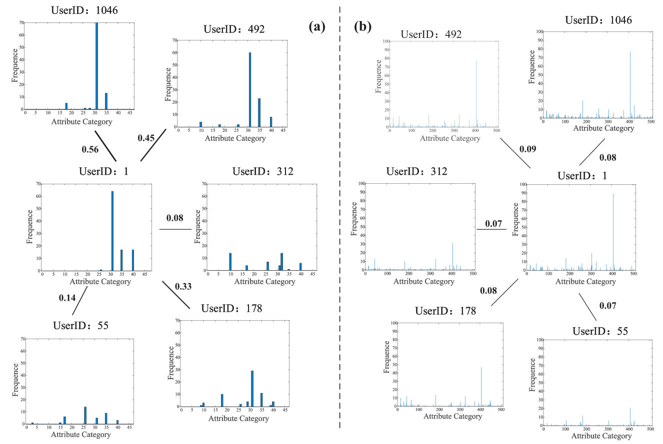

Fig. 6. Diverse user similarity from different perspectives. The decimal next to the connection line represents the similarity between the users. (a) City perspective. (b) Category perspective.

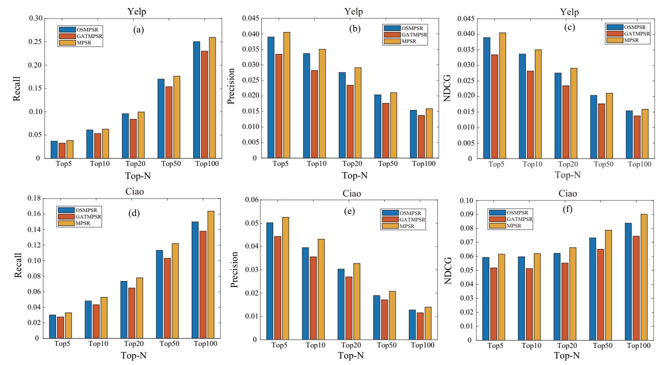

Fig. 7. Effect of different social trust setting. (a) Recall comparison on Yelp. (b) Precision comparison on Yelp. (c) NDCG comparison on Yelp. (d) Recall comparison on Ciao. (e) Precision comparison on Ciao. (f) NDCG comparison on Ciao. All comparisons are with different Top-*N* value group.

greater than Yelp, while the difference between GATMPSR and OSMPSR on Ciao is much smaller than Yelp. We attribute this difference to the social relationships with different sparsity, because modeling distinct trust might be easier to lose information especially with sparse social neighbors in Yelp.

4.4. Hyperparameter sensitivity analysis (RQ3)

In our work, L_2 regularization and embedding dropout are employed to prevent overfitting in our model. Fig. 8 displays the effect of the coefficient of L_2 regularization term λ and dropout ratio φ respectively.

As visualized in Fig. 8(a) and 8(b), applying L_2 regularization enhances the model performance. Without the L_2 regularization term, given that $-\ln(\lambda)$ moves to infinity, the performance of our model degrades dramatically based on the trajectory of the curve. Furthermore, 0.1 leads to the best performance on both datasets. Simultaneously, Fig. 8(c) and 8(d) show that embedding dropout does enhance the model performance compared with no dropout ($\varphi = 1$). Setting φ as 0.7 leads to the best performance on Yelp, whereas 0.2 is a better dropout ratio on Ciao. One reason might be that the ability of embedding representations towards different datasets should be different because of the different data characteristics. Therefore, the dropout ratio, which improves the generalization capability of the model, changes with different datasets.

To explore how the number of GCN layers affects performance, we vary the model depth in the range of [1, 2, 3]. Experimental results are shown in Fig. 9. We observe that there is little change in performance on both datasets as the layer number changes, indicating that our model is not sensitive to the number of layers. One reason might be that part of social relations with low trust will reduce noise in social networks, but also filters out some useful information when stacking more layers. Relying solely on the behavioral similarity between social relations does not seem to capture the trust relationship between users fully, which leads to a lower value of higher-order information.

5. Conclusion and future work

In this paper, we propose a novel multi-perspective social recommendation framework MPSR, which aims to implement the hierarchical model for user preferences. Considering the inconsistency of trust from different perspectives, the social network is re-established by the similarity of users' behaviors. Leveraging the information of item attributes, we can explicitly model user preference from multiple aspects. We uniformly model the embeddings of users and items under a representation framework. The graph convolutional layer is adopted to aggregate neighbouring embedding of users for a better user representation. Experimental

Table 3
Experimental Comparison between MPSR and IASR under Varing Top-N Values.

Datasets	Metrics	Models	Top5	Top10	Top50	Top100
Yelp	Recall	IASR	3.7997	6.1975	17.1412	25.1167
		MPSR	**3.8428**	**6.2843**	**17.6520**	**25.9484**
		Improv.	+1.13%	+1.40%	+2.98%	+3.31%
	Precision	IASR	3.8491	3.3151	1.9690	1.4937
		MPSR	**4.0496**	**3.5028**	**2.1068**	**1.5901**
		Improv.	+5.21%	+5.66%	+7.00%	+6.45%
	NDCG	IASR	5.0796	5.5930	8.6702	10.7288
		MPSR	**5.2322**	**5.7830**	**8.9884**	**11.1415**
		Improv.	+3.00%	+3.40%	+3.67%	+3.85%
Ciao	Recall	IASR	3.2912	5.1931	11.7176	15.7146
		MPSR	**3.2956**	**5.2940**	**12.2114**	**16.3811**
		Improv.	+0.13%	+1.94%	+4.21%	+4.24%
	Precision	IASR	5.0405	3.9529	1.8998	1.2611
		MPSR	**5.2536**	**4.3155**	**2.0808**	**1.4020**
		Improv.	+4.23%	+9.17%	+9.53%	+11.17%
	NDCG	IASR	6.1127	6.0300	7.6177	8.6037
		MPSR	**6.1630**	**6.2040**	**7.8698**	**9.0154**
		Improv.	+0.82%	+2.89%	+3.31%	+4.79%

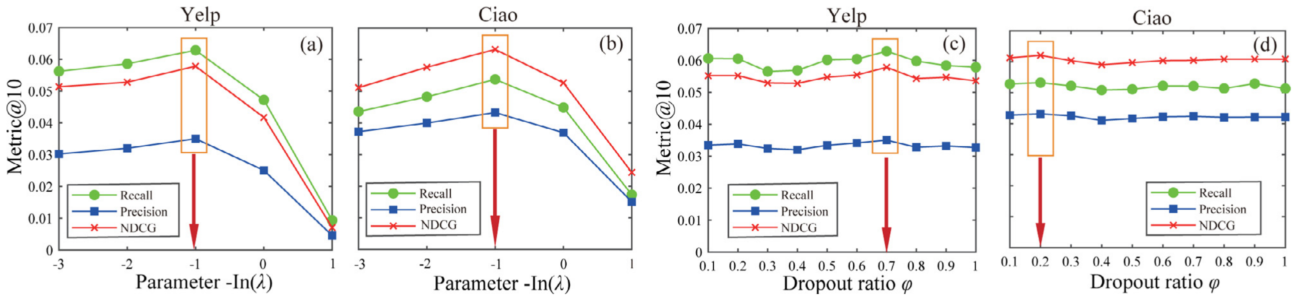

Fig. 8. Hyperparameter analysis of Top-10 recommendation on two datasets. (a) Effect of L_2 regularization term λ on Yelp. (b) Effect of L_2 regularization term λ on Ciao. (c) Effect of dropout ratio φ on Yelp. (d) Effect of dropout ratio φ on Ciao.

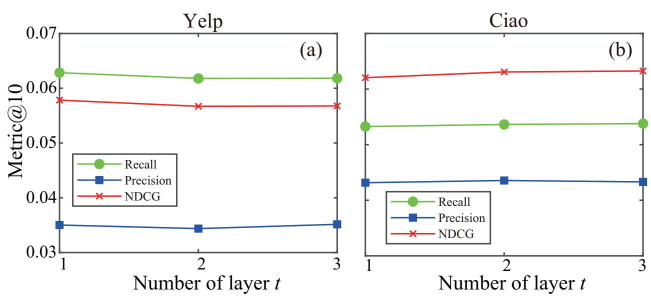

Fig. 9. Effect of GCN layer number in Top-10 recommendation on both datasets.

results show that MPSR achieves state-of-the-art results on two real-world datasets, demonstrating the effectiveness of our model and each module.

This work presents an initial attempt to exploit user preference with additional explicit attributes and proposes a new research aspect for enhancing the interpretability of user representation. The next research is to explore a more robust social relations measurement mechanism to represent the information of different social neighbors with inadequate data. Furthermore, some advanced technology, such as knowledge graph, NLP and reinforcement learning, will be introduced to extract a more reasonable prior knowledge for higher recommendation accuracy.

CRediT authorship contribution statement

Hai Liu: Writing - review & editing. **Chao Zheng:** Writing - original draft, Methodology. **Duantengchuan Li:** Writing - review & editing. **Xiaoxuan Shen:** Conceptualization. **Ke Lin:** Data curation. **Zhaoli Zhang:** Data curation. **Jiazhang Wang:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62177018, Grant 62177019, Grant 62011530436, Grant 62077020, Grant 62005092, and Grant 61875068, and China Postdoctoral Science Foundation under Grant 2020M682454, as well as the Fundamental Research Funds for the Central Universities under Grant CCNU20ZT017, Grant CCNU2020ZN008.

References

[1] Y. Wu, J. Gou, Leveraging neighborhood session information with dual attentive neural network for session-based recommendation, Neurocomputing 439 (2021) 234–242.
[2] X. Hu, J. Xu, W. Wang, Z. Li, A. Liu, A graph embedding based model for fine-grained poi recommendation, Neurocomputing 428 (2021) 376–384.
[3] Y. Hu, F. Xiong, D. Lu, X. Wang, X. Xiong, H. Chen, Movie collaborative filtering with multiplex implicit feedbacks, Neurocomputing 398 (2020) 485–494.
[4] Y. Jiang, H. Ma, Y. Liu, Z. Li, L. Chang, Enhancing social recommendation via two-level graph attentional networks, Neurocomputing 449 (2021) 71–84.
[5] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, WWW (2001) 285–295.
[6] R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, NIPS (2007) 1257–1264.
[7] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, SIGKDD (2008) 426–434.
[8] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, UAI (2009) 452–461.
[9] Y. Zhu, J. Lin, S. He, B. Wang, Z. Guan, H. Liu, D. Cai, Addressing the item cold-start problem by attribute-driven active learning, IEEE TKDE 32 (2020) 631–644.
[10] X. Shen, B. Yi, H. Liu, W. Zhang, Z. Zhang, S. Liu, N. Xiong, Deep variational matrix factorization with knowledge embedding for recommendation system, IEEE TKDE (2019), 1–1.
[11] D. Kim, C. Park, J. Oh, S. Lee, H. Yu, Convolutional matrix factorization for document context-aware recommendation, RecSys (2016) 233–240.
[12] D. Li, H. Liu, Z. Zhang, K. Lin, N.N. Xiong, Carm: Confidence-aware recommender model via review representation learning and historical rating behavior in-the-online-platforms, Neurocomputing 455 (2021) 283–296.
[13] D. Pedronette, R. Torres, Exploiting pairwise recommendation and clustering strategies for image re-ranking, Inform. Sci. 207 (2012) 19–34.
[14] L. Wu, L. Chen, R. Hong, Y. Fu, X. Xie, M. Wang, A hierarchical attention model for social contextual image recommendation, IEEE TKDE 32 (2020) 1854–1867.
[15] H. Ma, H. Yang, M.R. Lyu, I. King, Sorec: Social recommendation using probabilistic matrix factorization, in: CIKM, 2008, pp. 931–940..
[16] H. Ma, I. King, M.R. Lyu, Learning to recommend with social trust ensemble, SIGIR (2009) 203–210.
[17] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, WSDM (2011) 287–296.
[18] T. Zhao, J. McAuley, I. King, Leveraging social connections to improve personalized ranking for collaborative filtering, CIKM (2014) 261–270.
[19] G. Guo, J. Zhang, N. Yorke-Smith, Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, in: AAAI, 2015, pp. 123–129..
[20] J. Yu, M. Gao, J. Li, H. Yin, H. Liu, Adaptive implicit friends identification over heterogeneous network for social recommendation, CIKM (2018) 357–366.
[21] L. Wu, P. Sun, R. Hong, Y. Fu, X. Wang, M. Wang, Socialgcn: An efficient graph convolutional network based model for social recommendation, CoRR abs/1811.02815 (2018)..
[22] R. Ying, R. He, K. Chen, P. Eksombatchai, W.L. Hamilton, J. Leskovec, Graph convolutional neural networks for web-scale recommender systems, SIGKDD (2018) 974–983.
[23] W. Fan, Q. Li, M. Cheng, Deep modeling of social relations for recommendation, in: S.A. McIlraith, K.Q. Weinberger (Eds.), AAAI, 2018, pp. 8075–8076..
[24] J. Tang, H. Gao, H. Liu, mTrust: Discerning multi-faceted trust in a connected world, in: WSDM, 2012, pp. 93–102..
[25] X. Li, M. Chen, F. Nie, W. Qi, A multiview-based parameter free framework for group detection, in: AAAI, 2017..
[26] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, 2017..
[27] L. Wu, J. Li, P. Sun, R. Hong, Y. Ge, M. Wang, Diffnet++: a neural influence and interest diffusion network for social recommendation, IEEE TKDE (2020), 1–1.
[28] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.

[29] X. Su, T. Khoshgoftaar, A survey of collaborative filtering techniques, Adv. Artificial Intellegence, 2009..
[30] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, T.-S. Chua, Attentive collaborative filtering: multimedia recommendation with item- and component-level attention, SIGIR (2017) 335–344.
[31] W. Yu, H. Zhang, X. He, X. Chen, L. Xiong, Z. Qin, Aesthetic-based clothing recommendation, WWW (2018) 649–658.
[32] F. Zhang, N.J. Yuan, D. Lian, X. Xie, W.-Y. Ma, Collaborative knowledge base embedding for recommender systems, SIGKDD (2016) 353–362.
[33] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, RecSys (2010) 135–142.
[34] J. Tang, S. Wang, X. Hu, D. Yin, Y. Bi, Y. Chang, H. Liu, Recommendation with social dimensions, in: AAAI, 2016, pp. 251–257..
[35] T.-H. Lin, C. Gao, Y. Li, Recommender systems with characterized social regularization, CIKM (2018) 1767–1770.
[36] X. Wang, X. He, L. Nie, T.-S. Chua, Item silk road: recommending items from information domains to social users, SIGIR (2017) 185–194.
[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, NIPS, 2012.
[38] Y. Kim, Convolutional neural networks for sentence classification, 2014..
[39] R. Berg, T. Kipf, M. Welling, Graph convolutional matrix completion, 2017..
[40] W.L. Hamilton, R. Ying, J. Leskovec, Inductive representation learning on large graphs, in: NIPS, Red Hook, NY, USA, 2017, pp. 1025–1035..
[41] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, SIGIR (2019) 165–174.
[42] S. Ji, Y. Feng, R. Ji, X. Zhao, W. Tang, Y. Gao, Dual channel hypergraph collaborative filtering, SIGKDD (2020) 2020–2029.
[43] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in: SIGIR, Association for Computing Machinery, New York, NY, USA, 2019, pp. 235–244.
[44] W. Fan, Y. Ma, Q. Li, Y. He, Y.E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, CoRR abs/1902.07243 (2019)..
[45] Y. Liu, C. Liang, X. He, J. Peng, Z. Zheng, J. Tang, Modelling high-order social relations for item recommendation, IEEE TKDE (2020), 1–1.
[46] D. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2014..
[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[48] M. Wang, X. Zheng, Y. Yang, K. Zhang, Collaborative filtering with social exposure: a modular approach to social recommendation, AAAI (2017).
[49] C. Shi, Z. Zhang, Y. Ji, W. Wang, P.S. Yu, Z. Shi, Semrec: a personalized semantic recommendation method based on weighted heterogeneous information networks, WWW 22 (2019) 153–184.

Hai Liu (M'14-SM'21) received the M.S. degree in applied mathematics from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2010, and the Ph. D. degree in pattern recognition and artificial intelligence from the same university, in 2014. Since June 2017, he has been an Assistant Professor with the National Engineering Research Center for E-Learning, Central China Normal University, Wuhan. From 2017 to 2019, he was a postdoctoral fellow in the Department of Mechanical Engineering, City University of Hong Kong, Kowloon, Hong Kong, where he was hosted by the Professor You-Fu Li. In 2020, he was selected as "China-European Commission Talent Programme" under National Natural Science Foundation of China (NSFC). He is a senior researcher with UCL Interaction Centre, University College London, London, United Kingdom, where he will host by the Professor Sriram Subramanian; he will hold the position one year till February 2022. He has authored more than 70 peer-reviewed articles in international journals from multiple domains. More than six articles are selected as the ESI highly cited articles. His current research interests include deep learning, artificial intelligence, recommendation system, head pose estimation, gaze estimation, educational technology and pattern recognition. Dr. Liu has been frequently serving as a reviewer for more than six international journals including the IEEE Translations on Industrial Informatics, IEEE Translations on Cybernetics, IEEE/ASME Transactions on Mechatronics, and IEEE Translations on Instrumentation and Measurement. He is also a Communication Evaluation Expert for the NSFC from 2016 to present. He won the first prize of Science and Technology Progress Award by the Hubei Province of China in 2020.

Chao Zheng received the bachelor's degree in computer science from Huazhong University of Science and Technology, Wuhan, China, in 2017. Now he is pursuing the master's degree at National Engineering Laboratory For Educational Big Data, Wuhan, China.
His main research interests include deep learning, natural language processing, recommendation systems, and data mining techniques.

Duantengchuan Li is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University. His research interests include recommendation system, representation learning, natural language processing and their applications in services computing and software engineering.

Xiaoxuan Shen received the BE and ME degrees in 2013 and 2017, and the PhD degree from (CCNU), in 2020. He is currently working as post-doctor in the National Engineering Laboratory for Educational Big Data, Central China Normal University. His research interests include deep learning, representation learning, and their applications in recommendation system and intelligent e-learning environment.

Ke Lin is currently working toward the Ph.D. degree in Control Science and Engineering at Harbin Institute of Technology (Shenzhen), Shenzhen, China. His research interests include deep reinforcement learning, machine learning, and their applications in recommendation system.

Zhaoli Zhang (M'18) received the M.S. degree in Computer Science from Central China Normal University, Wuhan, China, in 2004, and the Ph.D. degree in Computer Science from Huazhong University of Science and Technology in 2008. He is currently a professor in the National Engineering Research Center for ELearning, Central China Normal University. His research interests include human–computer interaction, signal processing, knowledge services and software engineering. He is a member of IEEE and CCF (China Computer Federation).

Neal N. Xiong (M'08-SM'12) received the Ph.D. degrees in sensor system engineering and in dependable sensor networks from Wuhan University and the Japan Advanced Institute of Science and Technology, respectively. He is currently a Professor with the Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK, USA. His research interests include cloud computing, machine learning, representation learning, and optimization theory. He is serving as the Editorin-Chief, an Associate Editor, or an Editor member for over ten international journals, including an Associate Editor for the IEEE Transactions on Systems, Man, And Cybernetics: Systems and Information Sciences.

Jiazhang Wang is a Ph.D student in the ECE Department working in the Computational Photography Lab at Northwestern University. He is interest in information retrieval, pattern recognition, computational photography, Deflectometry and eye gazing estimation.