Heterogeneous Social Recommendation Model With Network Embedding

CHANG SU, ZONGCHAO HU, AND XIANZHONG XIE
College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
Corresponding author: Xianzhong Xie (xiexzh@cqupt.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61502067, and in part by the Key Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-K201800603.

ABSTRACT Due to the number of users and items increasing sharply, data sparsity has become an extremely serious problem for recommendation systems. Social relations consist of complex and rich information, which have a good alleviation effect on sparsity problems. Heterogeneous Information Network (HIN) is excellent in modeling the complex and structural information. Hence, we integrate HIN into the social recommendation. In this paper, we propose a model named Heterogeneous Social Recommendation model with Network Embedding (HSR). The social relations are divided into direct social relations and indirect social relations. We design a novel social influence calculation method to evaluate the influence of direct social relations. Based on the heterogeneous information network embedding method, we represent indirect social relations as feature embeddings and transform the learned embeddings into user-item feature interaction matrix by outer product. The final item list for a user is generated by the method of the convolutional neural network combined with the list of items generated by direct social relations. Extensive experiments on three real-world datasets show significant improvements of our proposed method over state-of-the-art methods. Additionally, experiments show that using heterogeneous network embedding can obtain better recommendation performance.

INDEX TERMS Heterogeneous information network, social recommendation, implicit feedback, network embedding.

I. INTRODUCTION
With the rapid development of web technology, electronic commerce like JD.com, Tmall and Amazon are gradually integrated into our life. Social medias such as QQ, WeChat, Zhihu, INS and Facebook are popular among netizens. The recommendation system plays a pivotal role in enhancing the user experiences. However, with the exponential growth of the number of users and items, data sparsity has become an extremely serious problem for recommendation systems. Usually, user’s consuming behavior depends on many factors, such as social relationships, life experience, personal hobbies. Social relations consist of complex and rich information, which have a good alleviation effect on sparsity problems. With the diversification of social relations, understanding how social relations affect consuming behavior in electronic commerce has been a long-term research issue in academia and industry [1].

The similarity between users is usually used to find the most similar users or items by discovering the historical records of users. In 2004, a recommendation system based on trust perception was proposed for the first time in literature [2], which utilized the similarity between users to enhance the recommendation effects. [3]–[8] show that the integration of social relations further improves the recommendation performance and the similarity among users in purchasing items has an important correlation with the social relations of users. In [9], social relations are used in social recommendation by discriminating the strengths and weaknesses of social relations. However, the effectiveness of these traditional approaches gradually decreases. It is not enough to make up for the extremely data sparsity problem.

Recently social relations have defined broadly with different types of friendships, which is constructed by users and items, e.g. social friends, location friends, neighbour
friends [10]. Social relations consist of complex and rich information, which have a good alleviation effect on sparsity problems. To effectively describe the relations in social networks, researchers proposed the Heterogeneous Information Network (HIN) model [11], which expresses complex relationships with different types of links between nodes in social networks. The social relationship chain and social behavior chain in human life can be reasonably expressed as a sequence of nodes in a heterogeneous information network. Hence, we can know that HIN has the advantage of extracting relation features in the real world to assist the recommendation system in improving performance. Previous studies [12]–[16] analyzed the relations between users and items based on HIN and achieved good results, indicating the effectiveness of HIN in the recommended domain. However, these methods rely on the richness of the explicit data and are lack of in-depth mining and analyzing the social relations of users and items via the HIN network embedding method.

Explicit data with rich information is not always available. In [17], Song et al. proposed a social recommendation system based on implicit friendship and achieved good results. Hence, social recommendation based on heterogeneous information networks is also reasonable for implicit feedback data. In this paper, the two main research questions are how to extract social embedding reasonably to effectively alleviate the lack of explicit data, and how to effectively improve recommendation performance based on implicit feedback data.

This paper focuses on social relations and analyzes users' social relations based on HIN. As social relations are sparse and weak for a specific user, to better integrate social relations into the recommendation system, we propose a model named heterogeneous social recommendation model with network embedding. Specifically, we divide social relations into direct social relations and indirect social relations. We learn indirect social embedding via HIN2Vec [18] to obtain users’ item list and design a novel social influence for the impact of direct social relations on users’ item list. To our knowledge, it is the first attempt which adopt HIN embedding with different types of social relations for social recommendation. The main contributions of this paper can be summarized as follows,

1) We propose a novel social recommendation model named Heterogeneous Social Recommendation model with Network Embedding. We divided social relations into direct social relations and indirect social relations. We propose a novel social influence calculation method and transform direct social influence into a kind of item list for each user.

2) We analyze the indirect social relations based on the heterogeneous network embedding method, and we express it as the user-item feature interaction matrix to analyze the impact of indirect social relations on the ranking list. We use an attention mechanism to distinguish the weights of different meta-paths in heterogeneous information networks;

3) Extensive experiments on three real-world datasets demonstrate the effectiveness of the proposed model. Moreover, we show the capability of the proposed model for the cold-start prediction problem, and reveal that the social influence from HINs can improve the recommendation performance;

II. RELATED WORK

The collaborative filtering method is the earliest method used in the recommendation field [19]. In [20], Goldberg et al. firstly proposed collaborative filtering and applied collaborative filtering to the recommendation system. In [21]–[25], collaborative filtering based methods are successively used to conduct a series of improvements for the recommendation systems and achieved good results. In [26]–[29], the contents of items were mainly exploited to mine user’s behavior characteristics and item characteristics. While with the rapid increase in the number of users and items, the sparsity problem has become a serious problem. The effectiveness of these collaborative filtering approaches gradually decreases.

As deep learning is characterized by its ability to extract features that cannot be learned directly, it has been widely used in recent recommendation research works. In [30], Wang et al. combined deep learning with traditional method for the recommendation. He et al. [31] combined neural network with collaborative filtering and used multilayer perceptron to extract the interaction between users and items. Cheng et al. [32] proposed a topic model of multimodal aspect perception, using auxiliary information to learn the shared potential topics, which are used to model users’ preferences and items attributes in different aspects. In order to use different aspect information more effectively, Cheng et al. [33] proposed an Adaptive Attention-based Neural Collaborative Filtering Model (A3NCF) based on aspect level to accurately identify users’ preferences in different aspects of items. Although deep learning methods made some improvements on recommendation, they are lack of interpretability and rationality of embeddings.

While the current data in social network is relational data, it is more feasible and reasonable to use HIN embedding methods to analyze users’ social behaviors. HIN contains multiple types of nodes and multiple types of links, which has been regarded as an effective information modeling method [11]. HIN has been used in the recommendation system to extract features from rich relational data and auxiliary information. For the first time, Sun et al. [16] proposed the concept of meta-path, to analyze a HIN based on meta-path. It simply described the meta-path as the type of edges connecting two nodes and the series of node types. Feng et al. [13] proposed a method named OptRank based on HIN, and used heterogeneous information contained in the social tag system to alleviate the cold start problem. In [34], a random walk strategy based on HIN is designed for feature embeddings. It used a set of fusion functions to transform the learned node embeddings, and integrated them into an extended matrix decomposition for recommendations.
Considering the diversity of different nodes and meta-paths, weighted mechanisms are used in HIN. Shi et al. [14] proposed a model named SemRec, which comprehensively considered the similarity between users and items in weighted HIN. Considering the weight distribution of the meta-path, Shi et al. [14] proposed a collaborative filtering method based on weighted HIN, connecting users and items with the rating and flexibly using heterogeneous information to make recommendations by weighted meta-paths and weighted integration methods. Implicit embedding features of users and items based on different meta-paths are obtained by matrix decomposition in [35], and then fitted real scores by assigning different weights to the inner product.

Heterogeneous information networks are also feasible and applicable for ranking recommendation. The meta-path method in [17] is used to find implicit friends to improve the accuracy of the recommendation. Yu et al. [15] presents an implicit feedback recommendation framework in HINs, used different types of information and relations to improve the quality of the recommendation system. Hu et al. [12] proposed a new deep neural network based on a co-attention mechanism and HIN, which made a top-n recommendation based on a rich meta-path context.

In conclusion, heterogeneous information network is excellent in modeling the complex and structural information, which is a reasonable and effective method for social recommendation. Although previous works used HIN embedding, they did not consider different types of social relations. To our knowledge, it is the first attempt which adopt HIN embedding with different types of social relations for social recommendation. The proposed approach mainly divides social relations into fine-grained, and borrows the capability of the HIN2vec method to process relational data and the capability of the convolutional neural network to recognize features, which can effectively reduce the dependence on auxiliary information and achieve good recommendation results.

III. PRELIMINARIES

In this section, we define the problem formulation and present basic methods and models needed to solve the problem we are studying.

A. PROBLEM DEFINITION

This work aims to analyze user behavior characteristics based on social networks using network embedding methods, so as to provide users with ranking recommendation services. In the following, we will formally define this problem.

**Heterogeneous Social Recommendation With Network Embedding:** Given the users’ friend relationships in social networks and users’ interaction history with items, the problem can be divided into three parts: 1) calculating direct social influence for each user; 2) mining users’ indirect social relations from the constructed heterogeneous information networks; 3) generating a customized ranking list items for each user, i.e., top-k ranked results include user liked items and never visited items. The framework of Heterogeneous Social Recommendation Model with Network Embedding is shown in Figure 1.

**B. IMPLICIT DATA**

Let \( M \) and \( N \) denote the number of users and items, respectively. We define the user-item interaction matrix \( Y \in \mathbb{R}^{(M \times N)} \) from users’ implicit feedback. Hence, \( y_{ij} \in Y \) is described as follows:

\[
    y_{ij} = \begin{cases} 
    1, & \text{if (user } i, \text{ item } j) \text{ is observed;} \\
    0, & \text{otherwise,} 
    \end{cases} 
\]

where the value of 1 for \( y_{ij} \) indicates that there is an interaction between user \( i \) and item \( j \). Though the value of \( y_{ij} \) is 0, it does not mean that \( i \) does not like \( j \) absolutely. It may be caused that the user \( i \) does not know the item \( j \).

**C. HETEROGENEOUS INFORMATION NETWORK**

HIN is a special graph \( G = (V, E, A, \theta, \phi) [36] \), in which \( V \) denotes the set of \( n \) entity nodes with practical significance \( V = \{v_1, v_2, v_3, \ldots, v_n\} \), \( E \) denotes the set of edges, \( \phi \) denotes a node type mapping function \( \phi : V \rightarrow A \), and \( \theta \) denotes the mapping function of relations between nodes \( \theta : E \rightarrow R \). Each node \( v \) corresponds to a node type \( A \), i.e., \( \phi(v) \in A, v \in V \) and each edge \( e \) in network corresponds to a relation type \( R \), i.e., \( \theta(e) \in R, e \in E \), and \( |A| > 1 \) or \( |R| > 1 \).

**D. META-PATH**

The meta-path is an manually defined random walk strategy for HIN, which contains the node types in HIN. In a given HIN \( G = (V, E, A, \theta, \phi) \), the meta-path is a sequence of nodes connected by different edges. For example, in network \( G \), the meta-path \( A_1 \xrightarrow{r_1} A_2 \xrightarrow{r_2} \ldots \xrightarrow{r_{m-1}} A_m \) denotes node sequence connected by different types of edges, which starts with entity object that the type is \( A_1 \) and end with entity object that type is \( A_m \).
To better illustrate the meta-path, we take the movie dataset as an example. As shown in **Figure 2**, \(U, D, M\) are three entities in the heterogeneous information network, where \(U\) denotes the user, \(D\) denotes the director, and \(M\) denotes the movie. Meta-path describes the relations between entities. For example, the meta-path “UMDM”, indicates that the user \(u\) watch the movie \(m_1\) which was directed by the director \(d\), and the director \(d\) also directs the movie \(m_2\) at the same time. Based on the meta-path we can infer that user \(u\) also has a high probability to watch the movie \(m_2\). Therefore, the relation can be used as a strategy to constrain the random walk in the heterogeneous information network. The path (we also call it social relationship and behavior chain for user) obtained by the meta-path guided random walk is interpretable and reasonable.

### E. RANDOM WALK ON HIN

Random walk [18], [37]–[39] is a way of traversing nodes in a graph, and it is effective method for selecting adjacent nodes (one-hop nodes or multi-hop nodes). The generated node sequence indicates semantic information in the network. Random walk method is widely used in network embedding. Given a homogeneous information network \(G\) and the current node \(v_i\), the next node \(v_{i+1}\) will be determined in Eq. 2:

\[
\rho = \begin{cases} 
\frac{1}{N_{v_i}}, & N_{v_i} \neq 0; \\
0, & \text{otherwise}, 
\end{cases}
\]

(2)

where \(N_{v_i}\) denotes the number of adjacent nodes of the current node. In an HIN \(G\), due to the deviation of node types, the selection of the next node needs to consider the node type. The specific formula is as follows,

\[
P(v_{i+1} | v_i) = \begin{cases} 
\frac{1}{|N_{v_i} \cup v_i|}, & (v_{i+1}, v_i) \in E, \phi(v_{i+1}) = t + 1; \\
0, & (v_{i+1}, v_i) \in E, \phi(v_{i+1}) \neq t + 1; \\
0, & (v_{i+1}, v_i) \notin E.
\end{cases}
\]

(3)

1. Given the HIN \(G\), the length of the generated meta path \(m\) is initialized to 1, \(m \in M^0;\)
2. Randomly select a node from the node set \(V\) as the current node;
3. According to Eq. 2, determine a set of adjacent nodes of the current node;
4. According to the type of the current node and the node type in the meta-path, the next node is determined by Eq. 3. Then we add this node into the path and set the node as the current node;
5. If the current path length is less than \(l\), we go back to step 3.

### IV. PROPOSED MODEL

In this section, we present the details of the proposed model, namely Heterogeneous Social Recommendation model with Network Embedding (HSR). Specifically, HSR includes two main modules, i.e., ranking module and re-ranking module. Each module has an independent loss function to obtain the optimal local results. Ranking module contains two submodules, i.e., network embedding and convolutional neural network. We use different network embedding methods to handle different types of network structure. Heterogeneous network embedding method extracts u-type social relation embeddings, while heterogeneous network embedding method extracts the item-type social relation embeddings. Then, the embeddings learned by the network embedding

![Figure 2](image-url)

**TABLE 1.** Part of the notation and explanation.

| Notation | Explanation |
|----------|-------------|
| \(M\) | the number of users |
| \(N\) | the number of items |
| \(v_{ij}\) | user i’s value of interaction with item j according to original dataset |
| \(Y\) | interaction dataset |
| \(G\) | heterogeneous information network |
| \(V\) | the set of all nodes |
| \(v_i\) | node i in \(G\), \(v_i \in V\) |
| \(E\) | the set of all relations |
| \(\theta\) | the mapping function of relations between nodes |
| \(\phi\) | the map function of entity node types |
| \(R\) | the set of all relation classes |
| \(r\) | a kind of relation \(r \in R\) |
| \(A\) | the set of all node types |
| \(M_p\) | the set of all meta-paths |
| \(i\) | node type, \(t \in A\) |
| \(m_j\) | the j-th meta-path |
| \(\omega^{m_j}\) | the weight of the j-th meta-path |
| \(u_i\) | user i’s friends |
| \(\rho\) | the probability of selecting the next node in heterogeneous information network |
| \(N_{C_i}\) | the number of adjacent nodes of the current node |
| \(P(v_{i+1} | v_i)\) | the probability of selecting the next node in heterogeneous information network |
| \(\beta\) | the damping coefficient |
| \(\Gamma(u)\) | social influence of user u |
| \(\Psi(u)\) | social embeddings of user u |
| \(\Delta(j)\) | the interaction matrix of user i for item j |
| \(N\) | sample number |
| \(r_{ij}\) | the predicted preference value of item j by user i |
| \(C_{ij}\) | number of user i’s friends who viewed item j |
method are combined to generate the user-item feature matrix by using outer product. Then, the preference feature matrix is further learned by the convolutional neural network method, to generate the user’s preference value for the item. The preference value is taken as the basis to get the item list for each user. Re-ranking module is a method we designed to obtain the item list for each user by evaluating the influence of direct social relations. The final item list for a user is generated by the method of the convolutional neural network combined with the list of items generated by re-ranking module.

A. THE NETWORK EMBEDDING

Network embedding is a method for data feature representation. The heterogeneous network embedding method focuses on the differences between network nodes, aiming to map each node in the network into a low-dimensional dense vector in a more readable and efficient way according to relations between nodes. With various methods of network embedding proposed recently, the application of network embedding has become more and more widespread. Inspired by [18], [40], we apply network embedding methods to build recommendation systems, as shown in Figure 3. HIN can consider the interactions between users and items. The semantic representation is generated through a given meta-path (random walk strategy). Then, we utilize the HIN2Vec [18] method to obtain the representation of the potential characteristics of users and items.

Based on social networks, we use the method of network embedding to analyze the potential social relations of each user and use it to modify the user’s list of items inspired by [41]. In order to effectively extract the social relations, we use different network embedding methods to extract different indirect social relations. In the M-type indirect social relations. If $u_1$ and $u_3$ have the same friend $u_5$, and the relation between $u_1$ and $u_3$ is a $u$-type indirect social relationship. Each item-type social relation has corresponding classification properties. Hence, we adopt different network embedding methods to extract different indirect social relations.

![FIGURE 3. Feature embedding process illustration based on HIN. The interaction network on the left part indicates the interactions between users and items. The semantic representation is generated through a given meta-path (random walk strategy). Then, we utilize the HIN2Vec [18] method to obtain the representation of the potential characteristics of users and items.](image-url)

![FIGURE 4. (a) shows the U-type indirect social relations, and (b) shows the M-type indirect social relations.](image-url)

Item-type social relations mainly refer to the second-order relation between two users due to the item. For different social relationships, we use different network embedding methods to learn social embeddings. The node2vec [38] method is taken to extract $u$-type social relations, and the social embedding of each user is denoted as a vector $u_i = \text{node2vec}(u)$.
HIN2Vec [18] method is adopted to learn item-type social relations, and the social embedding of each user is denoted as a vector \( u_{i2} \), which can discriminate between different objects and extract features of different social relations. The meta-paths here are obtained by the meta-path described in preliminary.

To effectively use direct social relations in our model, we design a novel social influence to evaluate the influence of direct social relations on the item list for users. As shown in Eq. 6:

\[
\Gamma(u) = \sum_{i \in u} \text{sim}(u, u'_i, I) \text{SL}(u),
\]

where \( \text{sim}(u, u'_i, I) = \left| \frac{(u \rightarrow I) \cap (u'_i \rightarrow I)}{(u \rightarrow I) \cup (u'_i \rightarrow I)} \right| \) indicates the similarity between user \( u \) and his/her friend \( u'_i \) based on item set \( I \). \( |\cdot| \) indicates the number of user \( i \)'s friends in the social interaction network, \( u \rightarrow I \) indicates the item set viewed by user \( u \), and \( \text{SL}(u) \) is the social liveness proposed in [42] that a user’s social behavior is effected by the number of his/her friends as shown in Eq. 7.

\[
\text{SL}^{l+1}(u) = 1 - \beta + \beta \sum_{v \in N(u)} \frac{\text{SL}(u) \cap N(v)}{|N(v)|},
\]

where \( \beta \) is the damping coefficient. It usually has a fixed value is 0.85 [42].

### C. CONVOLUTIONAL NEURAL NETWORK

Convolutional Neural Network (CNN) has been widely used to solve image recognition problem. We adopt CNN to identify the user-item interaction characteristics matrix. CNN consists of four parts: the embedding layer, the convolutional layer, the pooling layer, and the fully connected layer. We removed the embedding layer. The network structure as \( \text{cnn}() \) used in this paper is as follows,

1. **Convolutional layer**: \( h_1 = f(\omega_{11} \odot \Delta_{ij} \cdot \ell_{i} \cdot \mathbf{1}_{j} + b_{11}) \), where the batch is dealing with the amount of data at a time, \( \omega_{11} \) is for convolution kernels, \( \odot \) denotes the convolution operation, \( b_{11} \in R \) is bias, \( f(\cdot) \) is a non-linear function, which is linear rectifier function (Rectified Linear Unit, ReLU);
2. **Convolutional layer**: \( h_{11} = f(\omega_{12} \odot h_1 + b_{12}) \), \( \omega_{12} \) is for convolution kernels, \( b_{12} \in R \) is for bias, \( f(\cdot) \) is a non-linear function, which also is ReLU;
3. **Pooling layer**: Max pooling \( d_{f} = \max(h_{11}) \);
4. **Full connection layer**: convert \( d_{f} \) into one-dimension vector \( F \);
5. **Output**: \( r_{ij} = \text{softmax}(F) \);
6. **Loss function**: cross entropy loss function:

\[
L = -\frac{1}{N} \sum [y_{ij} \log r_{ij} + (1 - y_{ij}) \log(1 - r_{ij})]
\]

where \( N \) is the number of samples, \( y_{ij} \) is defined in Eq. 1 and \( r_{ij} \) is the predicted scores. We implement the CNN model based on Keras.\(^1\)

### D. HSR

In this section, we introduce our final model HSR based on the work introduced before with different network embedding methods and different user’s relations (social relations and user-item interaction relations), we can obtain the embeddings of users and items. This section mainly shows that how to obtain the user’s preference value for the item. First, to alleviate the weight difference of different meta-paths, inspired by the literature [43], we use the attention mechanism for different meta-paths as the basis for weight distribution. We use \( \omega^{m} \) to indicate the importance of the meta-path, which is defined as follows:

\[
\omega^{m} = \sigma(w_2[v_{m} + b_1] + b_2)
\]

where \( v_{m} \) denotes the embedding of node \( v \) with meta-path \( m \). The node \( v \) can be a user or an item. \( \sigma(\cdot) \) is the non-linear activation function Relu. We then normalize the above importance of all meta-paths via a softmax function:

\[
\omega^{m} = \frac{\exp(\omega^{m})}{\sum_{m=1}^{M} \exp(\omega^{m})}
\]

Then, the item embeddings \( I_k \) can be defined as following:

\[
I_k = \sum_{j=1}^{M} \omega^{m_{j}}
\]

Similarly, we learn the user embeddings \( u_{i2} \) based on indirect social relations:

\[
u_{i2} = \sum_{j=1}^{M} \omega^{m_{j}}
\]

Then, the embeddings of the user \( U \) are shown as follows:

\[
U = \alpha \ast u_{i1} + (1 - \alpha) \ast u_{i2}
\]

According to the user embeddings and item embeddings learned by Eq. 11 and Eq. 13, it is more reasonable to use the outer product method to generate user-item preference matrix based on user features and item features. The advantage of interaction matrix lies in the two-dimensional transformation of users preferences for items, which not only preserves more useful information, but also enriches the eigenvalues of one-dimensional features and reduces the influence of fluctuations of a single feature. We generate two-dimensional matrix of user-item interaction characteristics by using outer product as following:

\[
\Delta(ij) = U_i \cdot I_j
\]

\(^1https://keras.io/\)
TABLE 2. The statistics of three datasets used in our work.

| Dataset       | Relations(A-B) | Number of A | Number of B | Number of (A-B) | Average degrees of A | Average degrees of B | Density |
|---------------|----------------|-------------|-------------|-----------------|----------------------|----------------------|---------|
| Yelp          | user-movie     | 13,367      | 12677       | 1068278         | 79.9                 | 84.3                 | 0.681%  |
|               | user-user      | 2440        | 2294        | 4085            | 1.7                  | 1.8                  |         |
|               | movie-cat.     | 12678       | 38          | 27688           | 2.2                  | 728.1                |         |
| Douban Movie  | user-book      | 13024       | 22347       | 792026          | 60.8                 | 35.4                 | 0.272%  |
|               | user-user      | 12748       | 12748       | 169150          | 13.3                 | 13.3                 |         |
|               | book-cat.      | 22347       | 10806       | 22347           | 1                    | 2.1                  |         |
| Douban Book   | user-book      | 12884       | 33245       | 128311          | 9.9                  | 3.9                  | 0.029%  |
|               | user-user      | 12884       | 12844       | 45530           | 3.5                  | 3.5                  |         |
|               | bus-cat.       | 33191       | 786         | 100990          | 3                    | 128                  |         |

Then, we use the function \(cnn(\cdot)\) defined in subsection C of section IV to calculate the user’s preference value for each item based on the user’s embeddings:

\[
r_{pi} = cnn(\bigtriangleup (ij))
\]

Finally, we add the user’s direct social influence to re-rank the item list for user \(i\):

\[
r_{ij} = r_{pi} + w_{social} \sigma(C_{fi}) (i),
\]

where \(w_{social}\) denotes the importance of direct social influence, and \(C_{fi}\) denotes the number of user’s friends who viewed item \(j\). \(\sigma(\cdot)\) is the sigmoid function to make the order of magnitude consistent with \(r_{pi}\).

V. EXPERIMENTS

In this section, we report the experimental results on three real-world datasets and compare the performance of our HSR model with different baseline approaches.

A. DATASETS AND EXPERIMENTAL SETTINGS

Datasets: We adopt three real-world datasets in the experiments to evaluate our model. The first is Yelp\(^2\), which is a large dataset from business domain including users’ rating and review behaviors for local business, and containing users’ social relations and the attribute information of business. We preprocess the raw data by removing the user with less than 20 friends. Finally, yelp dataset includes 12884 users and 33245 items with 128311 interactions. Douban Book\(^3\) is the second dataset about books, including ratings and review behaviors for books, also containing users’ social relations and the attribute information of books. It includes 13024 users and 22347 books with 792026 interactions. Douban Movie\(^4\) is the third dataset about movies, including users ratings and review behaviors for movies, also containing users’ social relations and the attribute information of movies. Douban Movie dataset includes 13367 users and 12677 movies with 1068278 interactions. The details of the three datasets are shown in Table 2. Table 3 describes the meta-path of the three datasets. These three datasets are explicit feedback data. We transformed the explicit data into implicit data, where each entry is marked as 0 or 1 indicating whether the user has viewed the item, as defined in Eq. 1. To better train the model, we randomly sample 20 negative samples for each user to avoid overfitting.

B. EVALUATION METRIC

We adopt three well-known metrics \(Recall@k\), \(HitRatio@k\), and \(Precision@k\) (respectively denoted by \(Rec\), \(HR\), \(Prec\)) to evaluate recommendation performances. \(Rec\) quantifies the fraction of consumed items that are in top-k list sorted by their estimated rankings. \(Rec@k\) measures the fraction of the top-k list that are indeed consumed by the user. \(HR@k\) measures whether the test item is presented on the top-k list. The definitions of these metrics are as follow:

\[
Precision@k = \frac{1}{\mathcal{M}_t} \sum_{i=1}^{\mathcal{M}_t} \frac{|S_i(k) \cap T_i|}{k}, \quad (17)
\]

\[
Recall@k = \frac{1}{\mathcal{M}_t} \sum_{i=1}^{\mathcal{M}_t} \frac{|S_i(k) \cap T_i|}{|T_i|}, \quad (18)
\]

\[
HR@k = \frac{\sum_{i=1}^{\mathcal{M}_t} |S_i(k) \cap T_i|}{\sum_{i=1}^{\mathcal{M}_t} |T_i|}, \quad (19)
\]

where \(\mathcal{M}_t\) is the number of the testing users, \(|\cdot|\) denotes the size of the set, \(S_i(k)\) is the set of top-\(k\) items recommended to user \(i\), \(T_i\) is the set of items that are viewed by user \(i\) in the test set.

C. BASELINE

We compared our proposed HSR method with the following methods.
FIGURE 5. The performances of different recommendation methods on the Yelp dataset with respect to different k values.

FIGURE 6. The performances of different recommendation methods on the Movie dataset with respect to different k values.

BPR [44]: This is a basic method based on pairwise learning.

WRMF [40]: This is a classic matrix decomposition algorithm, which adds a weight to each training sample to denote the user’s confidence in item preference.

SBPR [45]: This method assumes that users tend to assign higher ranks to items that their friends prefer.

HERec-rank: HERec [34] is an HIN based model for rating prediction. We modify its optimization objective as pairwise ranking loss as in BPR [44] for top-k recommendation.

NeuMF [31]: A classic neural-based method for recommendation systems.

GraphRec [43]: This method is a social recommendation method based on graph neural network, which obtains embeddings of users and items by building user-user graph and user-item graph, which achieves good recommendation results.

CNN: In this approach, we adopt the same network structure as in subsection C of section IV, and randomly generate an interaction matrix with numpy for each user-item interaction with the same dimensions as in this paper.

D. IMPLEMENTATION DETAILS

According to the interaction data provided by the dataset between the user and the item, we randomly select 80% of the interaction data as the training set and the remaining 20% as the test data. Moreover, we take 10% of the data from the training set as the validation set to tune hyper-parameters. For each evaluation method, as the value of k increases, we take the average of the sum of small k to large k. The parameters of HSR are set as follows. The dimensions of embedding for users and items are set to 64. The weight of different social relationships is set to 0.85. Following [18], the length of the random walk is 40. We optimized the model with mini-batch Adam. The batch size is chosen from \{64, 128\}, and the learning rate is chosen from \{0.001, 0.005, 0.01\}. In the CNN module, we set the convolutional kernel values of first and second layers to \(5 \times 5\) and \(3 \times 3\), respectively, and set the filter values to 64. Note that large factors may cause overfitting and degrade the performance.

E. THE EFFECTIVENESS OF THE EXPERIMENTS

For each dataset, we split the entire datasets into a training set and a test set. To evaluate the performance of item recommendation effectively, we used three evaluate methods \(Rec@k\), \(Pre@k\) and \(HR@k\) defined in Eq. 16, Eq. 17 and Eq. 18. Empirically, we set k to 3, 5, and 10 in the experiments. The experimental results of the Yelp, movie, and book datasets are shown in Figure 5, 6, 7, respectively. Overall, we can see HSR outperforms all the baseline methods in terms of three
evaluate methods on three datasets. Next, we summarize the experimental results as follows:

1) We compare the item recommendation performances of the seven methods (HERec-rank, WRMF, BPR, SBPR, NeuMF, CNN, GraphRec). In order to show the result of the experiments more clearly, we show the result of $k=10$ in Table 4. As shown in Table 4, in the book dataset, the proposed HSR model outperforms NeuMF and GraphRec by 1.18% and 2.36%, in terms of Prec@10. For Rec@10, compared with NeuMF and GraphRec, HSR achieves 1.68% and 6.22% improvements respectively. In the book dataset, compared with other methods, our model can achieve better results too. In the movie dataset, among all methods, the proposed HSR model achieves the best results in terms of all evaluation metrics. Similarly, in the yelp dataset, our method also achieves the best results. Results obtained from three datasets clearly demonstrate that our model can achieve more superior performances in the face of sparse datasets.

2) The results of Rec@$k$ and Pre@$k$ are shown in Figure 5, 6, 7 for three datasets. The smaller the value of $k$ is, the larger probability of the items to be accessed by the user. It is enough to reflect ranking accuracy. Among them, as the value of $k$ increases, it reflects the overall effectiveness of the ranking list. In the Douban datasets, as the $k$ value increases, the effects of our proposed method on the accuracy rate are more obvious. The results show that as the length of the item list increases, the better the performances of our model can achieve.

3) Table 5 summarizes the effectiveness of different modules of HSR. HSRns indicates that there is an attention mechanism but no direct social relations in the model. HSRna means that there are direct social relations in the model, but no attention mechanism. The attention mechanism optimizes the weight distribution problem in different meta-paths. The social influence we proposed is based on the friend similarity and the probability that the user would like to participate in social activities. We use the social influence score to re-rank the item list obtained by the interaction relations. It can be seen from Table 5 that through the direct social influence we proposed, the performances of HSR on Yelp dataset can be improved by 4.10%, 4.51%, and 0.71%, in terms of HR@10, Pre@10, and Rec@10 respectively. Figure 8, we can clearly know that the social influence proposed in this paper is effective in direct user social networks. Therefore, in a specific item recommendation scenario, simple direct social relations cannot be greatly improved in performance. When we connect the direct relations with the item, the impacts of the direct social relations becomes more clear. Therefore, compared with the item-based indirect...
social relations, the effectiveness of the proposed direct social relations is minimal. Moreover, our social relations are relatively sparse, thus the effectiveness of direct social influence is not obvious.

4) We can also note that NeuMF and GraphRec model work well among all the baselines. An intuitive explanation is that NeuMF and GraphRec utilize the multi-layer perceptron to model the complex interactions between users and items, which implies the excellence of deep neural network in capturing complex interaction relations for recommendation. Maybe there is a question why the proposed model can achieve better performances than NeuMF and GraphRec. Two potential reasons are as follows. Firstly, our model uses the HIN embedding method to extract the feature representations of users and items, and effectively uses the auxiliary information in the social network to optimize the user’s feature representations. Compared with the feature representations generated by traditional methods, the feature representations generated by our method are more effective. Secondly, instead of using one-dimensional features to describe user-item interactions, we use a two-dimensional matrix to describe user interactions with items, and the resulting feature information has become more abundant.

F. STUDY ON CLOD-START-PROBLEM

Experimental results show that the HIN is effective in solving the cold start problem. Although data in social networks is relatively rare, the features of entities in social networks could be mined via HIN. Hence, the features of users can be accurately mined based on HIN without a large amount of data. In this paper, to test the effectiveness of the model in the cold start problem, we exclude users with less than 20 social relations in Yelp dataset. Thus, the Yelp dataset is sparser than Douban dataset, with a sparsity of 0.029%. According to the experimental results in Figure 5, data sparsity has a larger impacts on traditional learning methods. The shallow methods WRMF, BPR, and SBPR have average performances on the Yelp dataset. NeuMF and HERec-rank have superiority over traditional shallow methods. Moreover, the proposed model performs the best among all the methods. The results demonstrate that the HIN can improve recommendation performances. Compared with HERec-rank, the user-item interaction matrix proposed by us can effectively retain the user’s preference characteristics to the items, and the model can effectively use the information provided by HIN to improve the recommendation performances.

G. STUDY OF WEIGHT PREFERENCE

In this section, we study the performances of the proposed social influence on the model and conduct further experiments. In Eq. 16, $w_{social}$ controls the proportion of direct social influence in the model. Hence, the $w_{social}$ indicates the effectiveness of the social influence proposed by us. By adjusting $w_{social}$, different ranking list were obtained. Figure 9 summarizes the performances of HSR with respect to different settings of $w_{social}$. As shown in Figure 9, recommendation performance varies with the changes of $w_{social}$ in the model. When $w_{social}$ is 0, it denoted that social influence did not participate in the final model. The result of Eq. 15 was taken as the final result, but the performance is inferior to $w_{social}$, which fully demonstrates the effectiveness of

![Figure 8](image_url1)

**FIGURE 8.** The performances of different recommendation methods on the Book dataset with respect to different k values.

![Figure 5](image_url2)

**TABLE 5.** The performances of different HSR variants on Yelp dataset, Douban Book dataset and Douban movie dataset. Among them, HSRns indicates that social influence is not used to correct the ranking results, HSRna indicates that the attention mechanism is not used, different meta-paths are linearly weighted, and HSR is a comprehensive model.

|                | Yelp | Pre@10 | Rec@10 | HR@10 |
|----------------|------|--------|--------|-------|
| HSRns          | 0.2658 | 0.6441 | 0.3250 |
| HSRna          | 0.2627 | 0.6329 | 0.3217 |
| HSR            | 0.2778 | 0.6487 | 0.3383 |

|                | Book | Pre@10 | Rec@10 | HR@10 |
|----------------|------|--------|--------|-------|
| HSRns          | 0.2803 | 0.2780 | 0.0817 |
| HSRna          | 0.277 | 0.2756 | 0.0816 |
| HSR            | 0.2814 | 0.2781 | 0.0820 |

|                | Movie | Pre@10 | Rec@10 | HR@10 |
|----------------|-------|--------|--------|-------|
| HSRns          | 0.2799 | 0.3529 | 0.0355 |
| HSRna          | 0.2682 | 0.3517 | 0.0347 |
| HSR            | 0.2815 | 0.3531 | 0.0556 |

![Figure 9](image_url3)

**FIGURE 9.** The influence of the weight of social relations ($w_{social}$) on the HR@k and Rec@k for the Yelp dataset. Figures (a) depict the changes in Rec@k where k=10, and figures (b) depict the changes in HR@k where k=10.
the social influence. At the same time, with the increase of \( w_{social} \), the accuracy rate of the model fluctuated to a certain extent between [0, 1]. When \( w_{social} \) is 1, the rise gradually flattened. However, \( w_{social} \) also represented the weight ratio of direct social relations to indirect social relations. When \( w_{social} \) is greater than 1, the direct social weight is obviously greater than the indirect social relations. The main work of this paper is extracting the indirect social relations, so there is a question that why not use social influence directly for recommendation. The experimental data shows that due to the sparseness of the direct social relations between users, many users are not associated with their friends in visiting items, which leads to the zero value of the social influence. Hence, proposed the social influence uses friend relationships to optimize the ranking of items based on indirect social relations.

VI. CONCLUSION

In this paper, we propose a heterogeneous social recommendation model with network embedding to effectively integrate social relations into the recommendation model. We use the network embedding method to mine users’ social embeddings and exploit social relations to improve the recommendation performances. Motivated by the intution that the user’s social relation information is helpful to improve the recommendation accuracy, we propose a new method to extract the characteristics of indirect social relations, which takes the fluctuation of social relations into account and has a certain effects on solving the cold-start problem in the social network. For future work, we would like to study whether there exists commonality in social relationships in different fields of recommendation, so as to solve the sparseness of social relations in a single field.

REFERENCES

[1] F. Xu, J. Lian, Z. Han, Y. Li, Y. Xu, and X. Xie, “Relation-aware graph convolutional networks for agent-initiated social E-Commerce recommendation,” in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage., Beijing, China, Nov. 2019, pp. 529–538.

[2] P. Massa and P. Avesani, “Trust-aware collaborative filtering for recommender systems,” in Proc. Int. Conf., Agia Napa, Cyprus, Oct. 2004, pp. 492–508.

[3] Y. Liu, P. Zhao, X. Liu, M. Wu, L. Duan, and X.-L. Li, “Learning user dependencies for recommendation,” in Proc. 16th Int. Joint Conf. Artif. Intell., Melbourne, VIC, Australia, Aug. 2017, pp. 2379–2385.

[4] H. Ma, H. Yang, M. R. Lyu, and I. King, “SoRec: Social recommendation using probabilistic matrix factorization,” in Proc. 17th ACM Conf. Inf. Knowl. Mining, Napa Valley, CA, USA, Oct. 2008, pp. 931–940.

[5] H. Ma, I. King, and M. R. Lyu, “Learning to recommend with social trust ensemble,” in Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr., Boston, MA, USA, 2009, pp. 203–210.

[6] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, “Recommender systems with social regularization,” in Proc. 4th ACM Int. Conf. Web Search Data Mining, Hong Kong, 2011, pp. 287–296.

[7] W. Li, M. Gao, W. Rong, J. Wen, Q. Xiong, R. Jia, and T. Dou, “Social recommendation using Euclidean embedding,” in Proc. Int. Joint Conf. Neural Netw., Anchorage, AK, USA, May 2017, pp. 589–595.

[8] M. Wang, X. Zheng, Y. Yang, and K. Zhang, “Collaborative filtering with social exposure: A modular approach to social recommendation,” in Proc. 22nd Conf. Artif. Intell., New Orleans, LA, USA, Feb. 2018, pp. 2516–2523.

[9] X. Wang, W. Lu, M. Ester, C. Wang, and C. Chen, “Social recommendation with strong and weak ties,” in Proc. 25th ACM Int. Conf. Inf. Knowl. Manage., Indianapolis, IN, USA, Oct. 2016, pp. 5–14.

[10] H. Li, Y. Ge, R. Hong, and H. Zhu, “Point-of-Interest recommendations: Learning potential check-ins from friends,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 975–984.

[11] C. Shi and P. S. Yu, Heterogeneous Information Network Analysis and Applications, Cham, Switzerland: Springer, 2017.

[12] B. Hu, C. Shi, W. X. Zhao, and P. S. Yu, “Leveraging meta-path based context for Top-n recommendation with a neural co-attention model,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, London, U.K., Jul. 2018, pp. 1531–1540.

[13] W. Feng and J. Wang, “Incorporating heterogeneous information for personalized tag recommendation in social tagging systems,” in Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Beijing, China, 2012, pp. 1276–1284.

[14] C. Shi, Z. Zhang, P. Luo, P. S. Yu, Y. Yue, and B. Wu, “Semantic path based personalized recommendation on weighted heterogeneous information networks,” in Proc. 24th ACM Int. Conf. Inf. Knowl. Manage., Melbourne, VIC, Australia, 2015, pp. 453–462.

[15] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, “Recommendation in heterogeneous information networks with implicit user feedback,” in Proc. 7th ACM Conf. Recommender Syst., Hong Kong, 2013, pp. 347–350.

[16] Y. Sun, J. Han, X. You, S. Sun, and T. Wu, “PathSim: Meta-path based top-k similarity search in heterogeneous information networks,” Proc. VLDB Endowment, vol. 4, no. 11, pp. 992–1003, Aug. 2011.

[17] Y. Song, M. Gao, J. Yu, and Q. Xiong, “Social recommendation based on implicit friends discovering via meta-path,” in Proc. IEEE 30th Int. Conf. Tools with Artif. Intell. (ICTAI), Volos, Greece, Nov. 2018, pp. 197–204.

[18] T.-Y. Fu, W.-C. Lee, and Z. Lei, “HIN2Vec: Explore meta-paths in heterogeneous information networks for representation learning,” in Proc. ACM Conf. Inf. Knowl. Manage., Singapore, Nov. 2017, pp. 1797–1806.

[19] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep learning based recommender system: A survey and new perspectives,” ACM Comput. Surv., vol. 52, no. 1, pp. 1–38, Feb. 2019.

[20] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, “Using collaborative filtering to weave an information tapestry,” Commun. ACM, vol. 35, no. 12, pp. 61–70, Dec. 1992.

[21] Y. Koren, “Collaborative filtering with temporal dynamics,” Commun. ACM, vol. 53, no. 4, pp. 89–97, Apr. 2010.

[22] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, “A novel deep learning-based collaborative filtering model for recommendation system,” IEEE Trans. Cybern., vol. 49, no. 3, pp. 1084–1096, Mar. 2019.

[23] Y. Liu, P. Zhao, A. Sun, and C. Miao, “A boosting algorithm for item recommendation with implicit feedback,” in Proc. 24th Int. Joint Conf. Artif. Intell., Buenos Aires, Argentina, Jul. 2015, pp. 1792–1798.

[24] P. Yang, P. Zhao, Y. Liu, and X. Gao, “Robust cost-sensitive learning for recommendation with implicit feedback,” in Proc. Int. Conf. Data Mining, San Diego, CA, USA, May 2018, pp. 621–629.

[25] Y. Liu, L. Zhao, G. Liu, X. Lu, P. Gao, X.-L. Li, and Z. Jin, “Dynamic Bayesian logistic matrix factorization for recommendation with implicit feedback,” in Proc. Int. Joint Conf. Artif. Intell., Stockholm, Sweden, Jul. 2018, pp. 3463–3469.

[26] I. Cantador, A. Bellogín, and D. Vallet, “Content-based recommendation in social tagging systems,” in Proc. 4th ACM Conf. Recommender Syst., Barcelona, Spain, 2010, pp. 237–240.

[27] H. Zhang, Y. Sun, M. Zhao, T. W. S. Chow, and Q. M. J. Wu, “ Bridging user interest to item content for recommender systems: An optimization model,” IEEE Trans. Cybern., vol. 50, no. 10, pp. 4268–4280, Oct. 2020.

[28] S. Puglisi, J. Parra-Arnau, J. Forné, and D. Rebollo-Monedero, “On content-based recommendation and user privacy in social-tagging systems,” Comput. Standards Interfaces, vol. 41, pp. 17–27, Sep. 2015.

[29] Q. Wu, Y. Liu, C. Miao, B. Zhao, Y. Zhao, and L. Guan, “PD-GAN: Adversarial learning for personalized diversity-promoting recommendation,” in Proc. Int. Joint Conf. Artif. Intell., Macao, China, Aug. 2019, pp. 3870–3876.

[30] H. Wang, N. Wang, and D. Yeung, “Collaborative deep learning for recommender systems,” in Proc. 21th Int. Conf. Knowl. Discovery Data Mining, Sydney, Australia, 2013, pp. 1235–1244.

[31] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, “Neural collaborative filtering,” in Proc. 26th Int. Conf. World Wide Web, Perth, WA, Australia, Apr. 2017, pp. 173–182.
Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. Kankanalli, “3NCF: An adaptive aspect attention model for rating prediction,” in Proc. Twenty-Seventh Int. Joint Conf. Artif. Intell., Stockholm, Sweden, Jul. 2018, pp. 3748–3754.

C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, “Heterogeneous information network embedding for recommendation,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 2, pp. 357–370, Feb. 2019.

X. Yu, X. Ren, Y. Sun, Q. Gu, B. Sturt, U. Khandelwal, B. Norick, and J. Han, “Personalized entity recommendation: A heterogeneous information network approach,” in Proc. 7th Int. Conf. Web Search Data Mining, New York, NY, USA, Feb. 2014, pp. 283–292.

C. Shi, Y. Li, J. Zhang, Y. Sun, and P. S. Yu, “A survey of heterogeneous information network analysis,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 1, pp. 17–37, Jan. 2017.

B. Perozzi, R. Al-Rfou, and S. Skiena, “DeepWalk: Online learning of social representations,” in Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA, 2014, pp. 701–710.

A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, San Francisco, CA, USA, Aug. 2016, pp. 855–864.

Y. Dong, N. V. Chawla, and A. Swami, “Metapath2vec: Scalable representation learning for heterogeneous networks,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Halifax, NS, Canada, Aug. 2017, pp. 135–144.

Y. Hu, Y. Koren, and C. Volinsky, “Collaborative filtering for implicit feedback datasets,” in Proc. 8th IEEE Int. Conf. Data Mining, Dec. 2008, pp. 263–272.

T.-H. Lin, C. Gao, and Y. Li, “Recommender systems with characterized social regularization,” in Proc. 27th ACM Int. Conf. Inf. Knowl. Manage., Turin, Italy, Oct. 2018, pp. 1767–1770.

R. Li, H. Lin, Y. Shi, and H. Wang, “SocialST: Social liveness and trust enhancement based social recommendation,” in Proc. IEEE Int. Conf. Web Services, Milan, Italy, Jul. 2019, pp. 139–145.

W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, “Graph neural networks for social recommendation,” in Proc. World Wide Web Conf., San Francisco, CA, USA, 2019, pp. 417–426.

S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” in Proc. Conf. Uncertainty Artif. Intell., Montreal, QC, Canada, Jun. 2009, pp. 452–461.

T. Zhao, J. McAuley, and I. King, “Leveraging social connections to improve personalized ranking for collaborative filtering,” in Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage., Shanghai, China, Nov. 2014, pp. 261–270.