

## Research Article

# An Efficient Recommendation Algorithm Based on Heterogeneous Information Network

Ying Yin  and Wanning Zheng

*School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning 110169, China*

Correspondence should be addressed to Ying Yin; [yinying@mail.neu.edu.cn](mailto:yinying@mail.neu.edu.cn)

Received 5 December 2020; Revised 18 February 2021; Accepted 24 February 2021; Published 5 March 2021

Academic Editor: Jianxin Li

Copyright © 2021 Ying Yin and Wanning Zheng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Heterogeneous information networks can naturally simulate complex objects, and they can enrich recommendation systems according to the connections between different types of objects. At present, a large number of recommendation algorithms based on heterogeneous information networks have been proposed. However, the existing algorithms cannot extract and combine the structural features in heterogeneous information networks. Therefore, this paper proposes an efficient recommendation algorithm based on heterogeneous information network, which uses the characteristics of graph convolution neural network to automatically learn node information to extract heterogeneous information and avoid errors caused by the manual search for metapaths. Furthermore, by fully considering the scoring relationship between nodes, a calculation strategy combining heterogeneous information and a scoring information fusion strategy is proposed to solve the scoring between nodes, which makes the prediction scoring more accurate. Finally, by updating the nodes, the training scale is reduced, and the calculation efficiency is improved. The study conducted a large number of experiments on three real data sets with millions of edges. The results of the experiments show that compared with PMF, SemRec, and other algorithms, the proposed algorithm improves the recommendation accuracy MAE by approximately 3% and the RMSE by approximately 8% and reduces the time consumption significantly.

## 1. Introduction

With the rapid development of information technology, the amount of digital information is increasing rapidly. This makes it difficult for people to correctly judge the relevance of the retrieved items and make correct decisions. Recommendation systems have arisen at a historical moment. An effective recommendation can greatly reduce information overload. Recommendation systems have changed the communication mode between users and websites and have many applications in many fields, such as the economy, education, and scientific research. For example, in academic research, a recommendation system can provide researchers with papers to help them quickly find the required papers. For junior researchers with limited experience, a recommendation system may recommend new articles and classic articles in related fields to broaden their horizons and

research interests. For senior researchers with strong research abilities, a recommendation system helps researchers reduce information overload and find relevant papers by analyzing the publication records of papers and recommends papers related to researchers' research interests or research priorities. Currently, recommendation systems have become an indispensable tool. The recommendation algorithms are constantly updated, and the recommendation accuracy increases as time passes.

At present, the existing recommendation algorithms are usually based on homogeneous networks [1]; that is, there is only one type of object or relationship in the network. For example, the author collaboration network only contains the relationships between authors and the coauthors. These isomorphic information networks usually only consider one type of relationship with one type of object. However, most real interaction systems usually contain multiple types of

interaction information, which can be modelled as heterogeneous information networks including different types of objects and connections. For example, the document database DBLP may be organized into a heterogeneous information network, including various types of objects (e.g., paper, authors, and work units) and connections (e.g., by writing relationship between paper and authors and the relationships between published papers and work units). Clearly, an author collaboration network is implicitly a heterogeneous information network, which can be obtained from the writing relationship between papers and authors.

As a new direction, heterogeneous information networks can naturally simulate complex objects and their rich relationships in recommendation systems. Objects have different types, and the connections between objects represent different relationships. Some researchers [2, 3] have proposed to evaluate the similarity of objects in heterogeneous information networks based on several path-based similarity metrics. Feng and Wang proposed the OptRank method [4] to alleviate the cold start problem by using the heterogeneous information contained in a social labelling system. In addition, the concept of metapaths has been introduced into a hybrid recommendation system. Yu et al. used metapath-based similarity as a regularization term in a matrix factorization framework [5]. Yu et al. proposed a personalized recommendation framework with implicit feedback by using different types of entity relationships in heterogeneous information networks [6]. Luo et al. proposed a collaborative filtering social recommendation method based on heterogeneous relationships [7]. Recently, Shi et al. proposed the concept of a weighted heterogeneous information network, designed a collaborative filtering model based on metapaths, flexibly integrated heterogeneous information, and realized the personalized recommendations [8]. The similarity between users and items is evaluated by path similarity measures under metapaths with different semantics, and matrix factorization based on a biregularization framework is proposed for rating prediction [9]. Most methods based on heterogeneous information networks rely on path-based similarity, which may not fully mine the potential characteristics of users and recommend projects for heterogeneous information networks.

Recommendation algorithms are based on heterogeneous information networks, such as algorithms based on matrix factorization. The main idea is to decompose the heterogeneous information network. The relationships under different metapaths are decomposed into a relationship matrix about users and a relationship matrix about entities by using the random walk method, and then the corresponding algorithms for representing users and entities are used to represent the relationships. Then, the two metrics are fused, and the corresponding recommendation results are finally obtained through continuous iteration. Although there are many such algorithms, there are still some problems: (1) *The Representation of Heterogeneous Information Networks*. Most of the existing algorithms for processing heterogeneous information

networks are based on metapath methods; that is, they must determine relational paths that must be considered to prove that two nodes are similar. If an effective metapath cannot be found accurately, the result will be inaccurate, and the data fluctuation will be too large during metapath transformation; thus, the stability of the result cannot be guaranteed. (2) *Computational Efficiency*. As the information is transmitted through the graph convolutional neural network algorithm, the amount of information increases exponentially as the number of layers increases; therefore, the computational efficiency cannot be guaranteed.

Aiming at some problems of the existing recommendation algorithms based on heterogeneous information networks, a recommendation algorithm based on heterogeneous information networks is proposed. The main contributions are as follows:

- (1) The existing graph convolution neural network graph-based depth learning algorithm is improved to allow it make recommendations for heterogeneous information networks. Therefore, the recommendation results can be combined with the information of heterogeneous information networks more accurately. This method is not only suitable for homogeneous networks, but it can also obtain the corresponding results in increasingly complex heterogeneous information networks. Furthermore, this method also solves the problems of the manual selection of metapaths and inaccuracy in heterogeneous information network recommendation.
- (2) A method to speed up the training of heterogeneous graph convolutional neural networks is proposed. The training process of the heterogeneous network graph convolutional neural networks is processed by a sampling method. By reducing the receptive field and the training range, the entire recommendation process is accelerated.
- (3) We conduct a large number of comparative experiments on three large-scale real data sets. In the experiments, the recommended accuracy MAE is increased by approximately 3%, the RMSE is increased by approximately 8%, and the running time is reduced significantly.

The organizational structure of the remainder of this paper is as follows: Section 2 presents the related research work, which introduces the related algorithms and technologies recommended by heterogeneous information networks, such as graph convolutional neural networks and matrix decomposition. Section 3 presents the deep learning method based on a heterogeneous information network. Section 4 is the deep learning algorithm based on a heterogeneous information network. Finally, a large number of experiments are conducted on real data sets to verify the effectiveness of the algorithm. Section 5 is the summary and future prospects section, which mainly summarizes the work

and contributions made in this paper and plans the next research work to be conducted.

## 2. Related Work

*2.1. Graph Convolutional Neural Network.* Since 2012, in-depth learning has achieved great success in the fields of computer vision and natural language processing. Compared with traditional methods, in-depth learning method can learn more efficient features and patterns. For example, given a graph for classification, traditional methods need to manually extract some features, such as texture, color, or some more advanced features. Then these features are put into classifiers such as the random forest, and output labels are given to indicate which category these features belongs to. In-depth learning inputs a map and directly outputs a label through a neural network. Feature extraction and classification are solved in one step, avoiding manual feature extraction or manual rules. Deep learning is an end-to-end learning process that automatically extracts features directly from the original data.

Although the effect of the convolutional neural network is very good, it is still limited to Euclidean domain data. The most prominent feature of Euclidean data is that there is a regular spatial structure. For example, pictures are regular square grids and speech is a regular one-dimensional sequence. However, these structures can be expressed by one-dimensional and two-dimensional matrices, so convolutional neural networks are very efficient in processing. However, there are many data in real life that do not have regular spatial structures and are called non-Euclidean data. For example, graph networks abstracted from social networks have different connections for each node in these graph structures. Some nodes have three connections and some nodes have two connections, which are irregular data structures. To solve the above problems, Kipf and Welling proposed graph convolutional neural network method to deeply learn graph data [10]. Compared with traditional methods, graph convolutional neural networks can learn more efficient features and patterns. However, graph convolutional neural networks are mostly used to deal with homogeneous networks. There is an urgent need for a recommendation algorithm based on heterogeneous information networks using deep learning techniques. To solve the problem that the neighboring nodes in a graph are not fixed, the graph convolutional neural network method finds the learnable convolutional check graph data suitable for the graph for in-depth learning.

The process of a graph convolution algorithm is as follows: (1) *Send*. Each transmitting node transforms its own characteristic information to its neighboring nodes. This step extracts and transforms the characteristic information of nodes. (2) *Polymerization*. Each node aggregates the characteristic information of neighboring nodes. This step fuses the local structural information of nodes. The corresponding graph transfers the information of neighboring nodes to nodes so that each node contains the information of its neighboring nodes and its own information. (3) *Non-linear Transformation*. After aggregating the previous

information, a nonlinear transformation is applied to increase the expression ability of the model. These steps are iterated continuously until a convergent output result is obtained.

*2.2. Matrix Factorization.* The score prediction of a recommendation algorithm can be regarded as matrix completion. Matrix completion is the task of a recommendation algorithm, and matrix decomposition is the means to achieve its goal. Therefore, matrix factorization is used to better complete the matrix completion task. The reason matrix decomposition can be used to complete matrix completion is based on the assumption that the matrix is of low rank. That is, there will always be similar people or things in the world. For example, birds of feather flock together and people are divided into groups. Then, the matrix can be restored by multiplying two small matrices. The matrix factorization method is first developed using a model-based recommendation algorithm. There are many examples of successful recommendation algorithms based on matrix factorization, such as recommendation for large groups [11], recommendation based on user contextual information [12], and recommendation for groups of items [13]. The idea of matrix decomposition is to decompose the matrix into more representative eigenvectors similar to the prime factor decomposition of integers, thus facilitating research. There are many matrix factorization methods. In linear algebra, there are two common methods: eigendecomposition and singular value decomposition. There are also implicit semantic model methods [14], Funk-SVD [15, 16], SVD++ [17], and other methods used in recommendation systems.

*2.2.1. Characteristic Decomposition.* The application scope of eigendecomposition is a square matrix, and only diagonalizable matrices can be decomposed into eigenvalues. If a square matrix  $A$  is similar to a diagonal matrix, that is, if there is a reversible matrix  $P$  such that  $P^{-1}AP$  is a diagonal matrix, it is called diagonalizable. The  $N$ -dimensional nonzero vector  $v$  is the eigenvector of  $N \times N$  matrix  $A$  if and only if  $Av = \lambda v$ , where  $\lambda$  is the corresponding eigenvalue and  $v$  is the corresponding eigenvector and is the eigenvalue. Matrix  $A$  can be decomposed into

$$A = Q \Lambda Q^{-1}. \quad (1)$$

Any  $N \times N$  real symmetric matrix has  $N$  linearly independent eigenvectors. Moreover, these eigenvectors can be orthogonally unitized to obtain a set of orthogonal vectors with module 1. Therefore, symmetric matrix  $A$  can be decomposed into

$$A = Q \hat{\Lambda} Q^T, \quad (2)$$

where  $Q$  is an orthogonal matrix and  $A$  is a real diagonal matrix. However, the common matrices in recommendation systems are not square matrices, so the feature decomposition method is not applicable.

### 2.2.2. Singular Value Decomposition (SVD).

Eigendecomposition is a matrix product that decomposes a square matrix  $A$  into an eigenvector matrix, an eigenvalue diagonal matrix, and the inverse of the eigenvector matrix. Singular value decomposition is a method that can decompose  $M \times N$  matrices. It decomposes the matrix  $M$  into

$$M = U \Sigma V^T. \quad (3)$$

Assuming that  $M$  is an  $M \times N$  matrix, then  $U$  is an  $M \times M$  square matrix (where the vectors are orthogonal, and the vectors in  $U$  are called left singular vectors),  $\Sigma$  is an  $M \times N$  real diagonal matrix (the elements other than diagonals are 0, and the elements on diagonals are called singular values), and  $V^T$  is an  $N \times N$  matrix (where the vectors are also orthogonal, and the vectors in  $V$  are called right singular vectors). Using singular value decomposition, storage space can be used to represent the matrix. However, in actual scenarios, matrices are large and sparse. Singular value decomposition is effective only when a matrix is dense. Therefore, singular value decomposition cannot be used to solve specific recommendation problems.

**2.2.3. Latent Factor Model (LFM).** The main idea of the Latent Factor Model (LFM) or “implicit semantic model” is to decompose the original scoring matrix  $M$  ( $mn$ ) into two matrices  $P$  ( $mk$ ) and  $Q$  ( $kn$ ). In addition, only the accuracy of decomposition results of the items with scores in the original scoring matrix is investigated and the mean squared error (MSE). That is, matrix  $M$  ( $mn$ ) is decomposed into  $P$  ( $mk$ ) and  $Q$  ( $kn$ ). At this time, for the positions with scores in the original matrix, the corresponding values in the decomposed matrix are

$$M_{UI} = \sum_{k=1}^k P_{u,k} Q_{k,i}. \quad (4)$$

This method is based on the Latent Factor Model (LFM) matrix factorization. The interpretation of the algorithm’s mean level links user interests with project features through implicit features. The LFM artificially sets the number of “hidden classes,” which means there is no need to consider the meaning of each hidden category; and then the matrix is restored by training the weights of the users and items for each hidden category. In a recommendation system, the matrix is sparse. That is, users have not interacted with most items, and the implicit feedback data have large area of 0s. Then, if you want to use the model to restore the value of position  $(I, J)$  (user  $I$ ’s score on item  $J$ ), and this position has not interacted and thus is represented as 0, then you cannot train to this point. It is also difficult to train an ideal result using SVD because the matrix is too sparse, and it is difficult to learn the eigenvalues and eigenvectors that can represent the matrix. After using the LFM, the matrix is abstracted into two matrices  $P$  and  $Q$ : one representing the user interest matrix and the other representing the item matrix. The two matrices are multiplied to restore the matrix. Then, as long as this user or project interacts with other projects or users, its

matrix can be trained, which means that the method can be trained without a dense matrix.

**2.2.4. Funk-SVD.** Similar to the LFM, the algorithm considers the bias of scoring criteria, so that the matrix factorized user item vector only needs to learn the differences between scoring and scoring criteria. Funk-SVD considers the bias between users and items. For example, some users tend to give very low scores or a movie is very poor and everyone has very low scores. By using bias to address special cases, it is easier for users and item matrices to learn expressions with better generalization performance. The scoring function is

$$\widehat{r}_{ui} = \mu + b_i + b_u + p_u q_i^T. \quad (5)$$

The objective function is

$$\arg \min \sum (r_{ui} - \widehat{r}_{ui})^2 + \lambda (\sum (b_u^2 + \|p_u\|^2) + \sum (b_i^2 + \|q_i\|^2)). \quad (6)$$

**2.2.5. SVD++.** In SVD++, implicit feedback is introduced on the basis of SVD; and the user’s historical browsing data, user’s historical scoring data, movie’s historical browsing data, movie’s historical scoring data, etc., are taken as new parameters. If a user scores a movie, then he has seen the movie, and such behavior actually contains certain information. That is, the scoring behavior reflects the user’s preferences, and such information can be reflected in the model in the form of implicit parameters.

$$\widehat{r}_{ui} = \mu + b_i + b_u + \left( p_u + \frac{\sum_{j \in I_u} y_j}{\sqrt{|I_u|}} \right) q_i^T, \quad (7)$$

where  $I_u$  is the set of all movies evaluated by the user; and  $y_j$  is the setting of personal preference bias reflected by the hidden “evaluated movie  $j$ ” and is a vector with the same dimension as  $p_u$ , not a scalar. The root sign of the contraction factor taking the set size is an empirical formula and has no theoretical basis.

## 3. Recommendation Algorithm Based on Heterogeneous Graph Convolution Neural Network (HGCR)

Most of the existing recommendation algorithms based on heterogeneous information networks are based on data mining algorithms that manually formulate standards. When faced with different requirements, most of them will choose different metapaths to calculate according to the situation. First, the manual selection of metapaths is mostly based on subjective preferences, and then the performance of the selected metapaths is verified through experiments. Second, the selection of metapaths consumes considerable time and even requires different weights for different metapaths. If the selected metapath is unreasonable, the correct representation method of the heterogeneous

information network will not be obtained, and then the correct recommendation results will not be obtained. The following is a detailed explanation of the existing algorithm problems and the advantages of the algorithm proposed in this paper.

### 3.1. Basic Concepts and Problem Definitions

*Definition 1. Heterogeneous Information Network.* A model is given to represent the heterogeneous information network, node set, and connection set. Generally, a heterogeneous information network includes a mapping relation for nodes and a mapping relation for edges that represent the defined entity and relationship types, respectively. Among the types of networks, a heterogeneous information network is a special kind of information network that includes many types of entities and many types of relationships.

*Definition 2. Information Transfer Framework.* Since the model calculation is applied to the local neighbors of the graph, the model can be understood as a simple information transfer model, namely,

$$h_i^{l+1} = \sigma \left( \sum_{m \in M_i} g_m(h_i^l, h_j^l) \right). \quad (8)$$

Among the components of the model, the hidden state of nodes in the neural network at the first layer is the dimension represented by this layer. Information passed in this form is integrated and passed by activating functions. A function represents a collection of information passed to a node and should generally be sensitive to the incoming side information. Usually, a neural network function is used to represent the error, or a simple linear transformation matrix with weight  $W$  is used to represent the error [10].

$$g_m(h_i, h_j) = Wh_j. \quad (9)$$

The existing recommendation algorithms based on heterogeneous information networks also have the following problems: (1) The recommendation process relies too much on the existing concepts and cannot use deep learning technology to automatically extract features, which limits the existing features and patterns. (2) The scoring process for users does not consider the differences of users and items, resulting in the inability to effectively collect the characteristics of users. Based on the definition of a heterogeneous information network, the definition of recommendation based on a heterogeneous information network is given.

*Definition 3. Recommendation Based on a Heterogeneous Information Network.* In a recommendation algorithm network, multiple types of data can be modelled using a heterogeneous information network. In recommendations based on heterogeneous information networks, there are two main items that deserve attention, namely, users and projects. The sum is the set of users and items, and the triple represents the user's score on the item and represents the set of scores. Given a heterogeneous information network, the

essence of recommendation is to predict a user's score for an item.

As seen from Definition 3, there are many elements in heterogeneous information networks; however, the recommendation of heterogeneous information networks mainly focuses on users and projects. Information in other heterogeneous information networks is used to assist users and projects in recommending. Weights will be included on the edges of the connections of user projects to represent users' scores on the projects, which is also a reference for the final recommendation. The relevant symbols used in this chapter and their explanations are shown in Table 1.

*Problem Definition.* Given a heterogeneous information network  $G = \{V, \varepsilon, R\}$  satisfying Definition 3, the score of the edge from the user node to the project node is predicted by using the graph convolution neural network method.

*3.2. Overview of Algorithms.* The graph convolution neural network method usually obtains feature vectors by normalizing and accumulating the transformations of adjacent nodes. Different from the conventional graph convolution neural network, this section introduces the transformation of specific relationships; that is, information transmission depends on the types of edges. To ensure that the representation of the nodes at the level can be obtained by the corresponding representation at the level, a single self-connection of a special relationship type is added to each node in the data. In addition, more flexible functions, such as multilayer neural networks (at the expense of computational efficiency), can also be selected instead of simple linear message conversion. Updating the neural network layer includes the parallel computation of each node in the graph. In practical applications, sparse matrix multiplication can be effectively realized to avoid the explicit summation of neighbors. Multiple layers can be stacked to implement dependencies across several relational steps. According to this, this paper proposes a heterogeneous graph convolution neural network algorithm for recommendation (HGCR) to better solve the problems defined in Section 3.1.

The input of the HGCR algorithm is a heterogeneous information network, as shown in Definition 1, which includes the evaluation values of some users of the project. The algorithm process mainly consists of two stages, namely, the learning stage of heterogeneous information network nodes by the graph convolution neural network method and the matrix decomposition stage. This paper mainly considers the advantages of the graph convolution neural network algorithm in learning graph structures and the advantages of the traditional matrix decomposition method in using scoring information and combines the two to recommend heterogeneous information networks. As shown in Figure 1, the input of the HGCR algorithm is a heterogeneous information network diagram. The diagram contains different types of nodes and the user's rating information for the movie. First, this heterogeneous information network graph is input into the heterogeneous graph convolution network model proposed in this paper. This model has the ability to

TABLE 1: Related notations and explanations.

Symbols	Explanation
$G$	Heterogeneous information network
$V$	Node set
$\varepsilon$	Edge set
$A$	Collection of categories for nodes
$R$	Collection of categories for relationships
$U$	User set
$I$	Project set
$\widehat{r}_{u,i}$	Estimated user rating of the project
$e_v$	Vector representation of node $v$
$N_u$	Neighbors of node $u$
$e_u^{(U)}, e_i^{(I)}$	Final vector representation of user $u$ and item $i$
$d$	Dimensions of heterogeneous information network representation
$D$	Dimension of hidden variables
$x_u, y_i$	Hidden variables for user $u$ and item $i$
$\alpha, \beta$	Parameters for combining heterogeneous information networks and matrix factorization

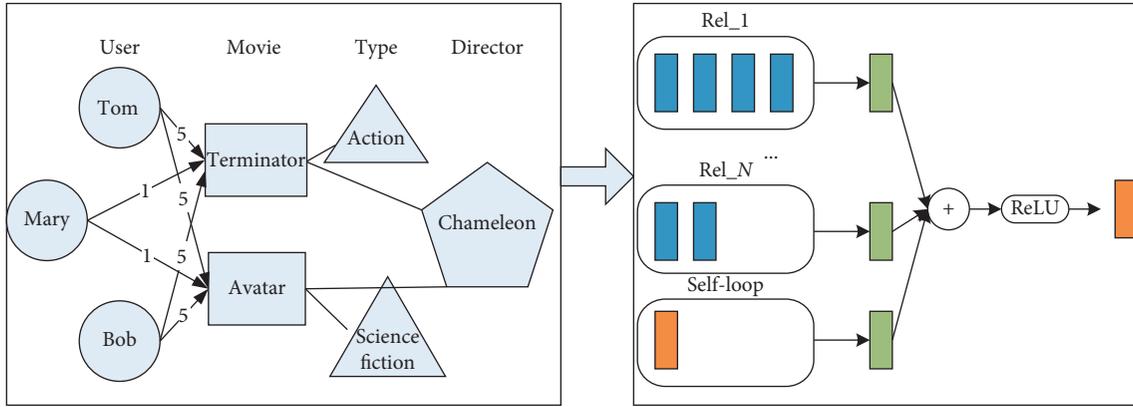


FIGURE 1: Example of the process of HGCR training.

learn that nodes are affected by different relationship types and output node representations that can reflect the heterogeneous information network structure, thus obtaining the representation of user nodes and the representation of movies. Then, the scoring matrix is constructed according to the scoring information contained in the heterogeneous information network. The matrix decomposition method decomposes a matrix into the product form of two matrices, namely, the matrix representing the user and the matrix representing the movie. The representation of the user and the representation of the movie obtained in the first step are combined with the hidden variables in matrix decomposition into the scoring matrix. Finally, the final matrix decomposition result is the recommended result to be obtained in this paper, and the score of each unknown position in the matrix is obtained.

**3.3. Pseudocode Description.** This section will describe the basic process of the heterogeneous information network recommendation algorithm based on a graph convolution neural network. The main pseudo code of the recommended algorithm is shown in Algorithm 1. The algorithm first calculates the node representation of the data on the graph, then updates the nodes by integrating the information from

different kinds of relationships, and combines the node representation with the matrix decomposition results to obtain a new representation of the nodes. The iteration is continued until convergence is obtained, and the corresponding score on the edge of each user and project is output.

When the algorithm is executed, the different relationships are first disassembled, and the node information of adjacent nodes is transmitted to the neighboring nodes according to the different relationships. The main purpose is to make better use of the spatial connection between users and items by using the characteristics of a graph convolution neural network that can flexibly combine the information of neighboring nodes. Combined with the continuous iteration of each neighbor's information, the processed information is aggregated and then subjected to nonlinear transformation to increase the expression ability of the model until convergence is obtained to obtain the node representation (lines 1–5). Then, the corresponding parameters of matrix factorization are initialized using the standard normal distribution (row 7), one triple is randomly selected for each cycle, and the new parameters are calculated via matrix factorization (rows 8–14). The matrix decomposition results are combined using the method obtained by the graph convolution neural network. We bring the combined results into

```

Input: heterogeneous information network, evaluation matrix  $R$ ;
Learning rate adjustment parameter regular parameter  $G = \{V, E, R\}; \eta; \alpha, \beta; \lambda$ 
Output: Hidden Factors for Users and Entities  $x, y$ 
(1) for  $r = 1$  to  $R$  do
(2)   for  $j \in N$  to  $N$  do
(3)     Use Equation (8) to obtain the representation of neighbor nodes
(4)   end for
(5) end for
(6) initialize  $x, y$ 
(7) initialize with standard normal distribution  $\gamma^{(U)}, \gamma^{(I)}, \theta^{(U)} \theta^{(I)}$ ;
(8) while not convergent do
(9)   Randomly select a triple  $(u, i, r_{u,i}) \in R$ 
(10)  update  $x_u, y_i$  by MF;
(11)  for  $l = 1$  to  $|P^{(U)}|$  do
(12)    calculation
(13)    Update  $\theta_u^{(U)}$ 
(14)  end for
(15)  Update  $\gamma_u^{(U)}$ 
(16)  for  $l = 1$  to  $|P^{(I)}|$  do
(17)    calculation
(18)    Update  $\theta_i^{(I)}$ 
(19)  end for
(20)  update  $\gamma_i^{(I)}$ 
(21) end while
(22) return  $x, y, \gamma^{(U)}, \gamma^{(I)}, \theta^{(U)}, \theta^{(I)}$ 

```

ALGORITHM 1: HGCR algorithm.

the formula to update  $\gamma_i^{(I)}, \gamma_u^{(U)}, \theta_i^{(I)}$ , and  $\theta_u^{(U)}$  (lines 16–20) and enter the updated results into Step 6 again. The matrix factorization process is continuously updated through iteration until the final output  $x, y, \gamma_i^{(I)}, \gamma_u^{(U)}, \theta_i^{(I)}$ , and  $\theta_u^{(U)}$  is obtained through convergence.

#### 4. Algorithm Details

This section is divided into four parts, describing the specific process of the algorithm from the aspects of the node representation process, objective function and model learning, and finally analyzing the complexity of the algorithm.

**4.1. Node Representation Process.** Because the information transfer framework method is very effective in representing structural information, it is necessary to define a similar transfer model to calculate the node representation for heterogeneous information networks. This paper proposes an information transfer method based on the following formula:

$$h_i^{l+1} = \sigma \left( \sum_{r \in R} \sum_{j \in N_r^i} \frac{1}{c_{i,r}} W_r^{(l)} h_j^{(l)} + W_0^{(l)} h_i^{(l)} \right). \quad (10)$$

$N_r^i \in Ric_{i,r}, c_{i,r} = |N_r^i|$ . The representation that represents the point under the relation is a constant specifically for a certain path, which can be learned or predefined.

Clearly, equation (10) collects the feature transformed vectors of all neighbors through regular addition.

Different from the conventional graph convolution neural network, this method uses the relationship information and directional information on the connection edges to collect the relationship information of neighboring nodes and other information, making the representation result more accurate. To make the representation of a node in layer  $L + 1$  also be affected by its representation in layer  $L$ , the transmission of the node's own information is increased; that is, the node information of each layer includes the neighboring node information of the upper layer and its own information.

A convolution network layer continuously updates the information of nodes by calculating equation (10) in parallel. In practice, equation (10) can be effectively applied in sparse matrix multiplication to avoid the display summation of neighbors. The multilayered structure can correspond to the information transmission in pairs of multistep relationships. This graph coding model is called the heterogeneous convolution neural network (H-GCN). The calculation process using the H-GCN model is shown in Figure 2. The triangle square node  $C$  represents the node itself that needs to be updated, and the elements in the three rectangular boxes represent the transmission of three different relationships. Under different relationships, the information of  $C$ 's neighbor nodes is extracted, and the information of  $C$  itself is extracted simultaneously from the nodes with the same type as  $C$ . Finally, the information of each relationship is aggregated, and the updated nodes are obtained through the ReLU activation function. The update of each node is

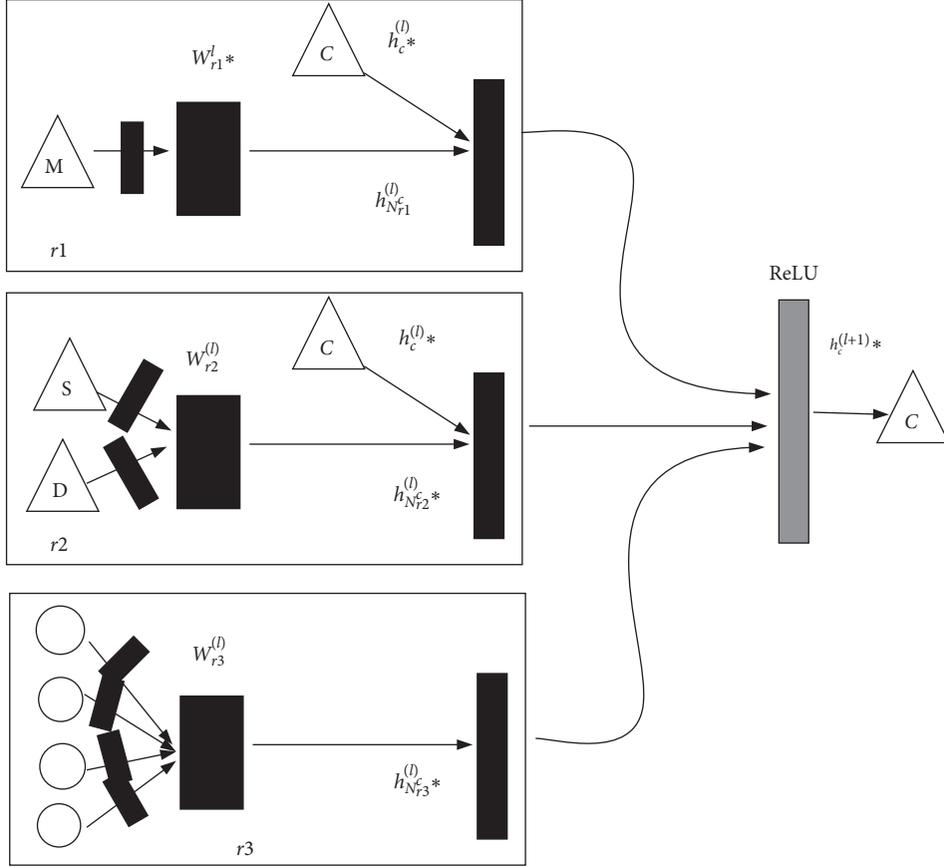


FIGURE 2: Process of H-GCN calculating.

conducted at the same time. A node is not only the node that receives information but also the node that transmits information to the outside in one iteration, transmitting the information to its neighbors and to itself. It can not only maintain the stability of its own information, but also update it according to the update of the neighboring node information to obtain accurate results.

When equation (10) is applied to multirelational data, a central problem is that the number of parameters in the graph increases rapidly with the number of relationships. In practice, when the model is large, data are easily overfit with a small number of processing relationships. Two intuitive strategies to solve this type of problem are to share parameters between the weight matrices and to retain the sparsity in the weight matrices to limit the total number of parameters.

$W_r^{(l)}$  corresponding to these two strategies, two methods are introduced to address the weight matrix on the layer, namely, basic decomposition and block diagonal decomposition. In basic decomposition, each matrix is represented as

$$W_r^{(l)} = \sum_{b=1}^B a_{rb}^{(l)} V_b^{(l)}. \quad (11)$$

$V_b^{(l)} \in R^{d^{(l+1)} \times d^l}$ ,  $a_{rb}^{(l)} \in R$ . That is, the linear combination of the basic transformation and the coefficients depends only on  $r$ . In block diagonal decomposition, each matrix is represented as the sum of multiple low-dimensional matrices:

$$W_r^{(l)} = \oplus_{b=1}^B Q_{br}^{(l)}. \quad (12)$$

$Q_{1r}^{(l)}, \dots, Q_{br}^{(l)} \in R^{(d^{(l+1)}/B) \times (d^{(l)}/B)}$ . This is the block diagonal matrix.

The basic function decomposition process of equation (11) can be regarded as parameter sharing for different relationships while equation (12) can be regarded as a sparse constraint for the parameters of each relationship. Block decomposition structure coding can group potential features into variable sets that are more closely coupled within groups than between groups.

Finally, the entry H-GCN model takes the following form: stacking layer  $L$  defined in equation (10), and the output of the previous layer is the input of the next layer. You can select the input of the first layer as the only input. If there are no other features, a vector is set for each node in the graph. For block representations, this vector is mapped to dense representations by a single linear transformation.

**4.2. Objective Function.** After the appearance of matrix factorization-based algorithms, many improved versions of these algorithms have emerged. BiasSVD is an improved and successful matrix factorization method. BiasSVD [18] assumes that the score includes some scoring factors unrelated to the user's items, and the user has some scoring factors unrelated to the items, which are called user bias items.

However, items also have some scoring factors unrelated to users, which are called item bias items. For example, a movie with its own quality problems cannot have a high score. Items with such poor attributes will directly lead to low user scores due to this factor, which is unrelated to users. This paper uses this algorithm to fully consider the influence of

irrelevant factors on the results to improve the recommendation accuracy.

Assuming that the average score of the scoring system is  $\mu$ , the user offset item of the  $i$ th user is  $b_i$ , and the item offset item of the  $j$ th item is  $b_j$ . The optimization objective function  $J(p, q)$  after adding the offset item is as follows:

$$\arg \min_{p, q, j} \sum_{i, j} (m_{ij} - \mu - b_i - b_j - q_j^T p_i)^2 + \lambda \left( \|p_i\|_2^2 + \|q_j\|_2^2 + \|b_i\|_2^2 + \|b_j\|_2^2 \right). \quad (13)$$

This optimization objective can also be solved by the gradient descent method. Different from other algorithms, there are two more paranoid terms  $b_i$  and  $b_j$  at this time, and the iteration formulas of  $b_i$  and  $b_j$  are similar to others; however, the gradient derivative of each step is slightly different, so it is not given here. In general, you can initially set the values for  $b_i$  and  $b_j$  to 0 and then participate in the iteration. The iterative method for  $b_i$  and  $b_j$  is presented as

$$\begin{aligned} b_i &= b_i + \alpha(m_{ij} - \mu - b_i - b_j - q_j^T p_i - \lambda b_i), \\ b_j &= b_j + \alpha(m_{ij} - \mu - b_i - b_j - q_j^T p_i - \lambda b_j). \end{aligned} \quad (14)$$

Finally,  $p$  and  $q$  can be obtained through iteration, which can be used for recommendation. BiasSVD adds some additional factors, and the influence of some factors can be ignored, which make the effect better.

Based on the classical matrix decomposition model, a score predictor is constructed, which decomposes the user-item score matrix into a user-specific matrix and an item-specific matrix. In the basic matrix factorization model, the score of user  $u$  on item  $i$  is simply defined as follows:

$$r_{u,i} = x_u^T \cdot y_i, \quad (15)$$

where  $x_u^T$  and  $y_i$  represent the potential factors corresponding to user  $u$  and item  $i$ . Since the representations of user  $u$  and

item  $i$  were also obtained in the previous section, they were further merged into the score predictor as follows:

$$r_{u,i} = x_u^T \cdot y_i + \alpha \cdot e_u^{(U)T} \cdot \gamma_i^{(I)} + \beta \cdot \gamma_u^{(U)T} \cdot e_i^{(I)}, \quad (16)$$

where  $e_u^{(U)T}$  and  $e_i^{(I)}$  are the outputs in the previous section,  $r_u^{(U)T}$  and  $r_i^{(I)}$  are the user-specific and project-specific potential factors, respectively, where  $e_u^{(U)T}$  and  $e_i^{(I)}$  are represented and paired with the nodes obtained by H-GCN, respectively,  $\alpha$  and  $\beta$  are the adjustment parameters integrating the three items. For equation (13), two points need to be noted. First,  $e_u^{(U)T}$  and  $e_i^{(I)}$  are the output of the function in equation (10). It is assumed that the derived representation function transformation is applicable to MF. Second, this article will not provide direct matches because the embedding method proposed in the previous section can indeed characterize the correlation between the same types of objects. This paper combines the  $e_u^{(U)T}$  and  $e_i^{(I)}$  of the new potential factors  $r_u^{(U)T}$  and  $r_i^{(I)}$  to relax the assumption that they must be in the same space, which increases the flexibility of the prediction model.

In this paper, the fusion function is incorporated into the matrix factorization framework to learn the parameters of the proposed model. The objective function can be expressed as

$$L = \sum_{\langle u, i, r_{u,i} \rangle \in R} (r_{u,i} - r_{u,i})^2 + \lambda \sum_u \left( \|x_u\|_2 + \|y_i\|_2 + \|\gamma_u^{(U)}\|_2 + \|\gamma_i^{(I)}\|_2 + \|\Theta^{(U)}\|_2 + \|\Theta^{(I)}\|_2 + b_i + b_j \right), \quad (17)$$

where the prediction score using the equation is  $r_{u,i} \lambda \Theta^{(U)} \Theta^{(I)}$ . Equation (15) is a regularization parameter, and the sum is a function of users and items.

**4.3. Model Learning.** This paper will use SGD to effectively optimize the final goal. The update of the original potential factor sum is the same as the matrix factorization algorithm in the formula. The parameters of the proposed model will be updated as follows:

$$\begin{aligned} \gamma_u^{(U)} &\leftarrow \gamma_u^{(U)} - \eta (b_i + b_j - \beta (r_{u,i} - \widehat{r}_{u,i}) e_i^{(I)} + \lambda_\gamma \gamma_u^{(U)}), \\ \gamma_i^{(I)} &\leftarrow \gamma_i^{(I)} - \eta (b_i + b_j - \alpha (r_{u,i} - \widehat{r}_{u,i}) e_u^{(U)} + \lambda_\gamma \gamma_i^{(I)}), \\ \Theta_{u,l}^{(U)} &\leftarrow \Theta_{u,l}^{(U)} - \eta \left( b_i + b_j - \alpha (r_{u,i} - \widehat{r}_{u,i}) \gamma_i^{(I)} \frac{\partial e_u^{(U)}}{\partial \Theta_{u,l}^{(U)}} + \lambda_\theta \Theta_{u,l}^{(U)} \right), \\ \Theta_{i,l}^{(I)} &\leftarrow \Theta_{i,l}^{(I)} - \eta \left( b_i + b_j - \beta (r_{u,i} - \widehat{r}_{u,i}) \gamma_u^{(U)} \frac{\partial e_i^{(I)}}{\partial \Theta_{i,l}^{(I)}} + \lambda_\theta \Theta_{i,l}^{(I)} \right). \end{aligned} \quad (18)$$

where the learning rate is  $\eta$ ,  $\lambda\Theta$  is the regularization of the parameters for  $\theta^{(U)}$  and  $\theta^{(I)}$ , and  $\lambda\gamma$  is the regularization of the parameters for  $\gamma^{(U)}$  and  $\gamma^{(I)}$ .

Figure 3 shows the algorithm flow that needs an iterative solution. First, according to the vector representation of the nodes obtained in the previous section, the required vector representation of relevant user items is obtained, and then the objective function solution is formed using the BiasSVD method. The yellow part is the input and output. The matrix at the time of the input contains some scoring information, and the final result includes all users' scores on all items.

In this paper, a sigmoid function is used for nonlinear transformation to solve the objective function, and the derivative calculation can be simplified by using the attributes of the sigmoid function. It is worth noting that the symbol  $\Theta$  represents all parameters in the fusion function, and the calculation will be different for different parameters. Next, the detailed derivation process is introduced for the personalized nonlinear fusion algorithm.

$$\frac{\partial e_i}{\partial \Theta_{i,l}} = \begin{cases} w_i^{(l)} \sigma(Z_s) \sigma(Z_f) (1 - \sigma(Z_s)) (1 - \sigma(Z_f)) e_i^{(l)}, & \Theta = M, \\ w_i^{(l)} \sigma(Z_s) \sigma(Z_f) (1 - \sigma(Z_s)) (1 - \sigma(Z_f)), & \Theta = b, \\ \sigma(Z_s) \sigma(Z_f) (1 - \sigma(Z_s)), & \Theta = w. \end{cases} \quad (19)$$

This paper calculates users and projects in the above way because it is relatively simple and omits the derivation of the linear fusion function.

**4.4. Complexity Analysis.** The HGCR consists of two main parts: (1) The embedded representation of heterogeneous information networks. The time complexity of a single graph convolution neural network algorithm is  $O(|\varepsilon|CHF)$ , in which  $|\varepsilon|$  is the number of edges,  $C$ ,  $H$ , and  $F$  represent the number of inputs, hidden layers, and output features, respectively. They are constant. Therefore, the complexity of the entire graph convolution neural network algorithm is linearly related to the number of edges in the graph. In this paper, the proposed algorithm H-GCN contains the relationship information on the edges and the time complexity becomes  $O(|\varepsilon|CHFR)$ . Since the number of  $R$  representing the relationship is constant, the time complexity indicates that the entire H-GCN algorithm remains linear with the number of edges in the graph. (2) *Matrix Decomposition.* For each triple  $\langle u, i, ru, i \rangle$ , the time complexity of updating parameters  $x$ ,  $y$ ,  $\gamma^{(U)}$ ,  $\gamma^{(I)}$  is  $O(D)$ , and  $D$  is the number of hidden variables. The update  $\theta^{(U)}$ ,  $\theta^{(I)}$  time complexity is  $O(|P|Dd)$ . Since  $|P|$  is small, it is defined by user.  $d$  is dimensions of heterogeneous information network.  $D$  and  $d$  do not exceed at most one thousand; this part is also very efficient on large data sets. The overall time complexity of the algorithm is still dominated by the previous part, which is linearly related to the number of edges.

## 5. Experimental Results and Analysis

**5.1. Data Set.** The laboratory environment is configured as follows: the operating system is Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-47-generic x86\_64). The GPU server is 2 GeForce RTX 2080 GPUs with 20 GB of memory. The python version is 3.5, the cuda version is 8.0.0, and the TensorFlow GPU version is 1.4.0. In addition, Networkx, c++11, NumPy, SciPy, etc., are also used.

To obtain more comprehensive heterogeneous information, this paper uses three real data sets for experiments. These data sets include the user movie data set, the Yelp Challenge data set, and the Douban Book data set, as shown in Table 2. The user movie data set includes 13,367 users and 12,677 movies, of which 106,8278 movies have a rating range of 1 to 5. The data set includes the social relations between users and the attribute information of users and movies. The Yelp Challenge data set contains the user scores of local merchants and the attribute information of users and merchants and ignores users and enterprises without relevant grades. The data set contains 198,397 scores ranging from 1 to 5: 162,39 users and 14,284 local enterprises. The Douban Book data set includes 13024 users and 22347 books, with 792026 scores ranging from 1 to 5, and includes the social relations between users and attribute information of users and books. These three data sets have different attributes. The Douban movie data set has the characteristics of dense scoring relationships and sparse social relationships, the Yelp data set has the characteristics of sparse scoring relationships but dense social relationships, and the Douban Book data set has the attributes of relatively medium-density scoring information and dense social relationships.

This paper uses several algorithms and matrix decomposition methods based on heterogeneous information networks to make comparison in the experiments in this chapter. PMF [19] is a matrix decomposition method that only uses the user item matrix as a recommendation. The SMF [20] algorithm adds social normalization terms to PMF to make users' potential factors closer to their friends' potential factors. CMF [21] is a collective matrix decomposition method that decomposes all relationships in heterogeneous information networks and shares potential factors of the same object type in different relationships. HeteMF [5] is a matrix factorization method with entity similarity regularization that also uses the relationships in heterogeneous information networks. SemRec [22] uses the sources. It is a collaborative filtering method based on weighted heterogeneous information networks and is built by connecting users to items with the same ratings. It can flexibly integrate heterogeneous information to make recommendations through weighted metapath and weight integration methods. I use the author's code to implement the model. DSR [23] is an MF-based recommendation method with double similarity regularization, which imposes constraints on users and commodities with high similarity and low similarity. HERec [24] is an algorithm that combines metapath representation with matrix

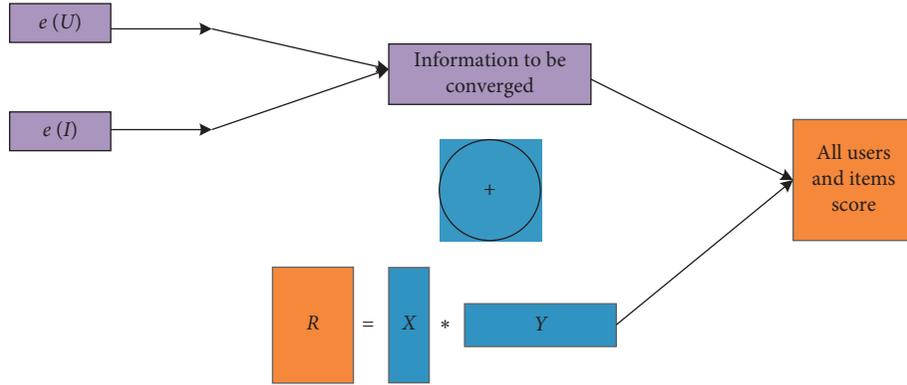


FIGURE 3: Process of information convergence.

TABLE 2: Statistics of data sets.

Data set	Relationship (A-B)	A	Number of B's	Number of (A-B)	Average degrees of A/B
Douban Movie	User-movie	13367	12677	1068278	79.9/84.3
	User-user	21192	2294	4085	1.7/1.8
	User-group	2440	2753	570047	42.7/207.1
Yelp	Movie-director	13337	2449	11276	1.1/4.6
	Movie-actor	10179	6311	33587	2.9/5.3
	Movie-type	11718	38	27668	2.2/728.1
	User-business	12678	14284	198397	12.2/13.9
	User-user	16239	10580	158590	15.0/15.0
	User-complement	10580	11	76875	5.3/6988
	Business-city	14411	47	14267	1.0/303.6
	Business-category	14267	511	40009	2.8/78.3
	User-book	14180	22347	792026	60.8/35.4
	User-user	13024	12748	169150	13.3/13.3
Douban Book	User-group	12748	2936	1189271	91.3/405.1
	Book-author	13024	10805	21907	1.0/78.3
	Book-publisher	21907	1815	21773	1.0/11.9
	Book-year	21773	64	21192	1.0/331.1

factorization, which can integrate users' structural information and attribute information.

## 5.2. Evaluation Indicators.

- (1) *Quality Evaluation of Recommended Results.* This experiment uses two evaluation indexes, the mean absolute error and the root mean square error, to test the quality of the evaluation results.

$$\text{RMSE} = \sqrt{\frac{\sum_{(u,i) \in R_{\text{test}}} (R_{u,i} - R_{u,i}^c)^2}{|R_{\text{test}}|}}, \quad (20)$$

$$\text{MAE} = \frac{\sum_{(u,i) \in R_{\text{test}}} |R_{u,i} - R_{u,i}^c|}{|R_{\text{test}}|}.$$

$R_{u,i}$ ,  $R_{u,i}^c$ , and  $R_{\text{test}}$  represent the score of the real user  $U$  on item  $I$ , the predicted score, and the entire test set, respectively. By definition, the smaller the MAE and RMSE are, the better the results.

- (2) *Cold Start Effect Evaluation.* For recommendation systems, cold start is an important issue. This index calculates the improvement of the effect of the basic algorithm by dividing the data into different degrees of sparsity and comparing the accuracy of this method with the accuracy of other algorithms.
- (3) *Evaluation of the Relationship Regularization Effect.* By comparing the recommended results before and after relationship regularization, as shown in Section 1, we can compare whether the relationship regularization strategy is effective.
- (4) *Iteration Effect.* We collect the scoring effect after every 20 iterations to observe the performance of each iteration and evaluate whether the algorithm converges quickly.

## 5.3. Experimental Results

*5.3.1. Evaluation of the Quality of the Recommended Results.* For each data set, this paper divides the whole scoring record into a training set and test set. For the Douban Movie and

Book data sets, this article sets four shares of training data to 80%, 60%, 40%, and 20%. For the Yelp data set, due to the sparsity of the data set [8], this paper sets four large shares of training data: 90%, 80%, 70%, and 60%. For each rate, this experiment randomly generates ten evaluation sets and takes the average of the results as the final performance.

As shown in Figures 4–6, the main findings of the experimental results are summarized as follows:

- (a) Among these benchmarks, the performance of heterogeneous information network-based methods (HeteMF, SemRec, CMF, and DSR) is better than that of traditional MF-based methods (PMF and SMF), which indicates that heterogeneous information is worth mining. It is worth noting that the CMF model is very effective in three benchmarks based on heterogeneous information networks. The intuitive explanation is that in the data set, most of the original features are attribute information of users or items, which may contain useful evidence to improve recommendation performance.
- (b) The proposed HGCR method is always superior to the baselines ranging from PMF to DSR. Compared with other methods based on heterogeneous information networks, HGCR adopts a more basic way to improve the recommendation system by using heterogeneous information networks, thus providing better information extraction (new representation of heterogeneous information network nodes) and utilization (extended MF model). In addition, the advantages of the proposed HGCR become more important when there are less training data. In particular, when using 20% of the Douban Book data set as training data, the improvement rate of HGCR compared with PMF is as high as 40%, which indicates that the performance has been significantly improved. As mentioned earlier, the data of the Yelp data set are very sparse. In this case, even with 60% of the data set as training data, the HGCR model is approximately 26% better than PMF. In contrast, at the same share of training data (i.e., 60%), the improvement rate of the RMSE for the proposed HGCR method relative to PMF is approximately 29%. These results show the effectiveness of the method, especially on sparse data sets. The results show that the heterogeneous information network embedding method is essential for recommendations based on heterogeneous information networks. The focus of this section is to learn the representation of users and projects while other types of objects are only used as bridges to build similar neighborhoods, thus contributing to the final recommendation results.

*5.3.2. Evaluation of the Cold Start Effect.* Heterogeneous information networks often improve the predictions affected by cold start because heterogeneous information networks contain rich text information, which can improve the effectiveness of prediction. Here, according to the number of

users' evaluations of the projects, users' evaluations are divided into three groups, namely, (0, 5), (5, 15), and (15,  $\infty$ ). That is, the sparsity of users' evaluations is distinguished. Those users with fewer than 5 evaluations, 5 to 15 evaluations, and more than 15 evaluations are used to perform experiments to verify the cold start effect and verify the cold start effect of this algorithm. The comparison algorithms include SMF, HeteMF, SemRec, DSR, and FMHIN. The results are compared with those of the algorithm for heterogeneous information and the algorithm for homogeneous networks.

Compared with the experimental results in Figures 7–12, the sparse algorithm provides a more obvious improvement compared with the basic recommendation algorithm. As the sparsity is improved, the results of this algorithm also improve correspondingly. As seen from the figure, this algorithm provides the most obvious improvement compared with the basic algorithm when evaluating less than 5 users' evaluations. Similarly, other heterogeneous information networks algorithms improve cold start performance by 40%. Compared with the recommendation algorithms based on heterogeneous information networks, the proposed algorithm can also make better use of heterogeneous information to achieve better results.

*5.3.3. Evaluation of the Regularization Effect.* Since the concept of relationship regularization is introduced in this paper, the number of relationship parameters cannot grow explosively and overfitting can also be prevented. The following two regularization formulas are used to verify this effect on the three data sets.

As seen from Figures 13 and 14, the regularization method in this paper does not obviously improve the results with a 0.1% increase per data set. The main reason is that the data set in this paper faces the recommendation problem in heterogeneous information networks. The number of relationships in most heterogeneous information networks is relatively small while the regularization method in this paper is mainly used to prevent overfitting when the number of relationships is too large; therefore, the proposed method has certain effects on several data sets in this paper, but it is not significant.

*5.3.4. Evaluation of the Iterative Effect.* In deep learning problems, the effect of iteration is often very critical. The following verifies the RMSEs on three data sets with different numbers of iterations.

Figures 15–17 show that as the number of iterations increase, the iteration effect can converge rapidly in the initial stage of the iteration. In Douban Book and Movie data sets, although the results did not converge completely until approximately 60 iterations, the results in the 20th iteration were optimal result, proving that conducting training using this algorithm can make the calculation results converge rapidly until the final results are obtained. Furthermore, due to the sparsity of the Yelp data set, the convergence speed of the proposed method is faster, and it is easier to obtain stable results. The algorithm in this paper can converge quickly

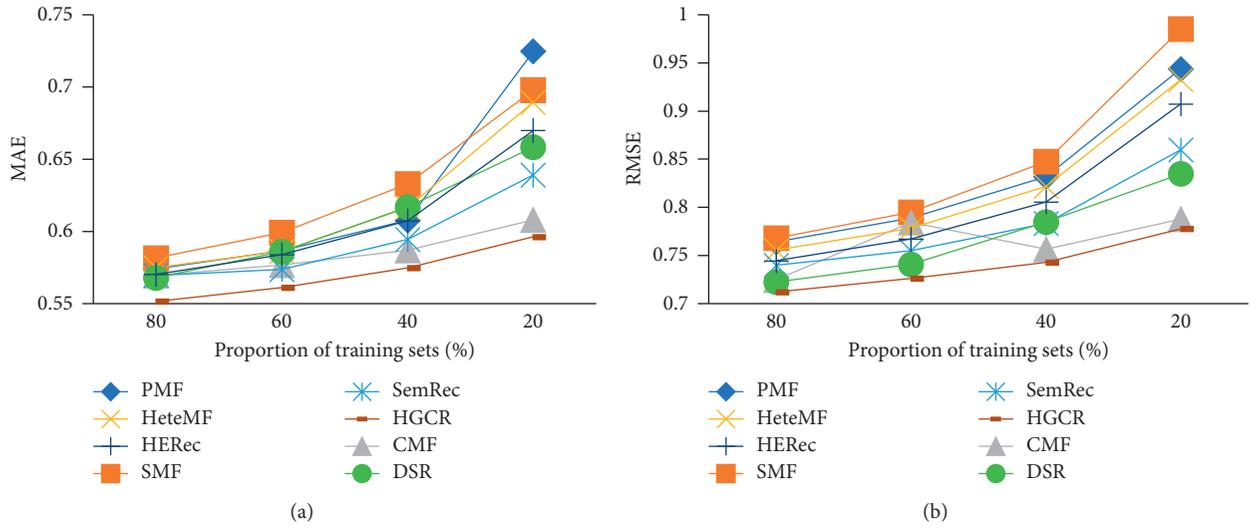


FIGURE 4: Performance of Douban Movie with different algorithms. (a) MAE vs. proportion of training sets. (b) RMSE vs. proportion of training sets.

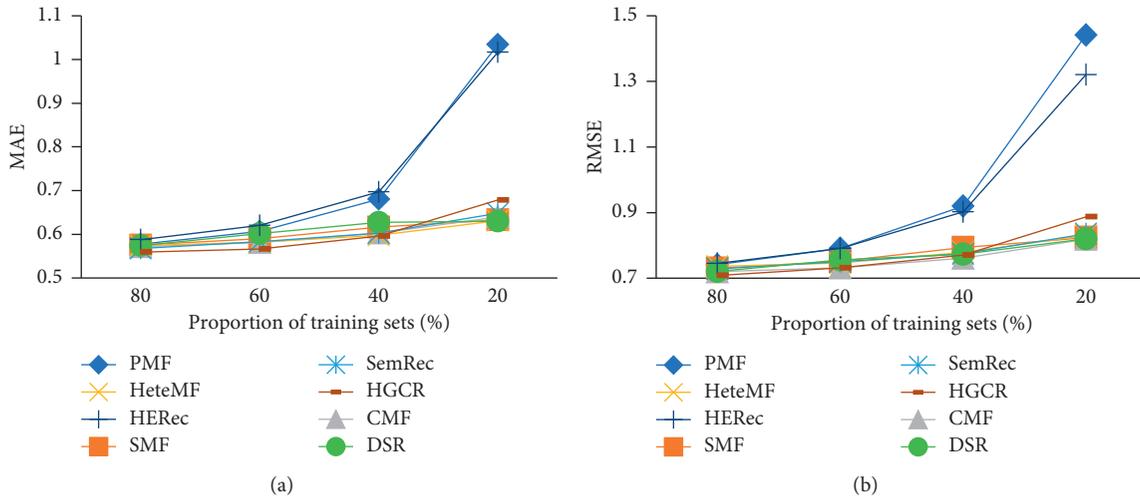


FIGURE 5: Performance of Douban Book with different algorithms. (a) MAE vs. proportion of training sets. (b) RMSE vs. proportion of training sets.

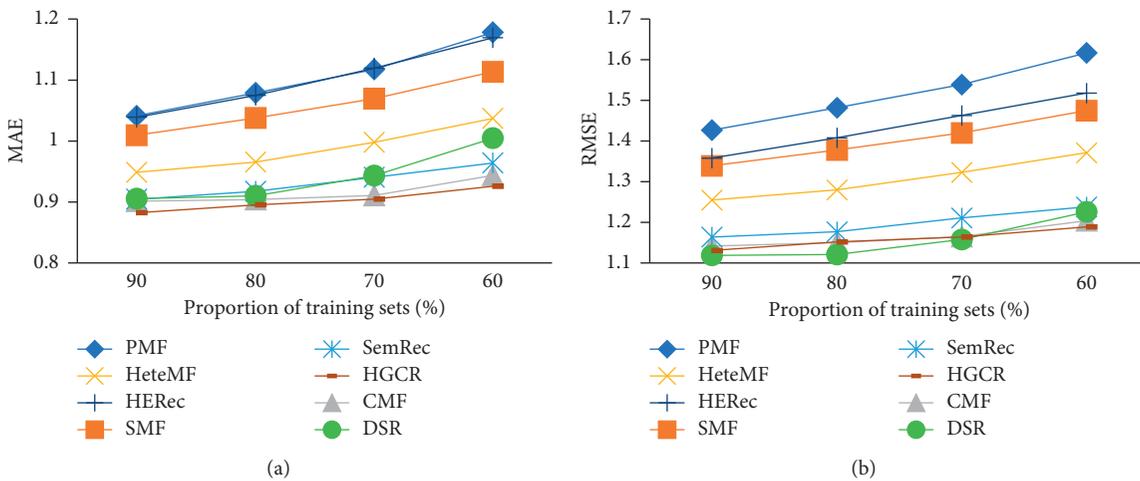


FIGURE 6: Performance of Yelp with different Algorithms. (a) MAE vs. proportion of training sets. (b) RMSE vs. proportion of training sets.

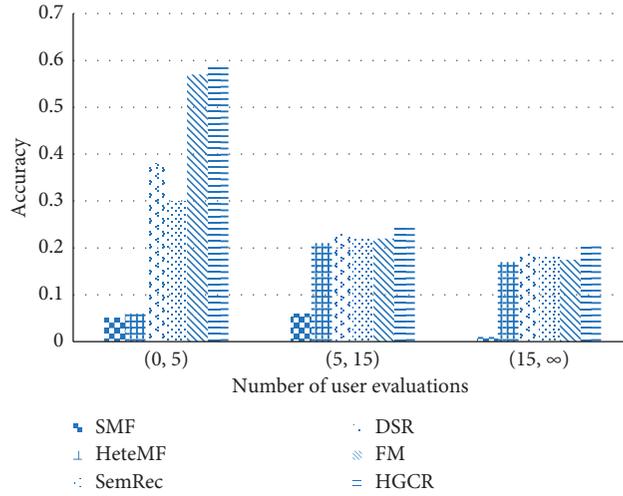


FIGURE 7: Performance of cold start on Douban Book.

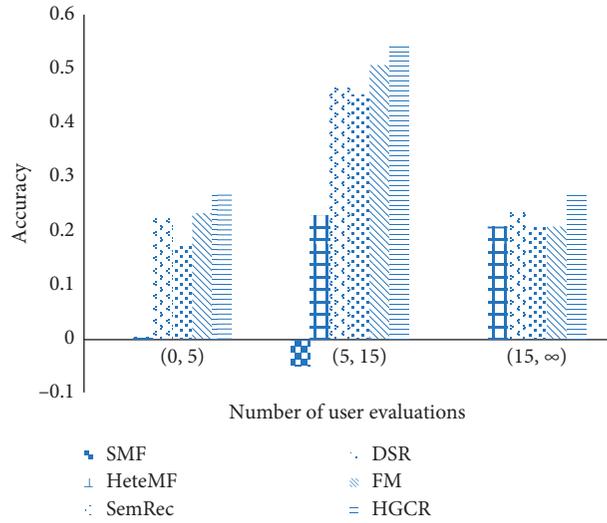


FIGURE 8: Performance of cold start on Douban Movie.

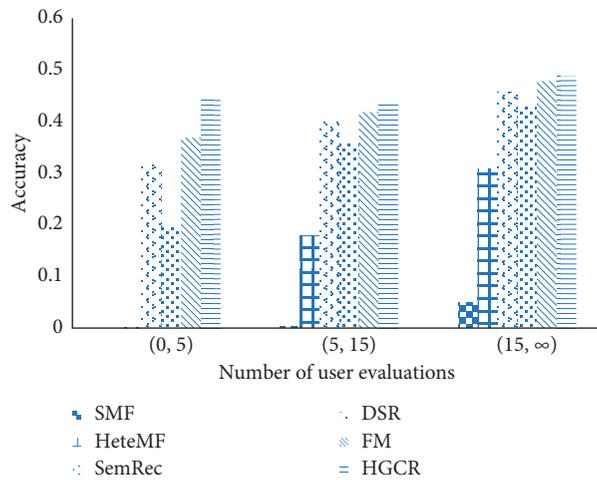


FIGURE 9: Performance of cold start on Yelp.

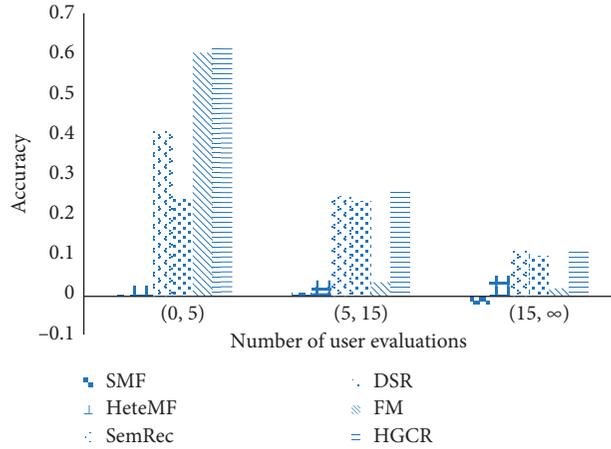


FIGURE 10: Performance of cold start on Douban Movie.

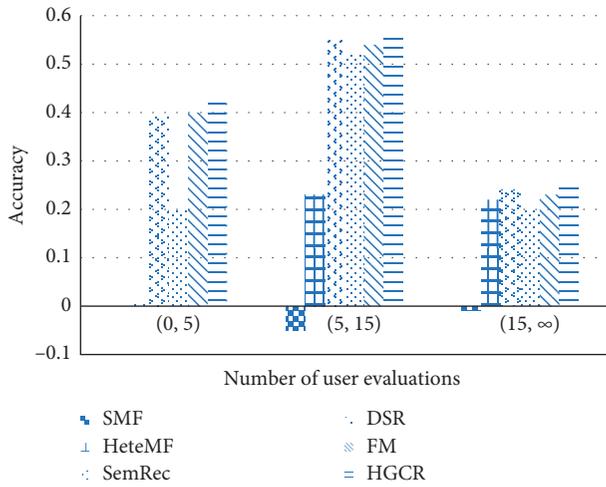


FIGURE 11: Performance of cold start on Douban Book.

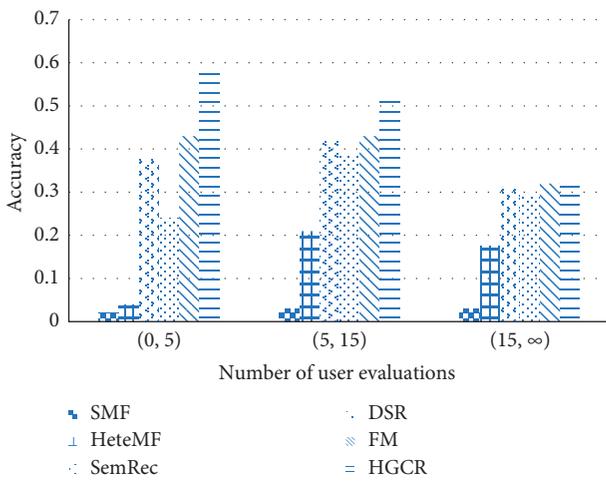


FIGURE 12: Performance of cold start on Yelp.

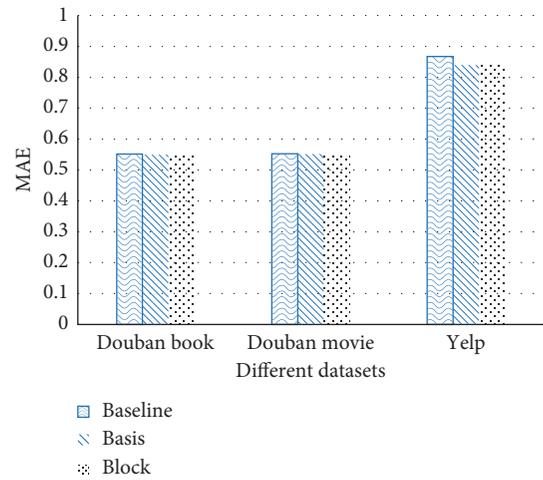


FIGURE 13: Normalization performance under MAE.

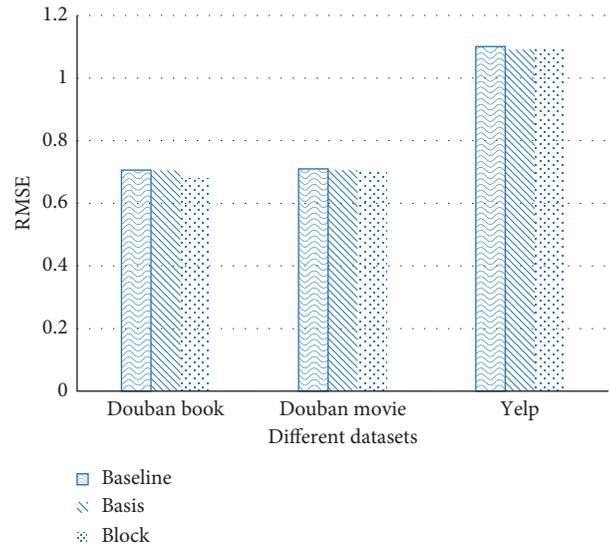


FIGURE 14: Normalization performance under RMSE.

within 60 iterations on the three data sets and has achieved good results, which proves that the algorithm in this paper is feasible.

5.3.5. *Evaluation of the Running Effect.* In order to verify the advantages of the algorithm in terms of time, the algorithm proposed in this paper and related algorithms NS and IS

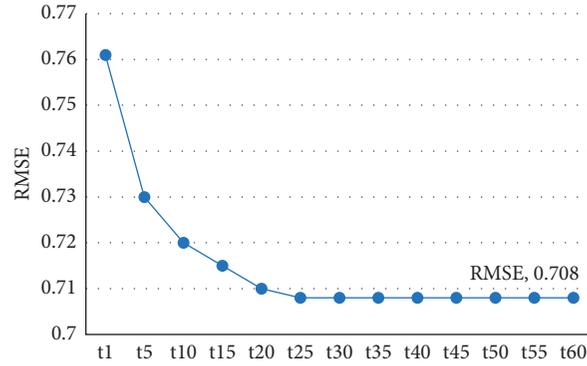


FIGURE 15: Performance of the number of iterations on Douban Movie.

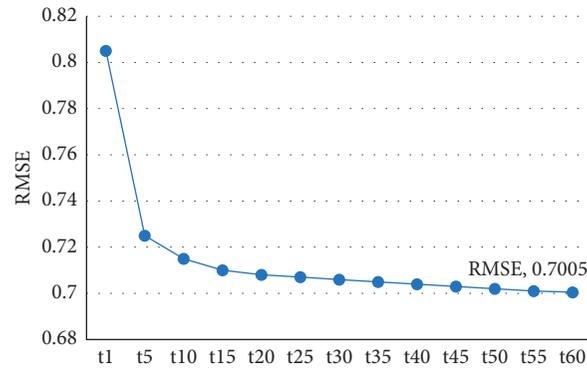


FIGURE 16: Performance of the number of iterations on Douban Book.

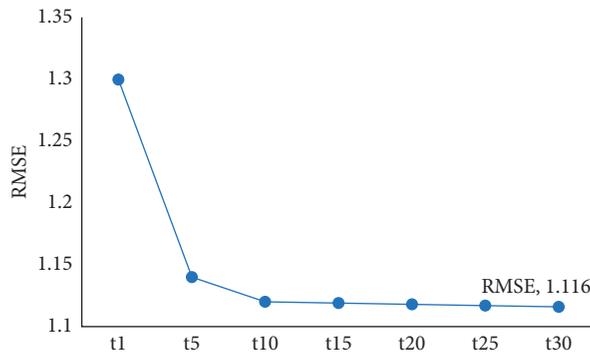


FIGURE 17: Performance of the number of iterations on Yelp.

TABLE 3: Evaluation of the running effect.

Algorithms	Douban Book			Douban Movie			Yelp		
	MAE	Rounds	Time (s)	MAE	Rounds	Time (s)	MAE	Rounds	Time (s)
NS	0.5735	100	445	0.5746	105	460	0.867	90	350
IS	0.5598	50	251	0.56	53	260	0.89	29	94
HGCR	0.5511	7.6	39	0.5522	9.6	59	0.84	4.8	28

were run in the data sets of Douban Book, Douban Movie, and Yelp to verify the effectiveness of the algorithm. NS represents the graph convolutional neural network training

algorithm [25] through Neighborhood Sampling (NS) method. IS is another training method [26]. IS is similar to NS; however, IS does not need to directly sample neighbors

for each node, but directly conducts subsampling for each layer.

As can be seen from Table 3, the proposed algorithm in this paper has fewer rounds than the existing sampling algorithm on different data sets, which can be reduced by up to 10 times and has shorter running times.

## 6. Conclusion

This section mainly introduces the process of making recommendations based on a heterogeneous graph convolution neural network. First, the research motivation of the algorithm is introduced, and the shortcomings of the existing recommendation algorithms based on heterogeneous information networks in accurately mining features and patterns and calculating results are analyzed. Then, a recommendation algorithm based on a graph convolution neural network for heterogeneous information networks is proposed. Second, the basic concepts of the HGCR algorithm are introduced, and the problems to be solved are defined through these concepts. Furthermore, the overall framework of the HGCR algorithm is introduced. First, the basic idea of the HGCR algorithm is summarized. Then, the detailed process of implementing the H-GCN algorithm is introduced, which is explained from the aspects of the transfer function and regularization function. Furthermore, the fusion process of the matrix factorization algorithm and H-GCN algorithm, the determination of the objective function, and the model learning of the solution algorithm are introduced. Finally, the complexity of the algorithm is analyzed. Furthermore, a large number of experiments are conducted on real data sets and synthetic data sets to verify the effectiveness and recommendation quality of the HGCR algorithm.

## Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study. The data used to support the findings of this study can be made available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the NSFC (Grant no. 61772124).

## References

- [1] A. K. Jacobs, E. M. Antman, G. Ellrodt et al., "Recommendation to develop strategies to increase the number of ST-segment-elevation myocardial infarction patients with timely access to primary percutaneous coronary intervention," *Circulation*, vol. 113, no. 17, pp. 2152–2163, 2006.
- [2] N. Lao and W. W. Cohen, "Relational retrieval using a combination of path-constrained random walks," *Machine Learning*, vol. 81, no. 1, pp. 53–67, 2010.
- [3] C. Shi, X. Kong, Y. Huang, P. S. Yu, and B. Wu, "HeteSim: a general framework for relevance measure in heterogeneous networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2479–2492, 2014.
- [4] W. Feng and J. Wang, "Incorporating heterogeneous information for personalized tag recommendation in social tagging systems," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1276–1284, ACM, London, UK, 2012.
- [5] X. Yu, X. Ren, Y. Sun et al., "Recommendation in heterogeneous information networks with implicit user feedback," in *Proceedings of the 7th ACM Conference on Recommender Systems*, pp. 347–350, ACM, New York, NY, USA, 2013.
- [6] X. Yu, X. Ren, Q. Gu et al., "Collaborative filtering with entity similarity regularization in heterogeneous information networks," in *Proceedings of the IJCAI HINA*, p. 27, Beijing, China, 2013.
- [7] X. Yu, X. Ren, Y. Sun et al., "Personalized entity recommendation: a heterogeneous information network approach," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, pp. 283–292, ACM, New York, NY, USA, 2014.
- [8] C. Luo, W. Pang, Z. Wang et al., "Hete-CF: social-based collaborative filtering recommendation using heterogeneous relations," in *Proceedings of the IEEE International Conference on Data Mining*, pp. 917–922, IEEE, Shenzhen, China, 2014.
- [9] C. Shi, Z. Zhang, P. Luo et al., "Semantic path based personalized recommendation on weighted heterogeneous information networks," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pp. 453–462, ACM, Melbourne, Australia, 2015.
- [10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, <https://arxiv.org/abs/1609.02907>.
- [11] Q. Dong, X. Zhou, L. Chen, G. Huang, and Y. Zhang, "Dynamic connection-based social group recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 453–467, 2020.
- [12] R. Lumbantoruan, X. Zhou, Y. Ren, and L. Chen, "I-CARS: an interactive context-aware recommender system," in *Proceedings of the 2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1240–1245, Beijing, China, 2019.
- [13] L. Zhang, R. Zhou, H. Jiang, H. Wang, and Y. Zhang, "Item group recommendation: a method based on game theory," in *Proceedings of the 26th International Conference on World Wide Web Companion*, pp. 1405–1411, Perth, Australia, 2017.
- [14] Y. Koren, "Factorization meets the neighborhood: a multi-faceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 426–434, ACM, Las Vegas, NV, USA, 2008.
- [15] G. H. Golub and C. Reinsch, *Singular Value Decomposition and Least Squares Solutions*, pp. 134–151, Springer, Berlin, Germany, 1971.
- [16] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proceedings of the KDD Cup and Workshop 2007*, pp. 5–8, San Jose, CA, USA, 2007.
- [17] V. Klema and A. Laub, "The singular value decomposition: its computation and some applications," *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, 1980.

- [18] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biological Cybernetics*, vol. 59, no. 4-5, pp. 291–294, 1988.
- [19] R. R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proceedings of the 2007 Conference Advances in Neural Information Processing Systems*, pp. 1256–1264, Cambridge, UK, 2007.
- [20] H. Ma, D. Zhou, C. Liu et al., "Recommender systems with social regularization," in *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, pp. 287–296, ACM, Hong Kong, 2011.
- [21] C. Lippert, S. H. Weber, Y. Huang et al., "Relation prediction in multi-relational domains using matrix factorization," in *Proceedings of the NIPS 2008 Workshop: Structured Input-Structured Output*, Vancouver, Canada, 2008.
- [22] C. Shi, Z. Zhang, Y. Ji, W. Wang, P. S. Yu, and Z. Shi, "SemRec: a personalized semantic recommendation method based on weighted heterogeneous information networks," *World Wide Web*, vol. 22, no. 1, pp. 153–184, 2019.
- [23] J. Zheng, J. Liu, C. Shi, F. Zhuang, J. Li, and B. Wu, "Recommendation in heterogeneous information network via dual similarity regularization," *International Journal of Data Science and Analytics*, vol. 3, no. 1, pp. 35–48, 2017.
- [24] C. Shi, B. Hu, W. X. Zhao et al., "Heterogeneous information network embedding for recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 2, pp. 357–370, 2018.
- [25] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017, <https://arxiv.org/abs/1706.02216>.
- [26] J. Chen, T. Ma, and C. Xiao, "FastGCN: fast learning with graph convolutional networks via importance sampling," 2018, <https://arxiv.org/abs/1801.10247>.