ICIC Express Letters
Volume **17**, Number **4**, April **2023**
ICIC International ©2023 ISSN 1881-803X
pp. **427–437**

LEARNING PEER RECOMMENDATION BASED ON HETEROGENEOUS INFORMATION NETWORK REPRESENTATION LEARNING AND DEEP LEARNING

Zhaoyu Shou*, Zhixuan Shi, Hui Wen, Jinghua Liu and Huibing Zhang

School of Information and Communication
Guilin University of Electronic Technology
No. 1, Jinji Road, Guilin 541004, P. R. China
{ 19022303055; 20022303188 }@mails.guet.edu.cn; { huiwen; zhanghuibing }@guet.edu.cn
*Corresponding author: guilinshou@guet.edu.cn

Received May 2022; accepted July 2022

Abstract. *This study proposes a learning peer recommendation algorithm based on heterogeneous information network representation learning and deep learning to better guide and motivate students to complete online learning courses and improve learning quality. Firstly, we integrate different objects (e.g., students, teachers, videos, exercises, and knowledge points) and various relations into a heterogeneous information network to retain semantic and structural information more comprehensively. Secondly, we propose a model that combines Multi-Layer Perceptron (MLP) with network representation learning to explore the correlations between students' preferences and auxiliary information in the network and solve the problems of the limited expressiveness of dot product and the weakness in capturing low-rank relations. Finally, the experimental results on the real datasets show the method is advanced and effective.*

Keywords: Online learning, Learning peer recommendation, Heterogeneous information networks, Network representation learning, Deep learning

1. **Introduction.** With the development of online education, problems such as insufficient course completion rates and poor learning effects have become increasingly prominent. This is due to the fact that teachers and students are separated in space and time. When encountering problems, students cannot communicate with each other in real time, which may result in feelings of loneliness and helplessness [1,2]. Learning peer recommendation can effectively alleviate the above phenomenon [3-5]. Potts et al. [6] developed the RiPPLE platform for helping target learners find suitable learning peers based on their learning logs. Hu et al. [2] presented the Learning Peer Recommendation (LPR) framework, which depicts the complex relationships among learners, learning content, and interaction behaviors with the help of a dynamic interaction tripartite graph and utilizes Convolutional Neural Network (CNN) to adjust the weight of interaction behaviors to make learning peer recommendations. However, the aforementioned studies extract only a portion of the interaction information in the online learning process for modeling purposes and fail to take account of the heterogeneity of the various types of objects and the complex interactions among these objects. Heterogeneous information networks offer an effective method to integrate heterogeneous information with diverse types and complex interactions, which opens up a lot of potential for data mining [7]. Xu et al. [8] applied a heterogeneous information network to making scholar-friend recommendations. However, this study relies on explicit path reachability and cannot fully mine the latent structural features of nodes in the network. Shi et al. [9] used a meta-path based random walk strategy for node embedding and integrated the embedding information into a

DOI: 10.24507/icicel.17.04.427

matrix factorization model. It is worth noting that the matrix factorization model simply uses the dot product and cannot learn the complex mapping mechanism. Karnyoto et al. [10] adopted a heterogeneous graph neural network to detect fake news related to COVID-19. Wang et al. [11] presented a disentangled heterogeneous graph attention network for top-N recommendation, which learned disentangled user/item representations from different aspects in a heterogeneous information network. Although these methods can learn more complex and accurate network representations and are suitable for large-scale application scenarios, they have high time and space complexity and require expensive hardware support. The structure of MLP is simple, and it has non-linearity and high-capacity characteristics that can learn better latent representation [12].

To overcome the above limitations, we propose a learning peer recommendation algorithm based on heterogeneous information network representation learning and deep learning (named as LPRRD). The main contributions of this study are as follows. First, we construct a heterogeneous information network that integrates different types of objects (e.g., students, teachers, videos, exercises, and knowledge points) and relations between objects into a unified framework and retains semantic and structural information. Second, we propose a model combining MLP with network representation learning to solve the problems of the limited expressiveness of dot product and the weakness in capturing low-rank relations, so as to mine the correlations between students' preferences and auxiliary information in the network.

The rest of this paper is organized as follows. Section 2 introduces the relevant definitions. Section 3 describes the proposed algorithm in detail. Section 4 reports experimental results. Section 5 concludes this paper and gives future research directions.

2. **Related Definitions.** This section describes the relevant definitions used in this study to illustrate the algorithm in this paper more clearly.

Definition 2.1. Heterogeneous information network. *A heterogeneous information network is defined as a directed graph $G = (V, E)$ with an object type mapping function ϕ: $V \to \mathcal{A}$ and a link type mapping function ψ: $E \to \mathcal{R}$, where the types of objects $|\mathcal{A}| > 1$ (or the types of relations $|\mathcal{R}| > 1$). each object $v \in V$ belongs to a particular object type $\phi(v) \in \mathcal{A}$ and each link $e \in E$ belongs to a particular relation type $\psi(e) \in \mathcal{R}$* [7].

Figure 1 shows the heterogeneous information network on an online learning platform, which contains five different types of objects (e.g., students, teachers, videos, exercises and knowledge points) and sixteen types of relations. R_i denotes a particular type of relation between two different types of objects (R_i^{-1} represents the inverse relation of R_i). The figure depicts different relations R_1, R_1^{-1}, R_2, R_2^{-1}, R_3, R_3^{-1}, R_4, R_4^{-1}, R_5, R_5^{-1}, R_6, R_6^{-1}, R_7, R_7^{-1}, R_8 and R_8^{-1}, which denote do, done, examine, examined-by, answer, answered-by, record, recorded-by, include, included-in, watch, watched-by, question, questioned-by, discuss and discussed-by, respectively.

Definition 2.2. Network schema. *The network schema $T_G = (\mathcal{A}, \mathcal{R})$ is the meta structure of a heterogeneous information network $G = (V, E)$, which consists of a set of object types $\mathcal{A} = \{A\}$ and a set of relation types $\mathcal{R} = \{R\}$* [7]. *Figure 2 illustrates the network schema of the heterogeneous information network on the online learning platform, which specifies the types of objects and relations.*

Definition 2.3. Meta-Path. *A meta-path $P = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \cdots \xrightarrow{R_l} A_{l+1}$ is a path defined based on a network schema $T_G = (\mathcal{A}, \mathcal{R})$* [7]. *$P$ describes a composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between object types $A_1, A_2, \ldots, A_{l+1}$. In addition, a concrete path following the object and relation requirement of meta-paths is called a path instance, and different meta-paths have different semantic meanings.*

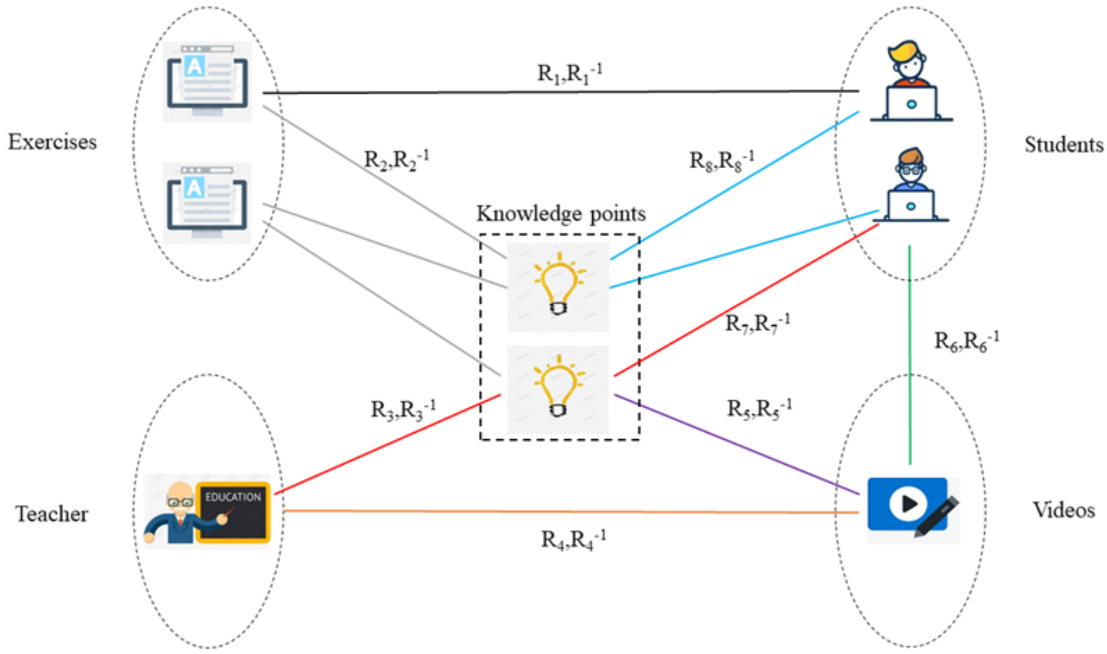


FIGURE 1. A heterogeneous information network on an online learning platform

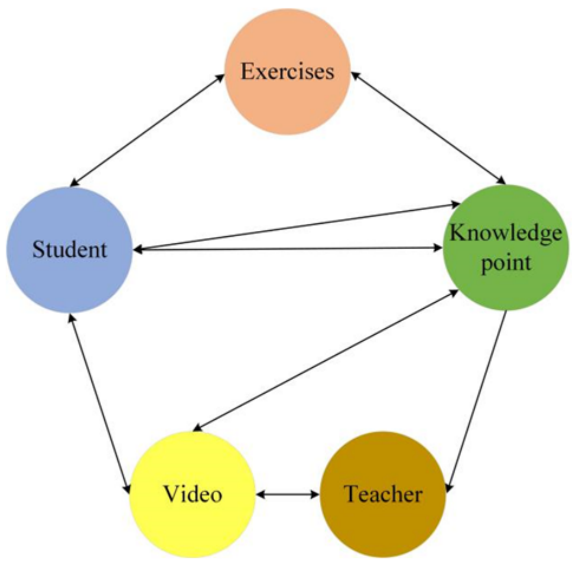


FIGURE 2. An example of network schema on an online learning platform

3. **Methodology.** On the basis of the above definitions, the proposed method LPRRD consists of the following three steps. First, we construct a heterogeneous information network on an online learning platform and use a heterogeneous information network representation learning method to learn student embeddings. Second, we integrate the node embeddings into representation learning and matching function learning framework for implicit feedback prediction. Third, we define the objective function for model learning, so that the LPRRD model can achieve the best performance and generate a learning peer recommendation list for each student.

3.1. **Heterogeneous information network representation learning.**

3.1.1. *Meta-path based random walk.* Given a heterogeneous information network $G = (V, E)$ and a meta-path $M_P = A_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{t-1}} A_t \xrightarrow{R_t} \cdots \xrightarrow{R_l} A_{l+1}$, in order to capture the complex semantic meanings of meta-paths in this network, we adopt a meta-path based random walk method to generate node sequences. The walking path is generated according to the following distribution:

$$P\left(n_{t+1}=x \mid n_t=u, M_P\right)=\begin{cases}\dfrac{1}{\left|\mathcal{N}^{A_{t+1}}(u)\right|}, & (u, x) \in \mathcal{E} \text{ and } \phi(x)=A_{t+1} \\ 0, & \text{otherwise}\end{cases} \tag{1}$$

where n_t is the t-th node in the walk, u has the type of A_t and $\mathcal{N}^{A_{t+1}}(u)$ is the first-order neighbor set for node u with the A_{t+1} type. Each walk will follow the pattern of a meta-path repetitively until it reaches the pre-defined length.

3.1.2. *Node type constraint.* Since the purpose of this study is to recommend learning peers, we only focus on meta-paths with student type as head and tail nodes. Once a node sequence has been generated, we remove the nodes with a type different from the student type. Taking the meta-path SES as an example, the semantic meaning is the students' test behavior. According to Formula (1), we can generate a sampled sequence "$s_1 \to e_1 \to s_2 \to e_2 \to s_3 \to e_1 \to s_4$", then remove the nodes with a type different from the student type, and finally obtain a homogeneous node sequence "$s_1 \to s_2 \to s_3 \to s_4$".

3.1.3. *Optimization objective.* We construct the neighborhood $\mathcal{N}_u$ for student u based on co-occurrence in fixed-length windows. Following node2vec [13], the following objective function can be optimized by learning the representations of student nodes:

$$\max\nolimits_f \sum_{u \in v} \log \Pr\left(\mathcal{N}_u \mid f(u)\right) \tag{2}$$

where f: $\mathcal{V} \to \mathbb{R}^d$ represents a function mapping each node to the d dimensional feature space, $\mathcal{N}_u \subset \mathcal{V}$ represents the neighborhood of node u.

3.1.4. *Node representation fusion.* Given a set of representations $\left\{\mathbf{e}_u^{(l)}\right\}_{l=1}^{|P|}$, where P represents the set of meta-paths, and $\mathbf{e}_u^{(l)}$ is the representation for the target student u over the l-th meta-path. In order to model each student's personalized preferences over each meta-path, we assign each student a weight vector on the meta-paths and use a personalized non-linear fusion function $g(\cdot)$ to enhance the fusion ability.

$$g\left(\left\{\mathbf{e}_u^{(l)}\right\}\right)=\sigma\left(\sum_{l=1}^{|P|} w_u^{(l)} \sigma\left(\mathbf{M}^{(l)} \mathbf{e}_u^{(l)}+\mathbf{b}^{(l)}\right)\right) \tag{3}$$

where $w_u^{(l)}$ is the preference weight of the target student u over the l-th meta-path, $\mathbf{M}^{(l)} \in \mathbb{R}^{D \times d}$ and $\mathbf{b}^{(l)} \in \mathbb{R}^{D \times d}$ are the transformation matrix and bias vector of the l-th meta-path, respectively. The node representation for candidate student v is calculated in the same manner.

$$\begin{cases}\mathbf{e}_u^{(U)} \leftarrow g\left(\left\{\mathbf{e}_u^{(l)}\right\}\right) \\ \mathbf{e}_v^{(V)} \leftarrow g\left(\left\{\mathbf{e}_v^{(l)}\right\}\right)\end{cases} \tag{4}$$

where $\mathbf{e}_u^{(U)}$ and $\mathbf{e}_v^{(V)}$ are the final representations for target student u and candidate students v, respectively.

3.2. **Implicit feedback prediction.** After obtaining final representations $\left\{\mathbf{e}_u^{(U)}\right\}_{u \in U}$ and $\left\{\mathbf{e}_v^{(V)}\right\}_{v \in V}$, we incorporate them into matrix factorization as below:

$$\widehat{r_{u,v}}=\mathbf{p}_u^T \mathbf{q}_v+\alpha \cdot \mathbf{e}_u^{(U)^T} \cdot \gamma_v^{(V)}+\beta \cdot \mathbf{e}_v^{(V)} \cdot \gamma_u^{(U)^T} \tag{5}$$

where $\mathbf{p}_u \in \mathbb{R}^{D\times d}$ and $\mathbf{q}_v \in \mathbb{R}^{D\times d}$ denote the latent factors corresponding to target student u and candidate students v, $\gamma_u^{(U)^T}$ and $\gamma_v^{(V)}$ are the target student-specific and candidate student-specific latent factors to pair with $\mathbf{e}_v^{(V)}$ and $\mathbf{e}_u^{(U)}$, respectively, and α and β are the tuning parameters to integrate the three terms.

Learning peer recommendation can be viewed as a matching problem that matches suitable candidate students for target students. The matrix factorization model uses the dot product to aggregate the latent factors in Formula (5), which leads to limited expressiveness. To solve the above problems, we introduce the representation learning and matching function learning model [12] to reconstruct Formula (5). The representation learning is to map target students and candidate students to the same representation space and learn the low-dimensional latent features of both target students and candidate students. If the similarity between students is higher in this space, it means that they match better. The matching function learning focuses on the matching between the target students and the candidate students, and the purpose is to map the interaction between students into the complex matching function.

3.2.1. *Representation learning.* Taking the student-student interaction matrix Y as input, the target student u is represented by the corresponding row $\mathbf{y}_{u^*}$ in Y, and the candidate student v is represented by the corresponding column $\mathbf{y}_{*_v}$ in Y. We adopt MLP to learn the latent feature representation for target students and candidate students. The representation learning for target students can be defined as

$$\begin{cases} \mathbf{a}_0 = \mathbf{W}_0^T \mathbf{y}_{u^*} \\ \mathbf{a}_1 = a\left(\mathbf{W}_1^T \mathbf{a}_0 + \mathbf{b}_1\right) \\ \vdots \\ \mathbf{p}_u = \mathbf{a}_x = a\left(\mathbf{W}_x^T \mathbf{a}_{x-1} + \mathbf{b}_x\right) \end{cases} \tag{6}$$

where $\mathbf{W}_x$, $\mathbf{b}_x$, and $\mathbf{a}_x$ denote the weight matrix, bias vector, and activation for the x-th layer's perceptron, respectively. In addition, the activation function $a(\cdot)$ used in this study is the ReLU function. The latent factors $\mathbf{q}_v$ for candidate student v is calculated in the same way. The matching function learning is defined as

$$\widehat{y_{uv}} = \sigma\left(\mathbf{W}_{out}^T\left(\mathbf{p}_u \odot \mathbf{q}_v\right)\right) \tag{7}$$

where $\mathbf{W}_{out}$ and $\sigma(\cdot)$ are the weight matrix and the sigmoid function, respectively.

3.2.2. *Matching function learning.* Since the initial representations for target students and candidate students are very sparse and have high dimensions, it is difficult for the model to directly learn the matching function. We adopt MLP to learn the matching function and take the interaction matrix Y as input. To calculate the matching score, we pass this joint representation into a fully connected layer that enables the model to assign different weights to the features, and the matching function learning component can be defined as

$$\begin{cases} \mathbf{p}_u = \mathbf{P}^T \mathbf{y}_{u^*} \\ \mathbf{q}_v = \mathbf{Q}^T \mathbf{y}_{*_v} \\ \mathbf{a}_0 = \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_v \end{bmatrix} \\ \mathbf{a}_1 = a\left(\mathbf{W}_1^T \mathbf{a}_0 + \mathbf{b}_1\right) \\ \vdots \\ \mathbf{a}_Y = a\left(\mathbf{W}_Y^T \mathbf{a}_{Y-1} + \mathbf{b}_Y\right) \\ \widehat{y_{uv}} = \sigma\left(\mathbf{W}_{out}^T \mathbf{a}_Y\right) \end{cases} \tag{8}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are the parameter matrices of the linear embedding layer. The latent factors $\mathbf{p}_u$ and $\mathbf{q}_v$ are aggregated by a simple concatenation operation. Finally, the matching score $\widehat{y_{uv}}$ is calculated using the MLP as the mapping function.

3.2.3. *Model fusion based on representation learning and matching function learning.* We consider concatenating the learned representations and feeding them back into a fully connected layer to fuse representation learning and matching function learning models. Assuming that the prediction vectors of the representation learning component and the matching function learning component are $\mathbf{a}_Y^{rl}$ and $\mathbf{a}_Y^{ml}$, respectively, then the output of the fusion model $\widehat{y_{uv}}_{fusion}$ can be defined as

$$\widehat{y_{uv}}_{fusion} = \sigma\left(\mathbf{W}_{out}^T \begin{bmatrix} \mathbf{a}_Y^{rl} \\ \mathbf{a}_Y^{ml} \end{bmatrix}\right) \tag{9}$$

3.2.4. *Objective function of model fusion.* Learning peer recommendation is an implicit feedback recommendation problem. The implicit feedback of the interaction matrix of target students and candidate students can be defined as

$$y_{uv} = \begin{cases} 1, & \text{if interaction } (u,v) \text{ is observed} \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Cross entropy can be used as the loss function, which can be expressed as (11):

$$\ell_{BCE} = -\sum_{(u,i)\in\mathcal{Y}^+\cup\mathcal{Y}^-} y_{uv}\log\widehat{y_{uv}}_{fusion} + (1-y_{uv})\log\left(1-\widehat{y_{uv}}_{fusion}\right) \tag{11}$$

where $\mathcal{Y}^+$ denotes all the observed interactions in $\mathbf{Y}$, and $\mathcal{Y}^-$ represents the sampled unobserved interactions.

The output of the standard matrix factorization model is replaced by the output of the model fusion based on representation learning and matching function learning. The predicted rating in the heterogeneous information network can be defined as

$$\widehat{r_{u,v}} = \widehat{y_{uv}}_{fusion} + \alpha\cdot\mathbf{e}_u^{(U)^T}\cdot\gamma_v^{(V)} + \beta\cdot\mathbf{e}_v^{(V)}\cdot\gamma_u^{(U)^T} \tag{12}$$

3.3. **Model learning.** In this study, we blend personalized non-linear fusion function $g(\cdot)$ into representation learning and matching function learning framework. The objective function of the whole model can be defined as

$$\begin{aligned} L = &\sum_{\langle u,v,r_{u,v}\rangle}\left(r_{u,v}-\widehat{r_{u,v}}\right)^2 \\ &+\lambda\sum_u\left(\|\mathbf{p}_u\|_2+\|\mathbf{q}_v\|_2+\left\|\gamma_u^{(U)}\right\|_2+\left\|\gamma_v^{(V)}\right\|_2+\left\|\Theta^{(U)}\right\|_2+\left\|\Theta^{(V)}\right\|_2\right) \end{aligned} \tag{13}$$

where λ is the regularization parameter, $\Theta^{(U)}$ and $\Theta^{(V)}$ are parameters of $g(\cdot)$ for the target students and the candidate students, respectively. In this study, we adopt SGD to efficiently optimize the final objective, and the parameters will be updated as follows:

$$\begin{cases} \Theta_{u,l}^{(U)} \leftarrow \Theta_{u,l}^{(U)} - \eta\cdot\left(-\alpha\left(r_{u,v}-\widehat{r_{u,v}}\right)\gamma_v^{(V)}\dfrac{\partial\mathbf{e}_u^{(U)}}{\partial\Theta_{u,l}^{(U)}}+\lambda_\Theta\Theta_{u,l}^{(U)}\right) \\ \Theta_{v,l}^{(V)} \leftarrow \Theta_{v,l}^{(V)} - \eta\cdot\left(-\beta\left(r_{u,v}-\widehat{r_{u,v}}\right)\gamma_u^{(U)}\dfrac{\partial\mathbf{e}_v^{(V)}}{\partial\Theta_{v,l}^{(V)}}+\lambda_\Theta\Theta_{v,l}^{(V)}\right) \\ \gamma_u^{(U)} \leftarrow \gamma_u^{(U)} - \eta\cdot\left(-\beta\left(r_{u,v}-\widehat{r_{u,v}}\right)\mathbf{e}_v^{(V)}+\lambda_\gamma\gamma_u^{(U)}\right) \\ \gamma_v^{(V)} \leftarrow \gamma_v^{(V)} - \eta\cdot\left(-\alpha\left(r_{u,v}-\widehat{r_{u,v}}\right)\mathbf{e}_u^{(U)}+\lambda_\gamma\gamma_v^{(V)}\right) \end{cases} \tag{14}$$

where η is the learning rate, λ_Θ is the regularization for parameters $\Theta^{(U)}$ and $\Theta^{(V)}$, and λ_γ is the regularization for parameters $\gamma_u^{(U)}$ and $\gamma_v^{(V)}$. In this study, we utilize the sigmoid function for non-linear transformation, and we can take advantage of the properties of sigmoid function for ease of derivative calculation. In addition, Θ denotes all the parameters in the fusion function $g(\cdot)$, and $\partial \mathbf{e}_v^{(V)} \big/ \partial \Theta_{v,l}^{(V)}$ can be defined as

$$\frac{\partial \mathbf{e}_v}{\partial \boldsymbol{\Theta}_{v,l}} = \begin{cases} w_v^{(l)} \sigma\left(Z_s\right) \sigma\left(Z_f\right)\left(1-\sigma\left(Z_s\right)\right)\left(1-\sigma\left(Z_f\right)\right) \mathbf{e}_v^{(l)}, & \boldsymbol{\Theta}=\mathbf{M} \\ w_v^{(l)} \sigma\left(Z_s\right) \sigma\left(Z_f\right)\left(1-\sigma\left(Z_s\right)\right)\left(1-\sigma\left(Z_f\right)\right), & \boldsymbol{\Theta}=\mathbf{b} \\ \sigma\left(Z_s\right) \sigma\left(Z_f\right)\left(1-\sigma\left(Z_s\right)\right), & \boldsymbol{\Theta}=w \end{cases} \tag{15}$$

where $Z_s = \sum_{l=1}^{|P|} w_v^{(l)} \sigma\left(\mathbf{M}^{(l)} \mathbf{e}_v^{(l)} + \mathbf{b}^{(l)}\right)$, $Z_f = \mathbf{M}^{(l)} \mathbf{e}_v^{(l)} + \mathbf{b}^{(l)}$.

4. Experimental Evaluation.

4.1. Dataset of the experiment. We conducted our experiments on three real-world datasets. In order to demonstrate the ability of our method on other heterogeneous information networks, we used two datasets, namely the DBLP[1] and Aminer[2] datasets. Additionally, we used the Online dataset for learning peer recommendations. The DBLP dataset is obtained from the DBLP original dataset, which extracts conference papers published from 2015 to 2020 and guarantees that each paper is not written by a single author. The network schema for the DBLP dataset [14] is shown in Figure 3, which covers author (A), paper (P), and conference venue (V), and two types of relations, namely paper-venue and paper-author. The Aminer dataset is extracted directly from the Aminer original dataset, which selects papers published from 2010 to 2014, and each paper is not written by a single author. Meanwhile, in order to make use of the text information, we selected ten terms with the highest TF-IDF score from the title and abstract of each paper. The network schema for the Aminer dataset [15] is illustrated in Figure 4, which contains author (A), paper (P), venue (V), and term (T). The Online dataset consists of historical behavior data generated by students in grades 2017, 2018, 2019, and 2020 in the process of learning "Data Structure and Algorithm". The network schema for this dataset is shown in Figure 2, which includes student (S), teacher (T), video (V), exercises (E), knowledge points (K) and their relations between each other. The relevant statistics of three datasets are shown in Table 1.

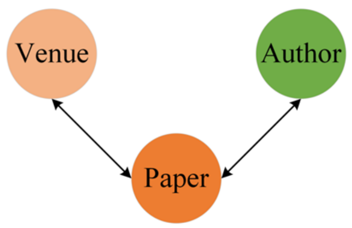


FIGURE 3. Network schema for DBLP

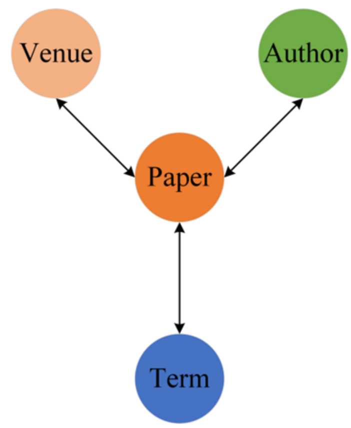


FIGURE 4. Network schema for Aminer

[1]https://www.aminer.cn/citation
[2]https://www.aminer.cn/data/#Academic-Social-Network

TABLE 1. Statistics of the experimental dataset

Datasets	Objects	Number	Links	Number
DBLP	papers	23,607	paper-venues	23,607
	venues	1,796	paper-author	80,535
	authors	4,524	–	–
Aminer	papers	16,358	paper-venues	16,358
	venues	3,765	paper-author	59,343
	authors	3,925	paper-terms	81,790
	terms	10,928	–	–
Online	students	1,055	video-teacher	207
	videos	207	student-exercises	427,478
	teacher	1	student-videos	283,061
	exercises	163	student-knowledge points	310,090
	knowledge points	207	exercises-knowledge points	7,505
	–	–	video-knowledge points	207
	–	–	teacher-knowledge points	10,490

4.2. **Evaluation metrics.** There are two evaluation metrics used to evaluate performance, namely precision and recall. These two metrics can be calculated as follows:

$$Precision = \frac{|rec(u) \cap real(u)|}{|rec(u)|} \tag{16}$$

$$Recall = \frac{|rec(u) \cap real(u)|}{|real(u)|} \tag{17}$$

where $rec(u)$ represents the recommendation list for target student u, and $real(u)$ represents the true learning peer set of target student u.

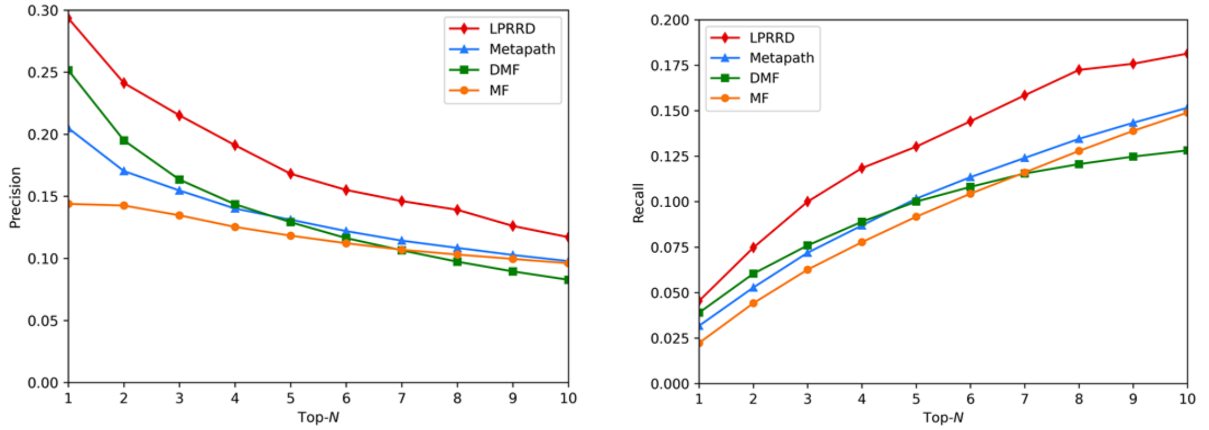
4.3. **Parameter setting.** In our experiment, each dataset is divided into a training set and a test set, with the training set accounting for 80% and the test set accounting for 20%. We set the node embedding dimension to 64, the random walk path length to 5, the window size to 10, and the learning rate to 0.001, the number of latent factors of target students and candidate students to 64. On the basis of these settings, tuning parameters α and β are set to 1 and 1, respectively.

4.4. **Experimental results and analysis.**

4.4.1. *Experimental results.* We conducted experiments on three different datasets using LPRRD, scholar-friend recommendation (named as Metapath) [8], Matrix Factorization (MF) [16], and Deep Matrix Factorization model (DMF) [17]. The experimental results are shown in Figure 5, Figure 6, and Figure 7.

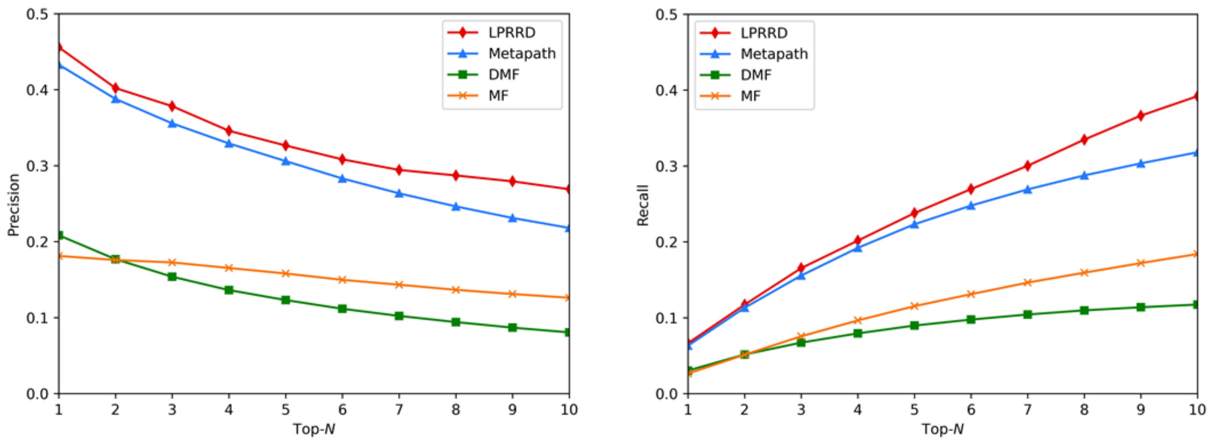
The results show that LPRRD outperforms other baseline methods in terms of precision and recall. Figure 5 shows the recommendation performance of LPRRD and other baseline methods on DBLP. Figure 5(a) shows the highest precision of LPRRD is up to 29.3% when N is 1. Figure 5(b) illustrates that DMF has the worst performance while LPRRD performs the best when N is 10. Figure 6 shows the recommendation performance of four methods on Aminer. Figure 6(a) demonstrates that LPRRD has the highest precision of 45.6%, which indicates that LPRRD has a 45.6% probability of recommending co-authors in the Top-1 recommendation list. Figure 6(b) shows that the recall of LPRRD is 39.2% when N is 10, which indicates that there are 39.2% true co-authors in the Top-10 recommendation list. Figure 7 shows the recommendation performance of four methods on Online. As shown in Figure 7(a), the highest precision of LPRRD reaches 28.3% when N is 1, which is 4.4% higher than the second best method DMF. As can be seen from

Figure 7(b), the highest recall of LPRRD reaches 29.9% when N is 10, which is 4.3% higher than the second best method DMF.



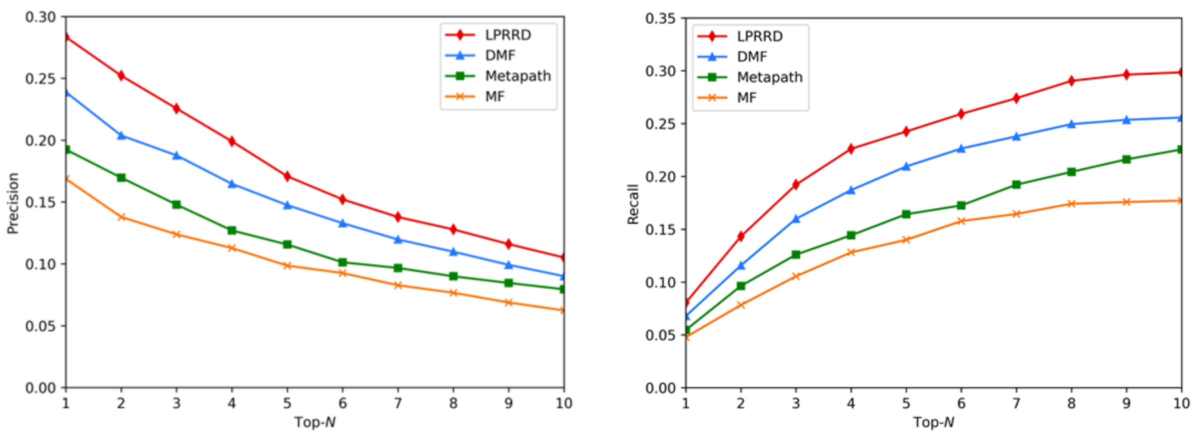
(a) Precision (b) Recall

FIGURE 5. Precision and Recall of LPRRD and baseline methods on DBLP



(a) Precision (b) Recall

FIGURE 6. Precision and Recall of LPRRD and baseline methods on Aminer



(a) Precision (b) Recall

FIGURE 7. Precision and Recall of LPRRD and baseline methods on Online

To sum up, the proposed algorithm LPRRD significantly outperforms other methods in precision and recall on the three datasets. Compared with Metapath, LPRRD combines MLP with network representation learning to solve the problems of the limited expressiveness of dot product and the weakness in capturing low-rank relations. The recommendation performance of DMF is inferior to that of LPRRD because it fails to take account of the heterogeneity of the various types of objects (e.g., students, teachers, videos, exercises and knowledge points) and the complex interactions among these objects. MF has the worst performance because MF only considers the interaction behavior between students and does not consider other behavior data in the learning process.

4.4.2. *Complexity analysis.* The complexity analysis of LPRRD should include two aspects. 1) Network representation learning. The complexity of deepwalk is $O(\tau \cdot |\mathcal{V}| \cdot t \cdot w \cdot (d + d \cdot \log |\mathcal{V}|))$, where τ is the number of random walks, t is the length of random walk, w is the size of neighbor, d is the embedding dimension and $|\mathcal{V}|$ is the number of nodes in the network. Hence, the total complexity of network representation learning can be described as $O(|P| \cdot \tau \cdot t \cdot w \cdot d \cdot (|u| \cdot \log |U| + |V| \cdot \log |V|))$, where the number of selected meta-paths is $|P|$, the number of users and items is $|U|$ and $|V|$, respectively. For each triplet $\langle u, v, r_{u,v} \rangle$, updating $\gamma_u^{(U)}$ and $\gamma_v^{(V)}$ takes $O(D)$ time, where D is the number of latent factors. And updating $\Theta_u^{(U)}$ and $\Theta_v^{(V)}$ takes $O(|P| \cdot D \cdot d)$ to learn the transformation matrices M for all meta-paths. 2) Deep learning. Deep learning mainly includes representation learning and matching function learning in this paper. When we calculate the complexity, we only focus on the layers with computing power. The above two parts' complexity includes space complexity and time complexity. The space complexity is expressed by the number of neural network layers and the number of parameters to be optimized in the neural network, and time complexity can be expressed by the number of multiplication and addition operations in neural networks. In order to unify the above network representation learning complexity analysis, we only analyze the time complexity. The complexity of representation learning is $O(l1 \cdot l2)$, where $l1$ and $l2$ are output dimension of input layer and input dimension of hidden layer, respectively. And the complexity of matching function learning is $O(L1 \cdot L2 + L2 \cdot L3)$, where $L1$, $L2$ and $L3$ are output dimension of first-layer, input dimension of second-layer, and output dimension of third-layer, respectively. Finally, the complexity of model fusion is $O(L4 * 1)$, where $L4$ is output dimension of fourth-layer.

5. **Conclusions.** In order to improve students' enthusiasm for online course learning, we propose a learning peer recommendation algorithm based on heterogeneous information network representation learning and deep learning. This algorithm constructs a heterogeneous network to integrate students, teachers, videos, exercises, knowledge points, and their relations into a unified framework. In addition, we combine MLP with network representation learning to solve the problems of the limited expressiveness of dot product and the weakness in capturing low-rank relations. Experimental results show that LPRRD outperforms other baseline methods in terms of precision and recall. Future research may combine heterogeneous information networks and broad learning to discover more fine-grained student preference patterns and greatly improve recommendation performance.

Acknowledgment. This work is partially supported by the National Natural Science Foundation of China (62177012, 61967005), Innovation Project of GUET Graduate Education (2020YCXS022, 2021YCXS033), the Key Laboratory of Cognitive Radio and Information Processing Ministry of Education (CRKL190107). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] B. Xu and D. Yang, Study partners recommendation for xMOOCs learners, *Computational Intelligence and Neuroscience*, DOI: 10.1155/2015/832093, 2015.

[2] Q. Hu, Z. Han, X. Lin, Q. Huang and X. Zhang, Learning peer recommendation using attention-driven CNN with interaction tripartite graph, *Information Sciences*, vol.479, pp.231-249, 2019.

[3] F. Bouchet, H. Labarthe, K. Yacef and R. Bachelet, Comparing peer recommendation strategies in a MOOC, *Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pp.129-134, 2017.

[4] D. Yang, M. Wen and C. Rose, Peer influence on attrition in massive open online courses, *Proceedings of Educational Data Mining*, vol.5, no.2, 2014.

[5] F. Bouchet, H. Labarthe, R. Bachelet and K. Yacef, Who wants to chat on a MOOC? Lessons from a peer recommender system, *European Conference on Massive Open Online Courses*, pp.150-159, 2017.

[6] B. A. Potts, H. Khosravi, C. Reidsema, A. Bakharia, M. Belonogoff and M. Fleming, Reciprocal peer recommendation for learning purposes, *Proc. of the 8th International Conference on Learning Analytics and Knowledge*, pp.226-235, 2018.

[7] C. Shi, Y. Li, J. Zhang, Y. Sun and P. S. Yu, A survey of heterogeneous information network analysis, *IEEE Trans. Knowledge and Data Engineering*, vol.29, no.1, pp.17-37, 2016.

[8] Y. Xu, D. Zhou and J. Ma, Scholar-friend recommendation in online academic communities: An approach based on heterogeneous network, *Decision Support Systems*, vol.119, pp.1-13, 2019.

[9] C. Shi, B. Hu, W. X. Zhao and P. S. Yu, Heterogeneous information network embedding for recommendation, *IEEE Trans. Knowledge and Data Engineering*, vol.31, no.2, pp.357-370, 2018.

[10] A. S. Karnyoto, C. Sun, B. Liu and X. Wang, Augmentation and heterogeneous graph neural network for AAAI2021-COVID-19 fake news detection, *International Journal of Machine Learning and Cybernetics*, pp.1-11, 2022.

[11] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang and M. Zhang, DisenHAN: Disentangled heterogeneous graph attention network for recommendation, *Proc. of the 29th ACM International Conference on Information & Knowledge Management*, pp.1605-1614, 2020.

[12] Z. H. Deng, L. Huang and C. D. Wang, DeepCF: A unified framework of representation learning and matching function learning in recommender system, *Proceedings of the AAAI Conference on Artificial Intelligence*, vol.33, no.1, pp.61-68, 2019.

[13] A. Grover and J. Leskovec, node2vec: Scalable feature learning for networks, *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.855-864, 2016.

[14] 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.

[15] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang and Z. Su, Arnetminer: Extraction and mining of academic social networks, *Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.990-998, 2008.

[16] Y. Koren, R. Bell and C. Volinsky, Matrix factorization techniques for recommender systems, *Computer*, vol.42, no.8, pp.30-37, 2009.

[17] H. J. Xue, X. Dai, J. Zhang, S. Huang and J. Chen, Deep matrix factorization models for recommender systems, *IJCAI*, vol.17, pp.3203-3209, 2017.