electronics

Article

Learning Peer Recommendation Based on Weighted Heterogeneous Information Networks on Online Learning Platforms

Zhaoyu Shou [1,†], Zhixuan Shi [1,†], Hui Wen [1,*], Jinghua Liu [1] and Huibing Zhang [2]

1 School of Information and Communication, Guilin University of Electronic Technology, Guilin 541004, China
2 School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
* Correspondence: huiwen@guet.edu.cn
† These authors contributed equally to this work.

Abstract: With the development of online education, there is an urgent need to solve the problem of the low completion rate of online learning courses. Although learning peer recommendation can effectively address this problem, prior studies of learning peer-recommendation methods extract only a portion of the interaction information and fail to take into account the heterogeneity of the various types of objects (e.g., students, teachers, videos, exercises, and knowledge points). To better motivate students to complete online learning courses, we propose a novel method to recommend learning peers based on a weighted heterogeneous information network. First, we integrate the above different objects, various relationships between objects, and the attribute values to links in a weighted heterogeneous information network. Second, we propose a method for automatically generating all meaningful weighted meta-paths to extract and identify meaningful meta-paths. Finally, we use the Bayesian Personalized Ranking (BPR) optimization framework to discover the personalized weights of target students on different meaningful weighted meta-paths. We conducted experiments using three real datasets, and the experimental results demonstrate the effectiveness and interpretability of the proposed method.

Keywords: online learning; learning peer recommendation; weighted heterogeneous information network; weighted meta-paths

Citation: Shou, Z.; Shi, Z.; Wen, H.; Liu, J.; Zhang, H. Learning Peer Recommendation Based on Weighted Heterogeneous Information Networks on Online Learning Platforms. *Electronics* **2023**, *12*, 2051. https://doi.org/10.3390/electronics12092051

Academic Editor: Xianzhi Wang

Received: 9 March 2023
Revised: 13 April 2023
Accepted: 27 April 2023
Published: 29 April 2023

1. Introduction

The booming development of the Internet has accelerated education reform, and online learning is growing rapidly. The Corona Virus Disease 2019 (COVID-19) pandemic has seriously restricted offline classroom teaching [1], and online learning, as an extension and an essential complement to offline education [2], is brought into the spotlight [3,4]. Currently, it is commonly observed that students drop out of online learning courses in the learning process [5]. This is because teachers and students are separated in space and time. When faced with difficulties, students are unable to communicate with each other in real time, which may result in feelings of loneliness and helplessness [6,7].

The above phenomenon can be effectively alleviated by learning peer recommendation [8–10]. Xu et al. [6] constructed a social network based on learners' activity information in a course forum and recommended learning peers to target learners based on the network's link relationships. Hu et al. [7] proposed a Learning Peer Recommendation (LPR) framework for learning peer recommendation that used a dynamic interaction tripartite graph to depict the complex relationships among learners, learning content, and interaction behaviors and adopted Convolutional Neural Network (CNN) to adjust the weight of interaction behaviors. Potts et al. [11] created the Open-Source Course-Level Recommendation (RiPPLE) platform to assist target learners in finding suitable learning peers based on their learning logs. Prabhakar et al. [12] developed a reciprocal recommendation system to match learners with similar interests. However, these methods extract only a portion of the interaction information in the online learning process for modeling purposes and fail to take into account the heterogeneity of the various types of objects and the complex interactions among these objects, resulting in a great deal of significant information loss and a negative impact on recommendation performance. In the learning process, heterogeneous information networks provide an effective solution for modeling data heterogeneity, which opens up a lot of potential for data mining [13]. Xu et al. [14] proposed a heterogeneous information network model to provide scholar-friend recommendations. Liu et al. [15] proposed a method based on heterogeneous information network embedding for scientific collaborator recommendation. Unfortunately, the performance of these methods is limited by three factors. First, it is difficult to manually generate the meta-path set in practice. Second, these methods are only applicable to symmetric meta-paths. Third, the effect of attribute values on links is not considered. Karnyoto et al. [16] adopted a heterogeneous graph neural network to detect fake news related to COVID-19. Wu et al. [17] proposed a heterogeneous graph attention network method for course recommendation, which learned student and course representations in a heterogeneous graph. 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.

To solve the problem of the low completion rate of online learning courses, this study overcomes the limitations of the above-mentioned existing research. We propose a learning peer-recommendation method based on a weighted heterogeneous information network (LPRWHIN) by leveraging the behavior data and exercises test data collected in the learning process. The main contributions of this study are summarized as follows:

- We construct a weighted heterogeneous information network to retain semantic, structural, and attribute information more comprehensively, which consists of multiple types of objects (e.g., students, teachers, videos, exercises, and knowledge points), relationships between objects (e.g., students-knowledge points, students-videos, and students-exercises) and attribute values on links (e.g., the degrees of student–system interaction, the degrees of student–teacher interaction, the degrees of student–student interaction, and test scores).
- A method for automatically generating meaningful weighted meta-paths is proposed, which makes full use of network structural information to flexibly extract and then effectively identify all meaningful meta-paths for learning peer recommendations.
- The Bayesian Personalized Ranking optimization framework is employed to calculate the personalized weights of target students on selected weighted meta-paths.

The rest of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 presents relevant definitions. Section 4 describes the proposed method in detail. Section 5 reports experimental results. Section 6 concludes this paper and gives future research directions.

2. Related Works

This section provides a brief review of the main approaches commonly used in recommendation systems.

2.1. Content-Based Filtering and Collaborative Filtering Methods

Content-based filtering methods recommend items by matching user preferences in user profiles with item attributes [18]. Huang et al. [19] presented a content-based method for students to choose online courses based on their own interests. Camposet et al. [20] recommended courses for students based on a content-based filtering method. However, the courses recommended by these methods are closely related to those that the target students were previously interested in, resulting in a lack of novelty and diversity in the recommended courses. Collaborative filtering methods provide recommendations based on the similar preferences of peer users. Pang et al. [21] adopted a collaborative filtering recommendation approach to recommend learning paths for students. Zhao et al. [22] developed a user-based collaborative filtering recommendation method based on students' behavior logs and learning resources' ratings, and applied it to the Massive Open Online Courses (MOOC) platform to provide them with resources. However, these methods rely heavily on student feedback or learning behavior data, and there are problems associated with the cold start or data sparsity.

2.2. Hybrid Recommendation Methods

Hybrid recommendation methods are characterized by the combination of multiple recommendation methods [23]. Safarov et al. [24] proposed a Deep Neural Networks (DNN) approach that combined the K-Means algorithm and deep neural networks to generate candidates on e-learning platforms. Wu [25] presented a method combining deep learning and collaborative filtering to recommend MOOC resources. Liu et al. [26] proposed a hybrid method based on deep learning and collaborative filtering, which not only recommended courses suitable for student interests and preferences but also predicted their performance in each course. Hu et al. [7] outlined a framework for learning peer recommendation by combining a tripartite graph and CNN. Gong et al. [27] proposed a knowledge concept recommendation method that constructed a heterogeneous information network to capture semantic relationships between different types of entities and incorporated them into the representation learning process. Although these methods can be used to overcome the limitations of a single recommendation method [28,29], their time complexity is relatively high due to the mixing of multiple recommendation models.

2.3. Network-Based Recommendation Methods

Network-based recommendation methods have been widely developed to model more interaction information and improve recommendation performance. Li et al. [30] proposed a novel social recommendation method based on a homogeneous information network that ignored the heterogeneity between users and topics. Ma et al. [31] proposed a novel method that leveraged bipartite graphs to provide social recommendations. Paleti et al. [32] utilized two tripartite graphs to integrate the interaction information among users, items, ratings, and opinions for social recommendations. Although bipartite and tripartite graphs can integrate heterogeneous information, they cannot integrate more types of objects and relationships. Liu et al. [15] modeled data heterogeneity based on a heterogeneous information network to provide scientific collaborator recommendations. Li et al. [33] proposed an approach based on a heterogeneous information network for making paper recommendations. Li et al. [34] applied a heterogeneous information network to approach the expert recommendation problem. Unfortunately, the above studies only consider various types of objects and relationships, and ignore the effect of the attribute value on links in heterogeneous information networks. For example, the order of authors in a paper corresponds to their relative contribution to the work. We can analyze the semantic relationships between authors and papers more precisely using the contribution of the authors as the attribute value on the "Author-Paper" link.

3. Relevant Definitions

This section explains the term definitions used in this study to illustrate the method in this paper more clearly.

3.1. Three Types of Interaction Degrees

3.1.1. The Degree of Student–System Interaction

The degree of student–system interaction can be defined as the workload that students watch knowledge point videos [35]. The degree of student–system interaction $SC_{u,K}$ can be expressed as Formula (1):

$$SC_{u,K} = \lambda_1 \times f_{SC_{u,K}} + \lambda_2 \times t_{SC_{u,K}} + \lambda_3 \times p_{SC_{u,K}}, \tag{1}$$

where $f_{SC_{u,K}}$ represents the frequency of student u learning knowledge point K, $t_{SC_{u,K}}$ represents the duration of student u learning knowledge point K, and $p_{SC_{u,K}}$ represents the pause and drag frequency of student u learning knowledge point K. Literature [36] concludes that the degree of student–system interaction is best characterized when $(\lambda_1, \lambda_2, \lambda_3) = (1, 5, 4)$.

3.1.2. The Degree of Student–Teacher Interaction

The degree of student–teacher interaction is measured by the workload that students interact with the teacher [35], which is primarily determined by the questioning and answering process between the teacher and students. The degree of student–teacher interaction $ST_{u,K}$ can be expressed as Formula (2):

$$\begin{cases} ST_{u,K} = W_{ST_{u,K}} \times f_{ST_{u,K}} \\ W_{ST_{u,K}} = \frac{t_{ST_{u,K}}}{\max\left\{t_{ST_{1,K}}, t_{ST_{2,K}}, \cdots, t_{ST_{d,K}}\right\}} \end{cases}, \tag{2}$$

where $W_{ST_{u,K}}$ represents the weight coefficient of the student–teacher interaction of student u to knowledge point K, $f_{ST_{u,K}}$ represents the student–teacher interaction frequency of student u to knowledge point K, d is the total number of students learning the online course for each grade, $t_{ST_{u,K}}$ represents the student–teacher interaction duration of student u to knowledge point K, and $\max\left\{t_{ST_{1,K}}, t_{ST_{2,K}}, \cdots, t_{ST_{d,K}}\right\}$ is the maximum duration of student–teacher interaction to the knowledge point K.

3.1.3. The Degree of Student–Student Interaction

The degree of student–student interaction represents the workload that a student puts into interacting with other students regarding knowledge points [35]. The degree of student–student interaction $SS_{u,K}$ can be expressed as Formula (3):

$$\begin{cases} SS_{u,K} = \frac{\sum_{v=1}^{d-1} W_{SS_{uv,K}} \times f_{SS_{uv,K}}}{d-1} \cdots u \neq v \\ W_{SS_{uv,K}} = \frac{t_{SS_{uv,K}}}{\max\left\{t_{SS_{u1,K}}, t_{SS_{u2,K}}, \cdots, t_{SS_{ud,K}}\right\}} \end{cases}, \tag{3}$$

where $W_{SS_{uv,K}}$ represents the weight coefficient of the student–student interaction between student u and student v to knowledge point K, $f_{SS_{uv,K}}$ represents the frequency of interaction between student u and student v to knowledge point K, $t_{SS_{uv,K}}$ represents the interaction duration between student u and student v to knowledge point K, and $\max\left\{t_{SS_{u1,K}}, t_{SS_{u2,K}}, \cdots, t_{SS_{ud,K}}\right\}$ represents the maximum duration of student–student interaction between student u and other students to the knowledge point K.

3.2. Weighted Heterogeneous Information Network

A weighted heterogeneous information network can be described as a directed graph $G = (V, E, W)$ with an object type mapping function $\phi : V \rightarrow \mathcal{A}$, a link type mapping function $\psi : E \rightarrow \mathcal{R}$ and an attribute value type mapping function $\theta : W \rightarrow \mathcal{W}$, where the types of objects $|\mathcal{A}| > 1$ (or the types of relationships $|\mathcal{R}| > 1$) and the types of attribute values $|\mathcal{W}| > 0$. Based on the graph, each object $\nu \in V$ belongs to a particular object type $\phi(v) \in \mathcal{A}$, each link $e \in E$ belongs to a particular relationship type $\psi(e) \in \mathcal{R}$, and each attribute value $\omega \in W$ belongs to a particular attribute value type $\theta(\omega) \in \mathcal{W}$ [37]. A conventional heterogeneous information network is an unweighted heterogeneous information network where there are no attribute values on links or we ignore them. For a weighted heterogeneous information network, there are attribute values on links, and these attribute values may be discrete or continuous.

Figure 1 illustrates the weighted heterogeneous information network of objects and their relationships on an online learning platform. R_i denotes a particular type of relationship between two different types of objects (R_i^{-1} represents the inverse relationship of R_i), and $\delta(R_i)$ is the range of attribute values on relationship R_i. The figure depicts different relationships 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.

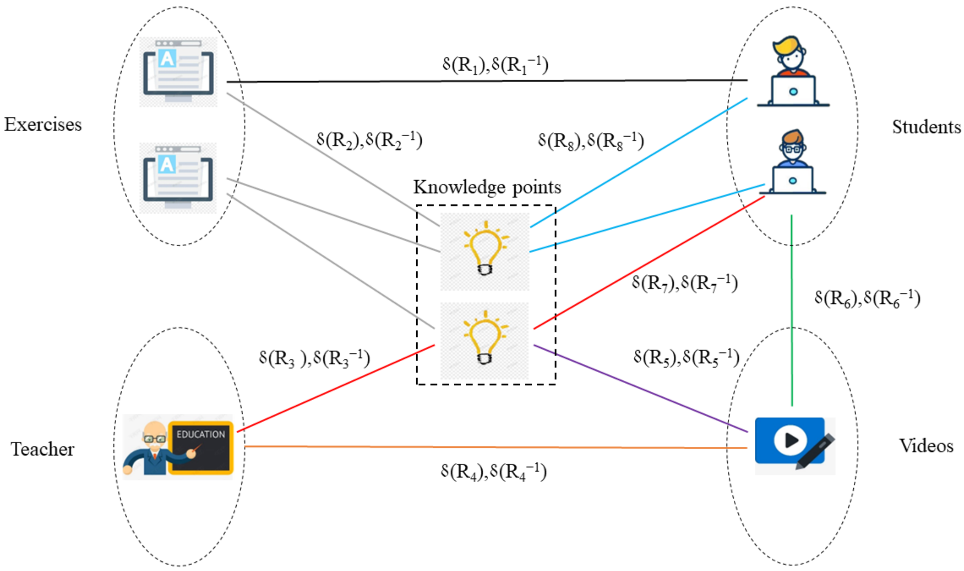


Figure 1. A weighted heterogeneous information network on an online learning platform.

According to literature [17,27], the network consists of five types of objects (e.g., students, teachers, videos, exercises, and knowledge points), sixteen types of relationships, and the attribute values on links. Links between two types of objects represent different relationships. For example, the network can be described in detail using various relationships among objects and the attribute values on links. "Students-do-Exercises" describes exercises test behavior, where links exist between students and exercises denoting the relationship R_1, and $\delta(R_1)$ represents the range of test scores. "Students-question-Knowledge points-answered by-Teachers" describes student–teacher interaction behavior, where links exist between teachers and knowledge points denoting the relationship R_3, and $\delta(R_3)$ represents the range of the degrees of student–teacher interaction. "Students-watch-Videos-include-Knowledge points" describes student–system interaction behavior, where links exist between students and videos denoting the relationship R_6 and $\delta(R_6)$ represents the range of the degrees of student–system interaction. "Students-discuss-Knowledge points-discussed by-Students" describes student–student interaction behavior, where links exist between students and knowledge points denoting the relationship R_8 and $\delta(R_8)$ represents the range of the degrees of student–student interaction.

3.3. Network Schema

The network schema $T_G = (\mathcal{A}, \mathcal{R}, \mathcal{W})$ is a meta structure of the weighted heterogeneous information network $G = (V, E, W)$, which comprises a set of object types $\mathcal{A} = \{A\}$, a set of relationships connecting object pairs $\mathcal{R} = \{R\}$, and a set of attribute values on relationships $\mathcal{A} = \{A\}$ [37].

Figure 2 illustrates the network schema of the weighted heterogeneous information network on the online learning platform, which contains five different types of objects: students (S), teachers (T), videos (V), exercises (E), and knowledge points (K). Multiple objects are connected via various links. It is worth noting that there are four types of relationships between students and knowledge points: (1) students discuss the knowledge points; (2) knowledge points are discussed by students; (3) students ask the teacher questions about a certain knowledge point; and (4) knowledge points are questioned by students.

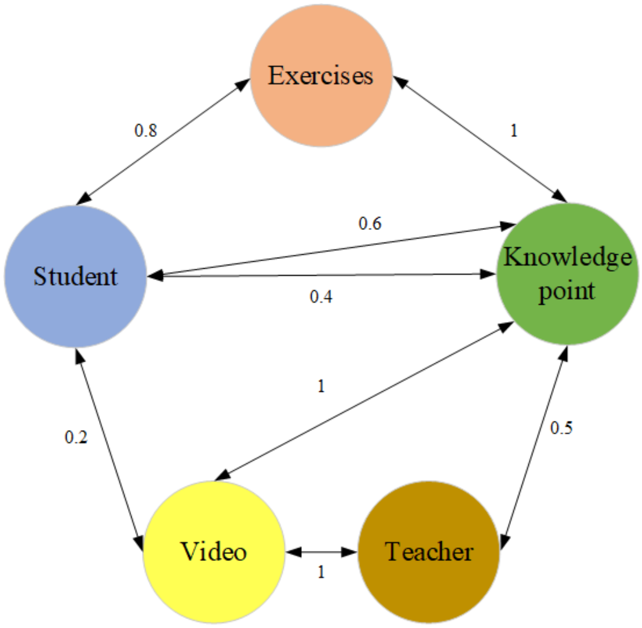


Figure 2. An example of network schema on an online learning platform.

3.4. Weighted Meta-Path

A meta-path $MP = A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \ldots \xrightarrow{R_l} A_{l+1}$ is a new composite relationship $R = R_1 \circ R_2 \ldots R_l$ between object types A_1, A_2, A_l, where A_i represents an object type, R_j represents a relationship and $\circ$ represents composition operator on relationships, $(i = 1, 2, \ldots, l+1; j = 1, 2, \ \ldots, l)$. In addition, a meta-path with symmetric relationship types is called a symmetric meta-path.

Based on the above description, a weighted meta-path $MP' = A_1 \xrightarrow{\delta(R_1)} A_2 \xrightarrow{\delta(R_2)} \ldots \xrightarrow{\delta(R_l)} A_{l+1}$ is a path defined based on a network schema $T_G = (\mathcal{A}, \mathcal{R}, \mathcal{W})$, where $\delta(R_j)$ represents the range of attribute values on relationship, $(i = 1, 2, \ldots, l+1; j = 1, 2, \ldots, l)$. Basically, a concrete path $mp = a_1 \xrightarrow{\omega_1} a_2 \xrightarrow{\omega_2} \ldots \xrightarrow{\omega_l} a_{l+1}$ is a path instance of the weighted meta-path if each link $e_i = < a_i, a_{i+1} > \in R_i$, each object $a_i \in A_i$, and each attribute value $\omega_i \in \delta(R_i)$ [38]. A conventional meta-path can be viewed as a special case of a weighted meta-path, and three types of interaction degrees and test scores can be regarded as ratings and used as attribute values on links. As a result, there are two types of attribute values in Figure 2: one is similar to the rating relationships between students and knowledge points, the attribute values on links of which range from 0 to 1, while the other is associated with all the relationships except rating relationships, the attribute values of its links all take value 1. In addition, Table 1 shows examples of weighted meta-paths and their corresponding semantic meanings in this study, these semantic information can effectively reflect the different types of learning behaviors of students and the workload put into learning knowledge points.

Table 1. Examples of weighted meta-paths and their semantic meanings.

Weighted Meta-Path	Path Instance	Semantic Meaning
Student $\xrightarrow{\delta(do)}$ Exercises $\xrightarrow{\delta(done)}$ Student	Liu $\xrightarrow{0.6}$ Binary tree $\xrightarrow{0.6}$ Li	Liu and Li have the same test score of 0.6 on the binary tree
Student $\xrightarrow{\delta(question)}$ Knowledge point $\xrightarrow{\delta(answered\ by)}$ Teacher	Wu $\xrightarrow{0.5}$ Stack $\xrightarrow{0.5}$ Zhang	Wu puts into interacting with Zhang on the stack, and the degree of student–teacher interaction is 0.5

4. Proposed Methodology

Based on the above definitions, we propose a learning peer-recommendation method based on a weighted heterogeneous information network, as shown in Figure 3. The proposed method consists of the following five steps. First, we construct a weighted heterogeneous information network containing multiple types of objects and relationships according to the corresponding network schema on an online learning platform. Second, we propose a method for automatically generating all meaningful weighted meta-paths based on the specified network schema for exploring the various links between target students and candidate students. Third, we apply random walk to calculate the recommendation scores of candidate students in each meaningful weighted meta-path. Fourth, we adopt a Bayesian Personalized Ranking optimization framework to learn personalized weights for all weighted meta-paths. Fifth, we make a Top-N recommendation list for each target student based on the recommendation scores.

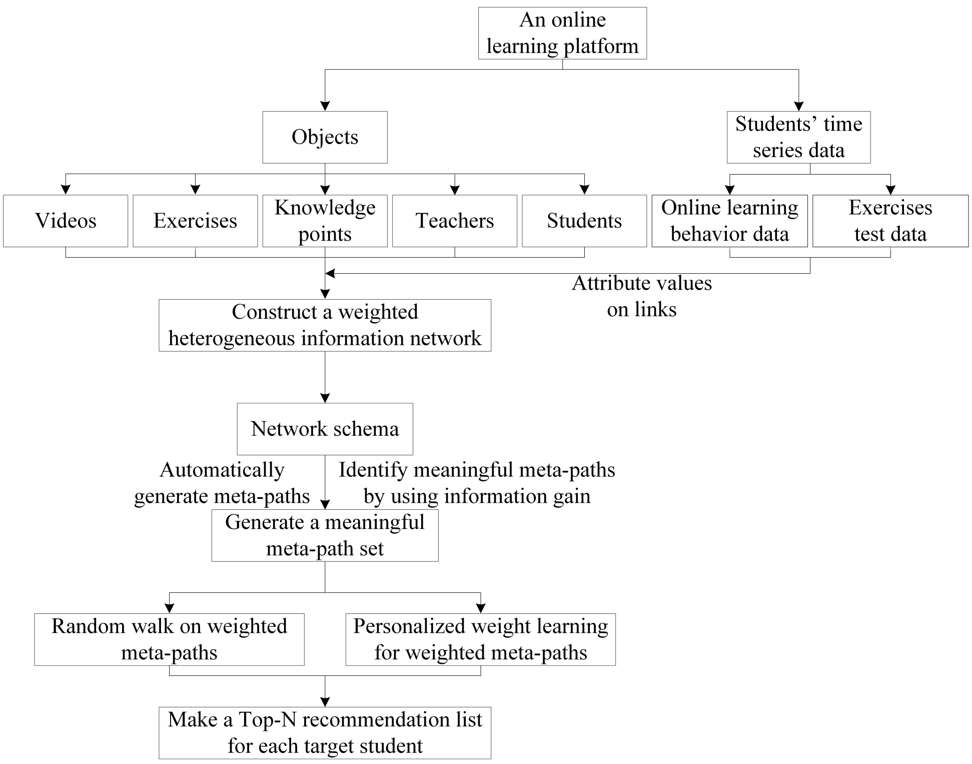


Figure 3. Block diagram of LPRWHIN.

4.1. Constructing a Weighted Heterogeneous Information Network

The construction of a weighted heterogeneous information network on the online learning platform is shown in Figure 1. First, the types of objects and relationships are determined from the online learning platform dataset (Online dataset) to specify a network schema $T_G = (\mathcal{A}, \mathcal{R}, \mathcal{W})$. Second, according to the obtained network schema, we build a weighted heterogeneous information network to model the interaction behaviors among students, teachers, videos, exercises, and knowledge points, in which the degrees of student–system interaction, the degrees of student–teacher interaction, the degrees of student–student interaction and test scores are used as attribute values.

We build the adjacency matrix of this network, and then the adjacency matrix $A_{MN} = [a_{ij}]_{p \times q}$ between object types M and N can be defined as Formula (4):

$$a_{ij} = \begin{cases} w_{ij} & \textit{if} \text{ a link exists between } M_i \text{ and } N_j \\ 0 & \textit{otherwise} \end{cases}, \tag{4}$$

where p and q are described as corresponding indexes of M and N, respectively. w_{ij} is the attribute value on the link of the relationship between M_i and N_j, and all the attribute values will be normalized to a unified range between 0 and 1 [39]. A point worth noting is that there are different relationship types between objects. For example, A_{SV} represents the adjacency matrix of the watch relationship between students and videos, and A_{VS} represents the adjacency matrix of the watched-by relationship between videos and students.

4.2. Generating a Meta-Path Set

Meta-path contains rich semantic information and can be used to discover students' interests and preferences. It is an effective tool for modeling student preferences in heterogeneous information networks [13]. Literature [14] employed a method for manually specifying meta-paths to develop a meta-path set related to the recommendation task. This method not only relies on domain expert knowledge but also omits some meta-paths due to manual negligence, which leads to an incomplete meta-path set, resulting in the loss of a large amount of information and the degradation of recommendation performance.

Theoretically, all meta-paths can be used for learning peer recommendation [33]. However, the redundancy and the noise in some meta-paths greatly increase the computational cost of personalized weight-learning for meta-paths and bring noise information into the recommendation model.

Therefore, this paper proposes a method for automatically generating meaningful meta-paths that can automatically extract and then effectively identify all meaningful meta-paths for learning peer recommendations. The method comprises two steps. First, a complete meta-path set is generated by the method for automatically extracting meta-paths to ensure that the meta-paths used for learning peer recommendation can cover richer and more objective semantic information and help to improve recommendation performance. Second, the information gained is used to estimate the values of these meta-paths, and based on these values, the meta-paths with noisy and redundant information are eliminated, and the meaningful meta-paths are retained for learning peer recommendation. Due to the reduction of meta-paths containing noisy and redundant information, the weights will be allocated to meaningful meta-paths, which will help to reduce computational costs.

4.2.1. Extracting Meta-Paths

The method for automatically extracting meta-paths can automatically extract any type of symmetric and asymmetric meta-paths. Since the purpose of this study is to recommend learning peers to students, we only focus on meta-paths with student type as source and target objects. In addition, to minimize the time complexity of searching for meta-paths, this paper only extracts meta-paths within a limited length. Figure 2 shows the network schema $T_G = (\mathcal{A}, \mathcal{R}, \mathcal{W})$ of the weighted heterogeneous information network on the online learning platform. In this paper, the maximum length of meta-paths to be extracted is set to 4, and the source object type and the target object type are both S (student), then meta-paths are automatically searched according to the given network schema.

4.2.2. Selecting Meaningful Meta-Paths

The meaningful meta-paths can reveal students' unique preferences when they choose learning peers. In the process of automatically generating a complete meta-path set, each merged relationship in the meta-path should add useful information to the recommendation model [40]. Therefore, according to the information gained from each meta-path, we abandon the meta-paths with more noisy and redundant information to ensure that the meaningful meta-paths are selected for our recommendation task. For a certain meta-path $P = M \xrightarrow{R_1} \ldots N$, the information gain obtained from object type M to object type N following the meta-path P can be expressed as Formula (5):

$$IG_P = H(M) - H_P(M \mid N), \quad (5)$$

where IG_P is the information gain of the meta-path P, which represents the information value starting from M and arriving at N via P. If IG_P is close to zero, it means that the meta-path P does not add additional useful information beyond what is already contained in the dimension M. $H(M)$ denotes the entropy of object type M, and $H_P(M \mid N)$ is the conditional entropy obtained following the meta-path P. $H(M)$ can be expressed as Formula (6):

$$H(M) = -\sum_{m \in M} p(m) \log(p(m)), \tag{6}$$

where $p(m)$ is defined by the degree centrality, which represents the probability of node m with object type M, and can be expressed as Formula (7):

$$p(m) = \frac{Degree\ (m)}{\sum\limits_{x \in M} Degree\ (x)}, \tag{7}$$

where $Degree(x)$ represents the number of edges that belong to the relationship type R_1 incident on node m, and R_1 is the first relationship type in the meta-path P. $\sum_{x \in M} Degree(x)$ represents the total degree of all nodes of the object type.

Conditional entropy $H_P(M \mid N)$ is used to measure the information uncertainty of the dimension M given dimension N. $H_P(M \mid N)$ can be expressed as Formula (8):

$$\begin{aligned} H_P(M \mid N) &= \sum_{n \in N} p(N = n) \times H_P(M \mid N = n) \\ &= -\sum_{n \in N} p(n) \times \sum_{n \in N} p_P(m \mid n) \log p_P(m \mid n) \end{aligned}, \tag{8}$$

where $p_P(m \mid n)$ represents the conditional probability of the meta-path P and can be expressed as Formula (9):

$$p_P(m \mid n) = \frac{\|\{k \mid m \in M, n \in N, k \in P, m \rightarrow_k n\}\|}{\sum\limits_{y \in N} \|\{k \mid m \in M, y \in N, k \in P, m \rightarrow_k y\}\|}, \tag{9}$$

where $\|\{k \mid m \in M, n \in N, k \in P, m \rightarrow_k n\}\|$ represents the number of path instances from node m to node n following the meta-path P, and $\sum_{y \in N} \|\{k \mid m \in M, y \in N, k \in P, m \rightarrow_k y\}\|$ represents the total number of path instances from node m to node y following the meta-path P.

The normalization of information gain can make this method suitable not only for those meta-paths with the same source and target objects in this study. $\widehat{IG_P}$ can be expressed as Formula (10):

$$\widehat{IG_P} = \frac{IG_P}{\sqrt{H(M)H(N)}}, \tag{10}$$

The meta-path with the lower information gain value has a higher probability of being noisy and containing less information. Taking the meta-path P_{eg} = Student $\xrightarrow{question}$ Knowledge point $\xrightarrow{answered\ by}$ Teacher as an example, the information gain calculation process is described. We first present how to obtain $H(Student)$. As shown in Formula (6), the key step is to calculate $p(m)$ using Formula (7), where $Degree(m)$ represents the number of edges that belong to the relationship type $question$ on the node m, and $\sum_{x \in M} Degree(x)$ represents total number of edges on nodes of type $Student$. We then figure out how to compute $H_{P_{eg}}(Student \mid Teacher)$. Its calculation process is similar to that of $H(Student)$. $p_{P_{eg}}(m \mid n)$ and $H_{P_{eg}}(Student \mid Teacher)$ can be calculated using Formula (9) and Formula (8), respectively. Finally, the information gain $IG_{P_{eg}}$ of P_{eg} can be calculated using Formula (5) and normalized using Formula (10).

Figure 4 shows the specific process of automatically generating meaningful meta-paths according to the given network schema in Figure 2, where the solid lines indicate the relationships between objects, the dashed lines represent the searched meta-path, and the solid lines marked with circles denote the selected meta-paths using information gain. when the maximum length of the meta-paths is set to 4, the source object type and target object type are both S (student), and the method for automatically extracting meta-paths will perform 4 iterations.

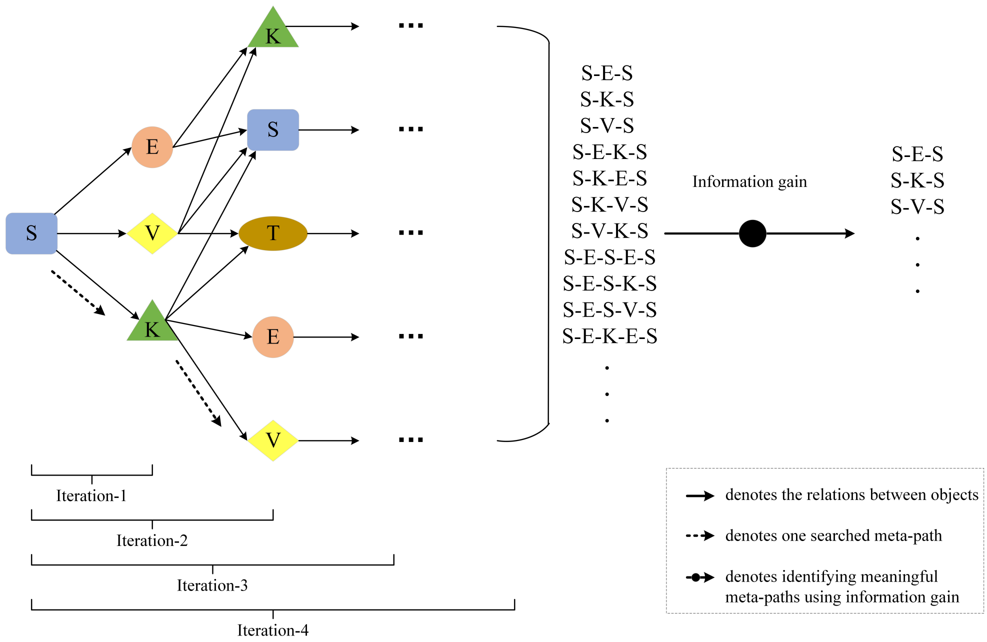


Figure 4. Example of automatically generating meta-paths.

In the first iteration, the meta-paths with source object type S are searched, which are expressed as $fst\_Set = \{SE, SV, SK\}$. The meta-paths with source object type S and target object type S are extracted in fst_Set, and these meta-paths are represented as $pathSet$. $pathSet$ is empty in this iteration.

In the second iteration, we extract the target object types of each element in fst_Set of the previous iteration, expressed as $targetObjects = \{E, V, K\}$. After obtaining $targetObjects$, it will be regarded as the source object type in this iteration. According to the network schema in Figure 2, we can obtain $snd\_Set = \{EK, ES, VK, VS, VT, KS, KT, KE, KV\}$. Then the fst_Set in this iteration can be obtained by merging the fst_Set in the previous iteration with the current snd_Set, which can be expressed as $fst\_Set = \{SEK, SES, SVK, SVS, SVT, SKS, SKT, SKE, SKV\}$. Similarly, we only focus on the meta-paths with source object type S and target object type S in fst_Set, $pathSet$ are now updated to $\{SES, SVS, SKS\}$.

The process of the third and fourth iteration is similar to that of the second iteration. After four iterations, we can extract the meta-paths suitable for learning peer recommendations within the set length.

To reduce the impact of meta-paths with noisy and redundant information in $pathSet$ on the recommendation performance, we use information gain to identify meaningful meta-paths for learning peer recommendation.

Algorithm 1 illustrates the detailed pseudo code that describes the method for automatically generating all meaningful meta-paths. Please note that function 1 and function 2 are the same as those used by Lu [41]. In addition, the further difference is that our method can automatically generate meta-paths and utilize the information gained to select meaningful meta-paths, while the previous method only provides the ability to generate meta-paths.

Algorithm 1 Procedures of a method for automatically generating meaningful meta-paths.

Input: T_G=$(\mathcal{A}, \mathcal{R}, \mathcal{W})$: network schema of heterogeneous information network in adjacency matrix
A_s: source object type of a meta-path, where $A_s \in \mathcal{A}$
A_t: target object type of a meta-path, where $A_t \in \mathcal{A}$
max_Len: maximum length for meta-paths to be searched
Output: $pathSet$ = $\{P_1, P_2, \ldots\}$: set of meta-paths searched from T_G in form of A_s-*-A_t

$finalPathSet \leftarrow NULL$
$pathSet \leftarrow NULL$
for all $R_i \in \mathcal{R}, R_i = (A_{si}, A_{ti})$ **do**
  $fst\_Set \leftarrow list(A_{si})$
  for $i \leftarrow 1$ to max_Len **do**
    $snd\_Set \leftarrow next\_list(fst\_Set)$ //function 1
    $m\_Set \leftarrow merge\_list(fst\_Set, snd\_Set)$ //function 2
    $fst\_Set \leftarrow m\_Set$
    for $j \leftarrow 1$ to length(fst_Set) **do**
      if last element of $fst\_Set[j] ==$ A$_{ti}$ **then**
        $finalPathSet \leftarrow finalPathSet \cup fst\_Set[j]$
      end if
    end for
  end for
end for
for $k \leftarrow 1$ to length($finalPathSet$) **do**
  Calculate $p(m)$ using Formula (7)
  Calculate $H(M)$ using Formula (6)
  Calculate $p_P(m \mid n)$ using Formula (9)
  Calculate $H_P(M \mid N)$ using Formula (8)
  Calculate IG_P using Formula (5)
  Calculate $\widehat{IG_P}$ using Formula (10)
  if $\widehat{IG_P} > 0$ **then**
    $pathSet \leftarrow finalPathSet[k]$
  end if
end for

4.3. Recommendation Score Calculation Based on Weighted Meta-Paths

Tracking a student's history in terms of meta-paths can capture information about students' preferences in choosing learning peers. PathSim is used to measure the similarity between objects in the meta-path [14]. However, this method can only be applied to symmetric meta-paths and does not consider the effect of the degree of student–student interaction on the similarity measure between two students.

To overcome the above limitations, we use random walk on each differently weighted meta-path to calculate the corresponding node proximity between source objects and target objects [33]. Given the student u, the student query vector $Q(i)$ can be defined as Formula (11):

$$Q(i) = \begin{cases} 1 & if\, i = u \\ 0 & otherwise \end{cases}, \tag{11}$$

where $Q(i)$ is the $i - th$ term of Q. When $i = u$, student u is assigned a value of 1, while the others are all 0.

Once the target student is assigned a value, the value will be transferred to its neighbor objects on this meta-path. Specifically, the transition matrix $T_{MN} = \left[t_{ij}\right]_{p \times q}$ is described as a representation of transition probabilities between object types M and N, and can be determined by normalizing the adjacency matrix $A_{MN} = [t_{ij}]_{p\times q}$ by columns as shown in Formula (12):

$$t_{ij} = \frac{a_{ij}}{\sum\limits_{p'=0}^{p} a_{p'j}}, \tag{12}$$

where $\sum\limits_{p'=0}^{p} a_{p'j}$ is the sum of elements in the column j of $A_{MN} = [a_{ij}]_{p\times q}$.

Based on the above statement, for a given weighted meta-path $MP_k = \text{Student} \xrightarrow{\delta(R_1)} A_1 \xrightarrow{\delta(R_2)} \ldots \xrightarrow{\delta(R_{l-1})}$ Student, we can define the recommendation score vector as shown in Formula (13):

$$r_u^k = T_{S_S A_{l-1}} \cdot \ldots \cdot T_{A_1 S_T} \cdot Q(u), \tag{13}$$

where r_u^k represents each candidate student's recommendation score to target student u on the meta-path k, S_T represents the set of target students, S_S represents the set of candidate students, $T_{S_S A_{l-1}}$ is the transition matrix of candidate students to object type A_{l-1}, $T_{A_1 S_T}$ is the transition matrix of object type A_1 to target students.

4.4. Personalized Weight-Learning for Weighted Meta-Paths

For every student, we develop a linear function that integrates the recommendation scores between the target student and candidate students on all selected weighted meta-paths, therefore obtaining the global score vector as shown in Formula (14):

$$g_u = \sum_k \alpha_u^k \cdot r_u^k, \tag{14}$$

where g_u represents the global score vector of target student u, α_u^k denotes personalized weight of student u on weighted meta-path k, r_u^k denotes the recommendation score vector of student u on meta-path k.

Each meta-path in the weighted heterogeneous information network has exact semantic information, and the weight of the meta-path indicates how much students attach importance to the information on the meta-path. Different weighted meta-paths reflect different learning behaviors of students (such as student–student interaction behavior, student–teacher interaction behavior, and student–system interaction behavior). Through personalized weighted learning, a higher value can be assigned to the selected weighted meta-path that expresses students' preference for choosing a learning peer.

There is no need to pursue accurate recommendation scores of candidate students based on weighted meaningful meta-paths [33]. Therefore, to ensure that the recommendation scores of recommended students are higher than those of students that are not recommended, the BPR optimization framework is employed to build our objective function [42]. We consider students who are learning peers of a target student as a positive set, denoted as PS_u, and students who are not learning peers of the target student as a negative set, denoted as NS_u [43]. In addition, we outline the objective function for target student u as Formula (15):

$$\text{maxobj}_u(\alpha) = \frac{\sum\limits_{i\in PS_u} \sum\limits_{j\in NS_u} \sigma(g_u(i) - g_u(j))}{|PS_u||NS_u|}, \tag{15}$$

where PS_u denotes the positive set of student u, NS_u denotes the negative set of student u, $|PS_u|$ denotes the size of PS_u, $|NS_u|$ denotes the size of NS_u, $g_u(i)$ is the global recommendation score of candidate student i to target student u, $g_u(j)$ is the global recommendation score of candidate student j to target student u, and $\sigma(\bullet)$ is an indicator function, which can be expressed as Formula (16):

$$\sigma(x) = \frac{1}{1+e^{-x}}, \tag{16}$$

The gradient ascent (GA) algorithm can be applied to maximize the objective function for student u. For each student, we update the parameter α_u according to the ascending direction of the gradient, and the update process is shown in Formula (17):

$$\alpha_u^{(t+1)} := max(0, \alpha_u^{(t)} + lr \cdot \frac{\partial obj_u(\alpha)}{\partial \alpha}), \tag{17}$$

where lr denotes learning rate, $\alpha_{\mathrm{u}}^{(t)}$ and $\alpha_{\mathrm{u}}^{(t+1)}$ are the parameters of student u at time t and $t+1$. The gradient of $\mathrm{obj}_u(\alpha)$ to parameter α is calculated as Formula (18):

$$\frac{\partial obj_u(\alpha)}{\partial \alpha} = \frac{\sum\limits_{i \in PS_u} \sum\limits_{j \in NS_u} \frac{\partial \sigma(\Gamma_{ij})}{\partial \Gamma_{ij}} \left(\frac{\partial g_u(i)}{\partial \alpha} - \frac{\partial g_u(j)}{\partial \alpha} \right)}{|PS_u||NS_u|}, \tag{18}$$

where $\Gamma_{ij} = g_u(i) - g_u(j)$. We can have $\partial \sigma(\Gamma_{ij}) / \partial \Gamma_{ij} = e^{-(\Gamma_{ij})} / \left(1 + e^{-(\Gamma_{ij})}\right)^2$ by deriving the sigmoid function. The derivative $\partial g_u / \partial \alpha_u^k$ of each parameter α_u^k is calculated as Formula (19):

$$\frac{\partial g_u}{\partial \alpha_u^k} = r_u^k, \tag{19}$$

Please note that if the sum of all parameters α_u of student u is not equal to 1, these parameters are normalized to ensure the sum is 1. The specific procedure of personalized weight-learning is shown in Algorithm 2.

Algorithm 2 Personalized weight-learning for weighted meta-paths.

Input: student u, learning rate lr, threshold ξ
Output: personalized weight α_u

1: $t = 0$
2: Initialize $\alpha_u^{(0)}$
3: **while** $\alpha_u^{(t+1)} - \alpha_u^{(t)} > \xi$ **do**
4: Compute the recommendation ranking score $g_u(v)$ of each candidate student v
5: Compute $\frac{\partial g_u}{\partial \alpha_u^k}$ using Formula (19)
6: Update α based on Formula (17)
7: Update $t = t + 1$
8: **end while**

4.5. Personalized Learning Peer Recommendation Based on Weighted Meta-Paths

The basic principle of recommending learning peers is that the target students tend to be interested in interacting with other students who are connected through important weighted meta-paths that the target students emphasize [14]. We use a global recommendation score to incorporate recommendation scores between the target and candidate students through differently weighted meta-paths. The global recommendation score $g_u(v)$ is determined by the recommendation scores of the candidate student on each meta-path and the target student's corresponding personalized weight of the weighted meta-path, and can be expressed as Formula (20):

$$g_u(v) = \sum_k \alpha_u^k \cdot r_u^k(v), \tag{20}$$

For each target student, the Top-N recommendation list is generated based on the recommendation scores of all candidate students to become potential learning peers.

5. Experiments and Results Analysis

5.1. Datasets

We conducted our experiments on three real-world datasets. To demonstrate the ability of our method on other weighted heterogeneous information networks, we used two other datasets, namely the DBLP dataset (DBLP (https://www.aminer.cn/citation, accessed on 7 January 2022)) and Aminer dataset (Aminer (https://www.aminer.cn/data/#Academic-Social-Network, accessed on 17 December 2021)). Additionally, the Online dataset was used for learning peer recommendations. The relevant statistics of three datasets are shown in Table 2.

Table 2. Statistics of the experimental datasets.

Datasets	Objects	Number	Links	Number
DBLP	papers	23,607	paper-venues	23,607
	venues	1796	paper-authors	80,535
	authors	4524	-	-
Aminer	papers	16,358	paper-venues	16,358
	venues	3765	paper-authors	59,343
	authors	3925	paper-terms	81,790
	terms	10,928	-	-
Online	students	1055	videos-teachers	207
	videos	207	students-exercises	427,478
	teachers	1	students-videos	283,061
	exercises	163	students-knowledge points	310,090
	knowledge points	207	exercises-knowledge points	7505
	-	-	videos-knowledge points	207
	-	-	teachers-knowledge points	10,490

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 [37] is shown in Figure 5a, which covers author (A), paper (P), and conference venue (V), and two types of relationships, 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, to make use of the text information, We select five terms with the highest Term Frequency Inverse Document Frequency (TF-IDF) score from the title and abstract of each paper. The network schema for the Aminer dataset [44] is illustrated in Figure 5b, which contains author (A), paper (P), venue (V), and term (T). Please note that "1∼N" in Figure 5a,b represents the order of authors in a paper.

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 point (K), and their relationships with each other.

5.2. Evaluation Metrics

In each experiment, each dataset is divided into five partitions to obtain a training set and a test set. We use four partitions for training different recommendation approaches and the remaining partition for testing. After the training, we recommend learning peers for each target student.

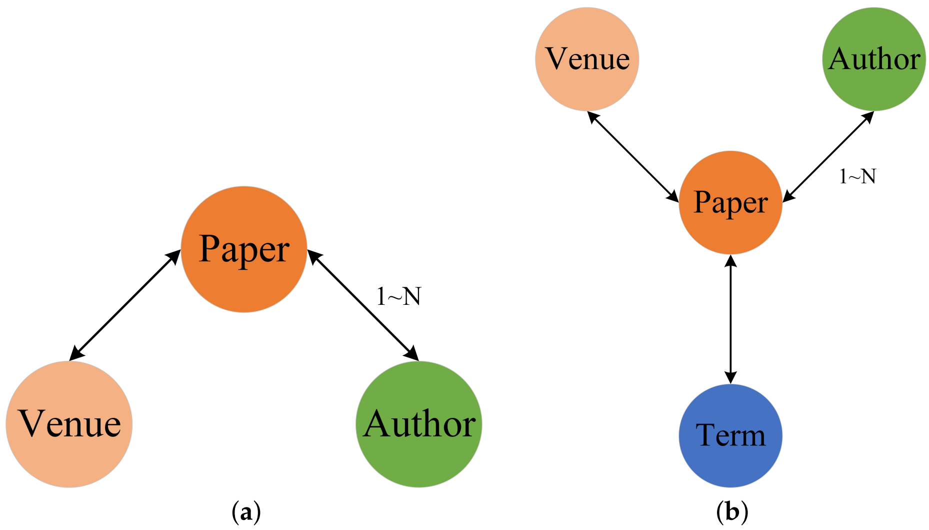


Figure 5. Network schema for different datasets. (**a**) Network schema for DBLP. (**b**) Network schema for Aminer.

5.2.1. Precision and Recall

If the recommended students communicate with the target students, the recommendation is valid [6]. In addition, there are two evaluation metrics used to evaluate performance, namely precision and recall. These two metrics can be calculated as follows:

$$\text{Precision} = \frac{|\operatorname{rec}(u) \bigcap \operatorname{real}(u)|}{|\operatorname{rec}(u)|}, \tag{21}$$

$$\text{Recall} = \frac{|\operatorname{rec}(u) \bigcap \operatorname{real}(u)|}{|\operatorname{real}(u)|}, \tag{22}$$

where $\operatorname{rec}(u)$ denotes the recommendation list for target student u, $\operatorname{real}(u)$ denotes the true learning peer set of target student u.

5.2.2. The Achievement Degree of Curriculum Objectives

To study the practical benefits of our method, we use the achievement degree of curriculum objectives in the process of engineering education certification to scientifically evaluate students' learning outcomes and effectively feed them back to teachers [45]. This method involves the concepts of assessment methods and curriculum objectives. Different assessment methods are used to test whether students have passed the course in all respects. Different curriculum objectives correspond to different engineering index points in the graduation requirements to determine whether students have met the graduation requirements of the course. The evaluation method is described in detail below.

There are m' assessment methods for a certain course, and the total score of each assessment method is S'. In addition, the course has n' curriculum objectives. Please note that the score of assessment method j corresponding to curriculum objective i is a_{ij}, and the average score of assessment method j corresponding to curriculum objective i of all students participating in the course is b_{ij}. The total achievement degree D of a group's curriculum objectives is defined as Formula (23):

$$D = \sum_{i=1}^{n'} D_i \cdot \frac{\sum_{j=1}^{m'} a_{ij}}{m' \cdot S'} \tag{23}$$

where D_i represents the achievement degree of the group's curriculum objective i. The calculation process can be expressed as Formula (24):

$$D_i = \sum_{j=1}^{m'} v_j \cdot c_{ij} \tag{24}$$

where v_j is the weight of assessment method j, and $\sum_{j=1}^{m'} v_j = 1$. c_{ij} represents the evaluation value of the assessment method j corresponding to the curriculum objective i of all students, and the calculation process is $c_{ij} = b_{ij}/a_{ij}$.

5.3. Baseline Methods

This paper compares LPRWHIN with the following baseline methods:

(1) Recommendation Based on the Meta-Path (Metapath) [14]: This method first integrates students' historical preference information into a heterogeneous information network and generates a meta-path set. Second, the similarity between students is calculated based on the meta-path, and the regularization-based optimization method is introduced to learn the personalized weight on the meta-path.
(2) Paper Recommendation Based on a Heterogeneous Information Network (PRHN) [33]: This method first constructs a heterogeneous information network containing multiple types of objects and relationships. Second, random walks are performed on each meta-path to obtain recommendation scores of candidate students, and personalized weight-learning is performed on each meta-path.
(3) Matrix Factorization (MF) [46]: This model maps the interaction between students into a joint latent factor space of dimensionality f, so that the interactions between students are modeled as inner products in this space.
(4) Deep Matrix Factorization (DMF) [47]: This method uses a neural network framework to project target students and candidate students to low-dimensional vectors. In the experiment, the interaction matrix between students is used as input.

5.4. Parameter Settings

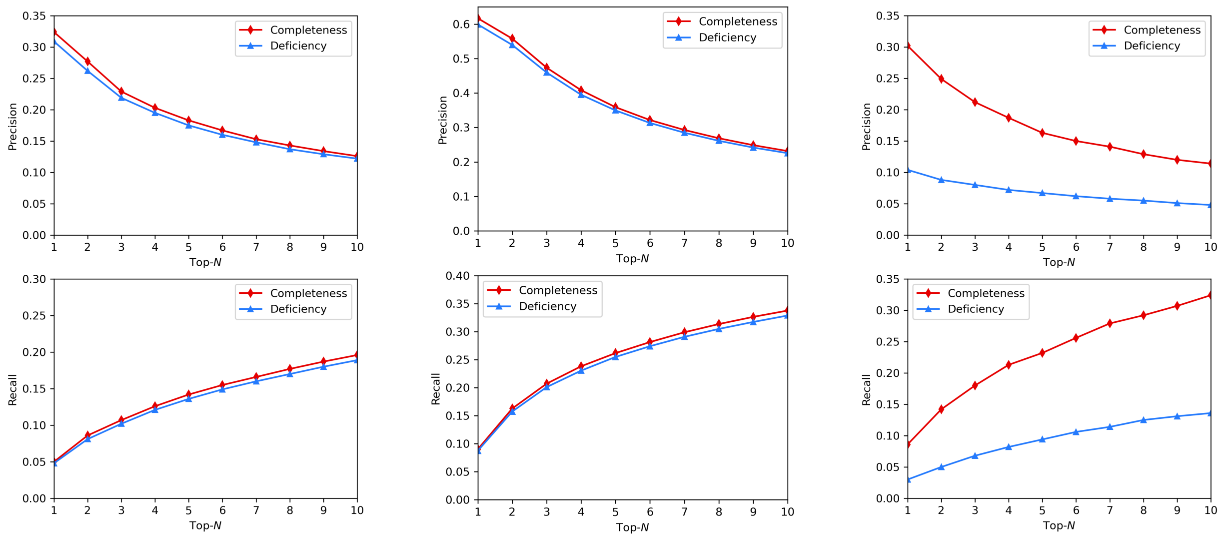
In the process of personalized weight-learning for weighted meta-paths, threshold ξ and learning rate lr have a greater impact on recommendation performance. In this experiment, we set ξ to be an infinitesimal random number and lr to 0.001 empirically.

5.5. Results of Comparative Experiments

5.5.1. Impact of Different Numbers of Meaningful Meta-Paths

To demonstrate the impact of generating a completed meaningful meta-path set on recommendation performance, the following experiments were conducted in this paper. First, we extracted completed meaningful meta-path sets with a maximum length of 4 on DBLP, Aminer, and Online datasets using a method for automatically generating meaningful meta-paths, respectively. Second, we simulated the manual enumeration method by reducing one meta-path in each of those completed meaningful meta-path sets. Third, we used the corresponding meta-path sets on these three different datasets for experiments. The experimental results are shown in Figure 6.

Figure 6a–c show the results of LPRWHIN under different numbers of meta-paths on DBLP, Aminer, and Online datasets. The experimental results show that using the complete meaningful meta-path set performs better. Therefore, a method for automatically generating meaningful meta-paths can achieve better precision and recall. Especially on the Online dataset, reducing a meaningful meta-path has a significant impact on the precision and recall. It indicates that a small number of meaningful meta-paths have a great impact on the recommendation performance, which is consistent with the conclusion in the literature [48].

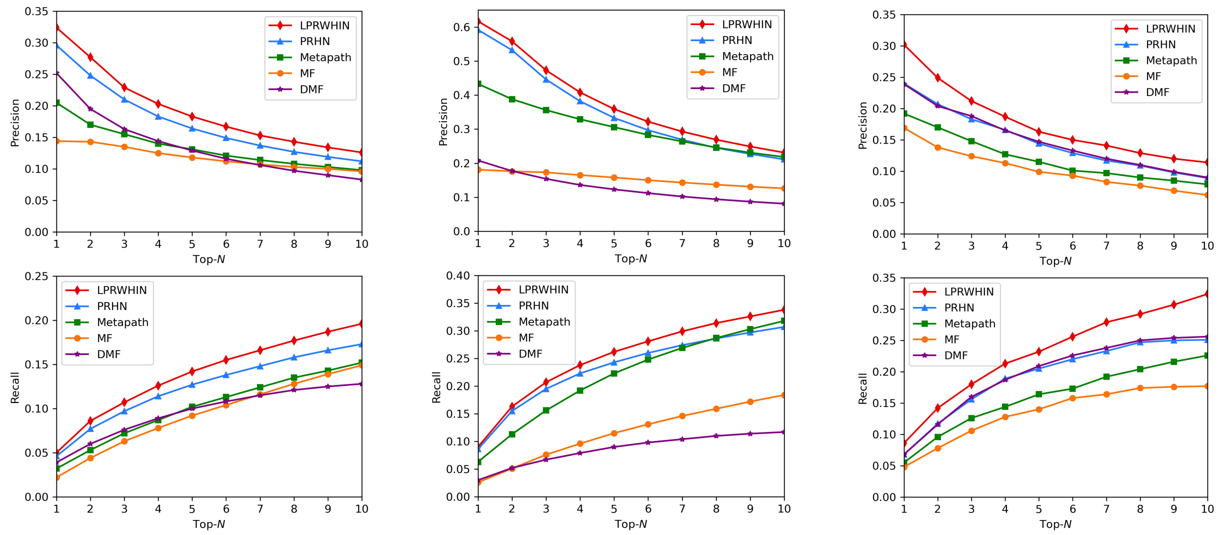


(**a**) (**b**) (**c**)

Figure 6. Results of LPRWHIN under different numbers of meta-paths on three datasets. (**a**) Results on DBLP. (**b**) Results on Aminer. (**c**) Results on Online.

5.5.2. Analysis of Experimental Results

We conducted experiments on three different datasets using LPRWHIN, PRHN, Metapath, MF, and DMF. The experimental results are shown in Figure 7, Tables 3 and 4.



(**a**) (**b**) (**c**)

Figure 7. Performance comparison of five recommendation methods on three datasets. (**a**) Results on DBLP. (**b**) Results on Aminer. (**c**) Results on Online.

Although the scale of the three datasets is different, the performance shows similar trends. Figure 7 and Tables 3 and 4 show that LPRWHIN outperforms other baseline methods in terms of precision and recall.

As can be seen from Tables 3 and 4, the precision decreases with increasing recommendation list length, while the recall increases with increasing recommendation list length. Figure 7a shows the recommendation performance of LPRWHIN and other baseline methods on the DBLP dataset. It illustrates that LPRWHIN performs the best, and the highest precision of LPRWHIN is up to 32.4% when N is 1. Figure 7b shows the recommendation performance of five methods on the Aminer dataset. It demonstrates that LPRWHIN has the highest precision of 61.7%, which indicates that LPRWHIN has a 61.7% probability of recommending co-authors in the Top-1 recommendation list. In addition, it shows that the recall of LPRWHIN is 33.8% when N is 10, which indicates that there are 33.8% true co-authors in the Top-10 recommendation list. Figure 7c shows the precision and recall of five methods on the Online dataset. As shown in Figure 7c, the highest precision of LPRWHIN reaches 30.2% when N is 1, which is 6.2% higher than the second-best method PRHN. The highest recall of LPRWHIN reaches 32.4% when N is 10, which is 7.3% higher than the second-best method PRHN.

Table 3. Results of five recommendation methods on different datasets-precision.

DBLP										
Recommendation methods	Pre@1	Pre@2	Pre@3	Pre@4	Pre@5	Pre@6	Pre@7	Pre@8	Pre@9	Pre@10
LPRWHIN	0.324	0.277	0.229	0.203	0.183	0.167	0.153	0.143	0.134	0.126
PRHN	0.296	0.248	0.210	0.183	0.164	0.149	0.137	0.127	0.119	0.112
Metapath	0.205	0.170	0.155	0.140	0.131	0.122	0.114	0.108	0.103	0.098
MF	0.144	0.143	0.135	0.125	0.118	0.112	0.107	0.103	0.100	0.096
DMF	0.252	0.195	0.163	0.144	0.129	0.116	0.106	0.097	0.090	0.083
Aminer										
Recommendation methods	Pre@1	Pre@2	Pre@3	Pre@4	Pre@5	Pre@6	Pre@7	Pre@8	Pre@9	Pre@10
LPRWHIN	0.617	0.558	0.473	0.408	0.359	0.322	0.293	0.269	0.249	0.231
PRHN	0.592	0.532	0.446	0.382	0.333	0.297	0.269	0.245	0.227	0.211
Metapath	0.433	0.388	0.356	0.329	0.306	0.283	0.264	0.246	0.231	0.218
MF	0.181	0.176	0.173	0.165	0.158	0.150	0.143	0.137	0.131	0.126
DMF	0.208	0.177	0.154	0.136	0.123	0.112	0.102	0.094	0.087	0.081
Online										
Recommendation methods	Pre@1	Pre@2	Pre@3	Pre@4	Pre@5	Pre@6	Pre@7	Pre@8	Pre@9	Pre@10
LPRWHIN	0.302	0.249	0.212	0.187	0.163	0.150	0.141	0.129	0.120	0.114
PRHN	0.240	0.207	0.183	0.166	0.144	0.129	0.117	0.109	0.098	0.089
Metapath	0.192	0.170	0.148	0.127	0.115	0.101	0.097	0.090	0.085	0.079
MF	0.169	0.138	0.124	0.113	0.099	0.093	0.083	0.077	0.069	0.062
DMF	0.239	0.204	0.188	0.165	0.147	0.133	0.120	0.110	0.099	0.090

In conclusion, the proposed method LPRWHIN significantly outperforms other baseline methods in terms of precision and recall on three different datasets. Compared with PRHN, LPRWHIN utilizes an algorithm to automatically generate meaningful meta-paths to ensure the completeness of the meta-path set, and all the weights are assigned to meaningful meta-paths. In addition, the effect of attribute values on links is also considered. The recommendation performance of Metapath is inferior to that of LPRWHIN because Metapath uses a method of manually generating meta-paths and the similarity measure used is only suitable for symmetric meta-paths as well as does not consider the effect of attribute values on links. MF has the worse performance because it only considers the interaction behavior between students and does not consider other behavior data in the learning process. DMF's performance is unstable on different datasets and performs worse than expected because it only considers the interaction behavior between students and requires large-scale data.

Table 4. Results of five recommendation methods on different datasets-recall.

DBLP										
Recommendation methods	Rec@1	Rec@2	Rec@3	Rec@4	Rec@5	Rec@6	Rec@7	Rec@8	Rec@9	Rec@10
LPRWHIN	0.050	0.086	0.107	0.126	0.142	0.155	0.166	0.177	0.187	0.196
PRHN	0.046	0.077	0.097	0.114	0.127	0.138	0.148	0.158	0.166	0.173
Metapath	0.032	0.053	0.072	0.087	0.102	0.113	0.124	0.135	0.143	0.152
MF	0.022	0.044	0.063	0.078	0.092	0.104	0.116	0.128	0.139	0.149
DMF	0.039	0.060	0.076	0.089	0.100	0.108	0.115	0.121	0.125	0.128
Aminer										
Recommendation methods	Rec@1	Rec@2	Rec@3	Rec@4	Rec@5	Rec@6	Rec@7	Rec@8	Rec@9	Rec@10
LPRWHIN	0.090	0.163	0.207	0.238	0.262	0.281	0.299	0.314	0.326	0.338
PRHN	0.086	0.155	0.195	0.223	0.243	0.260	0.274	0.286	0.297	0.307
Metapath	0.063	0.113	0.156	0.192	0.223	0.248	0.269	0.287	0.303	0.318
MF	0.026	0.051	0.076	0.096	0.115	0.131	0.146	0.159	0.172	0.184
DMF	0.030	0.052	0.067	0.079	0.090	0.098	0.104	0.110	0.114	0.117
Online										
Recommendation methods	Rec@1	Rec@2	Rec@3	Rec@4	Rec@5	Rec@6	Rec@7	Rec@8	Rec@9	Rec@10
LPRWHIN	0.086	0.142	0.180	0.213	0.232	0.256	0.279	0.292	0.307	0.324
PRHN	0.068	0.117	0.156	0.189	0.205	0.220	0.233	0.247	0.250	0.251
Metapath	0.055	0.096	0.126	0.144	0.164	0.173	0.192	0.204	0.216	0.226
MF	0.048	0.078	0.106	0.128	0.140	0.158	0.164	0.174	0.176	0.177
DMF	0.068	0.116	0.160	0.187	0.209	0.226	0.238	0.250	0.254	0.256

5.6. Case Study

We have conducted case studies on students in grade 2020 to evaluate the practical benefits of this study in the online learning process. The course “Data Structure and Algorithm” we selected has three assessment methods and four curriculum objectives. The assessment methods of this course are composed of final examinations, experimental reports, and homework. The total score of each assessment method is 100, and the corresponding weights are 0.7, 0.2, and 0.1, respectively. In addition, the course has four curriculum objectives, of which curriculum objective 1 mainly examines the application ability of students to basic theories and methods in data structure. Curriculum objective 2 mainly examines students’ ability to accurately describe the process of dealing with complex engineering problems. Curriculum objective 3 mainly examines students’ abilities to comprehensively consider various technical requirements and design modules and solutions to meet specific needs. Curriculum objective 4 mainly examines the ability of students to conduct system simulation and optimization research through correlation analysis of relevant complex engineering problems. Table 5 shows the score relationship between curriculum objectives and assessment methods.

Table 5. The score relationship between curriculum objectives and assessment methods.

	Assessment Method 1	Assessment Method 2	Assessment Method 3	Target Score	Equivalent Score
Curriculum objective 1	20	20	20	60	20
Curriculum objective 2	20	20	20	60	20
Curriculum objective 3	30	30	30	90	30
Curriculum objective 4	30	30	30	90	30
Total	100	100	100	300	100

5.6.1. Comparison of the Total Achievement Degree of Curriculum Objectives

We divide the students in grade 2020 into three groups: students who become learning peers according to the recommendation results (Group 1), students who do not become learning peers according to the recommendation results (Group 2), and all students (Group 3). In engineering education certification, it is generally believed that the expected value of the achievement degree of curriculum objectives is 0.7, which indicates that the student learning outcomes meet the expectations. Based on this consensus, we analyzed the total achievement degree of the curriculum objectives of Group 1, Group 2, and Group 3. The experimental results of the total achievement degree of curriculum objectives for different groups are shown in Figure 8.

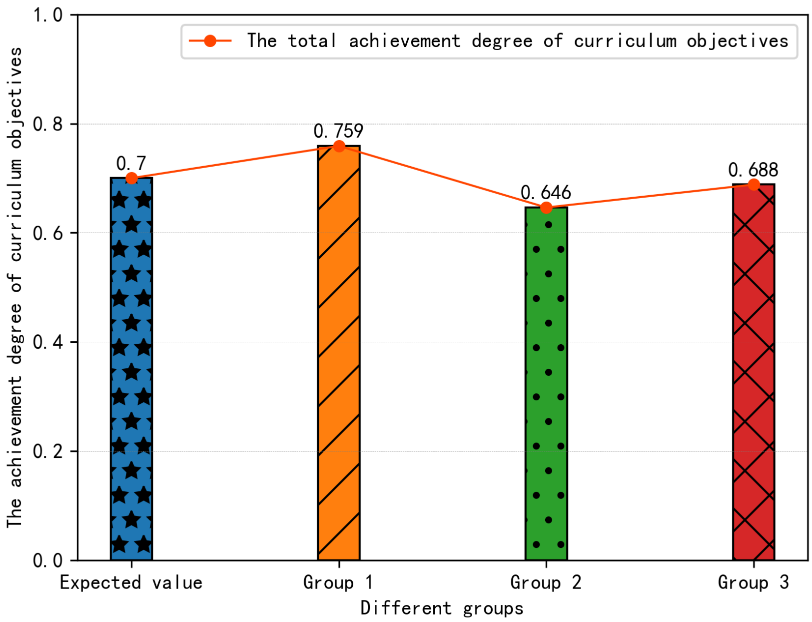


Figure 8. Comparison of the total achievement degree of curriculum objectives for different groups.

From Figure 8, it can be observed that the total achievement degree of curriculum objectives in Group 3 is 0.688, which is composed of Group 1 and Group 2. The total achievement degree of curriculum objectives in Group 1 is greater than that in Group 2 and exceeds the expected value, which makes a great contribution to the improvement of the total achievement degree of curriculum objectives in Group 3. However, the total achievement degree of curriculum objectives in Group 2 is lower than 0.7, indicating that the completion of this group's course has not met expectations. It can be concluded that Group 1 can complete the course well and achieve the expected objectives, indicating that the proposed method can encourage students to keep learning and help them pass the course.

5.6.2. Comparison of the Achievement Degree of the Four Curriculum Objectives

We analyzed the achievement degree of the four curriculum objectives for different groups, and the comparative results are shown in Figure 9 and Table 6.

Table 6. Comparative results of the achievement degree of the four curriculum objectives.

	Curriculum Objective 1	Curriculum Objective 2	Curriculum Objective 3	Curriculum Objective 4
Expected values	0.700	0.700	0.700	0.700
Group 1	0.781	0.766	0.740	0.760
Group 2	0.683	0.680	0.614	0.631

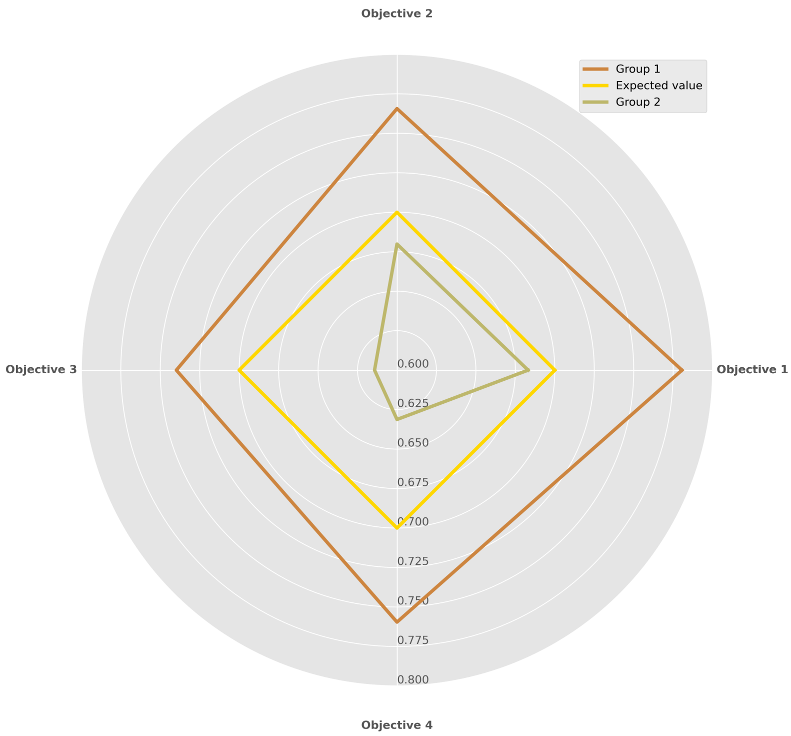


Figure 9. Results of the achievement degree of the four curriculum objectives.

It can be seen from Figure 9 and Table 6 that the achievement degree of the four curriculum objectives in Group 1 is greater than that in Group 2. The achievement degree of the four curriculum objectives in Group 1 exceeds the expected value, and the achievement degree of curriculum objective 1 is the largest, which is 0.781. The achievement degree of curriculum objective 3 is the smallest, 0.740. On the contrary, the achievement degree of the four curriculum objectives in Group 2 is lower than the expected value, but the achievement degrees of curriculum objective 1 and curriculum objective 2 are close to 0.7, namely 0.683 and 0.680, respectively.

In groups 1 and 2, the achievement degree of curriculum objective 1 is the highest, and the achievement degree of curriculum objective 3 is the lowest, which indicates that students have a strong ability to apply the basic theories and methods of data structure, but a weak ability to accurately describe the processing process of complex engineering problems. Based on the analysis of the achievement degree of curriculum objectives and the historical data, it can be concluded that there are two reasons for this situation:

1. When some students in Group 1 encounter difficulties, there is less interaction between students and teachers, resulting in insufficient learning of the corresponding difficult knowledge points.
2. Most of the students in Group 2 only have little interaction, and even many students do not complete the learning of course knowledge points.

Based on the analysis of the total achievement degree of curriculum objectives and achievement degrees of the four curriculum objectives, we can conclude that the learning quality of Group 2 is better. It can be shown that the proposed method can improve students' participation in the course and help students better learn relevant knowledge, which is of great significance for students to successfully pass the course and meet the graduation requirements.

6. Conclusions

In this paper, we propose a learning peer-recommendation method based on a weighted heterogeneous information network to increase student participation in online courses. First, we use a weighted heterogeneous information network to incorporate the multiple types of objects, relationships, and attribute values. Second, we propose a method for automatically generating all meaningful weighted meta-paths, which can effectively generate meaningful meta-paths for learning peer recommendations. Finally, this study employs an optimization process using Bayesian Personalized Ranking to learn the personalized weights on different meta-paths and calculate the recommendation score for candidate students. In addition, the method fuses heterogeneous information in the learning process and provides interpretability for the recommendation results from three perspectives. First, the method analyzes the effect of attribute values on links, which not only reveals what objects with which target students and candidate students commonly interact, but also indicates how much workload they devote to interacting with these objects. Second, the weighted meta-path can reflect students' preferences. The selected meaningful weighted meta-paths reflect what information target students care about when choosing learning peers. Third, personalized weights for each meaningful meta-path are assigned via personalized weight-learning, reflecting the extent to which target students care about the information when choosing their learning peers. In this paper, we compare the proposed method LPRWHIN with several baseline methods on three real-world datasets. Experimental results indicate that LPRWHIN outperforms other baseline methods in terms of precision and recall. Moreover, we use the achievement degree evaluation system based on the concept of outcome-based education to reasonably evaluate the student learning outcomes. The final evaluation results can not only give feedback about the problems existing in the learning process to students, but also help teachers find the deficiencies in their teaching and provide suggestions for follow-up teaching.

However, there are some limitations to our research. First, this study relies on students' historical behavioral preferences, and data sparsity affects recommendation performance. Second, this study only focuses on the information relating to objects and the attribute values on links but does not consider the specific text content involved in the interaction between students and teachers.

Future research may go in three directions. First, as the learning process continues, interactions between students will constantly evolve. Therefore, future research can implement a dynamic recommendation model using incremental computing in heterogeneous information networks to make adaptive recommendations. Second, broad learning has attracted much attention for its ability to overcome deep learning's complex network structure and numerous parameters. Future research may integrate heterogeneous information networks and broad learning to discover more fine-grained student preference patterns and greatly improve recommendation performance. Third, real-world feedback from students is vital for improving recommendation methods. Future research should pay more attention to students' responses to recommendation results.

Author Contributions: Conceptualization, Z.S. (Zhaoyu Shou) and Z.S. (Zhixuan Shi); methodology, Z.S. (Zhaoyu Shou) and Z.S. (Zhixuan Shi); software, Z.S. (Zhixuan Shi) and J.L.; validation, Z.S. (Zhaoyu Shou), Z.S. (Zhixuan Shi) and H.W.; formal analysis, Z.S. (Zhixuan Shi); investigation, Z.S. (Zhaoyu Shou) and Z.S. (Zhixuan Shi); resources, Z.S. (Zhaoyu Shou); data curation, Z.S. (Zhaoyu Shou), Z.S. (Zhixuan Shi) and H.W.; writing—original draft preparation, Z.S. (Zhixuan Shi); writing—review and editing, Z.S. (Zhaoyu Shou), Z.S. (Zhixuan Shi) and J.L.; visualization, Z.S. (Zhixuan Shi); supervision, Z.S. (Zhaoyu Shou) and H.W.; project administration, Z.S. (Zhaoyu Shou), Z.S. (Zhixuan Shi), H.W. and H.Z.; funding acquisition, Z.S. (Zhaoyu Shou). All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by The National Natural Science Foundation of China (62177012, 61967005, 62267003), Innovation Project of GUET Graduate Education(2020YCXS022, 2021YCXS033), the Key Laboratory of Cognitive Radio and Information Processing Ministry of Education (CRKL190107).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data of the DBLP dataset and Aminer dataset supporting the study can be available at https://www.aminer.cn/citation (accessed on 7 January 2022) and https://www.aminer.cn/data/#Academic-Social-Network (accessed on 17 December 2021). However, the Online dataset is not suitable for direct publishing because it involves the personal privacy of students.

Acknowledgments: We are grateful to anonymous reviewers for their helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BPR	Bayesian Personalized Ranking	An Optimization Framework is Employed to Build the Objective Function
COVID-19	Corona Virus Disease 2019	Respiratory Diseases Caused by the Corona Virus
LPR	Learning Peer Recommendation	A Method for Learning Peer Recommendation
CNN	Convolutional Neural Network	A Kind of Feedforward Neural Network with Convolution Calculation and Deep Structure
RiPPLE	Open-Source Course-Level Recommendation Platform	An Online Learning Platform for Anyone
LPRWHIN	Learning Peer Recommendation Method Based on a Weighted Heterogeneous Information Network	Our Proposed Methodology for Learning Peer Recommendation
MOOC	Massive Open Online Courses	Free Online Courses are Available for Anyone
DNN	Deep Neural Networks	A Technology in the Field of Machine Learning
Online	Online Learning Platform Dataset	A Dataset for Learning Peer Recommendation
DBLP	DBLP Dataset	A Dataset for Evaluating Performance
Aminer	Aminer Dataset	A Dataset for Evaluating Performance
TF-IDF	Term Frequency Inverse Document Frequency	A Commonly Used Weighting Technique for Information Retrieval and Data Mining
Metapath	Recommendation Based on the Meta-Path	A Method for Scholar-Friend Recommendation
PRHN	Paper Recommendation Based on a Heterogeneous Information Network	A Method for Paper Recommendation
MF	Matrix Factorization	The Method of Decomposing a Matrix into the Products of Several Matrices
DMF	Deep Matrix Factorization	A Deep Matrix Factorization Model for Recommender Systems

References

1. Butnaru, G.I.; Niță, V.; Anichiti, A.; Brînză, G. The effectiveness of online education during covid 19 pandemic—A comparative analysis between the perceptions of academic students and high school students from romania. *Sustainability* **2021**, *13*, 5311. [CrossRef]
2. Runtian, Z. The Game and Integration between Online Education and Traditional School Education. In Proceedings of the 2020 11th International Conference on E-Education, E-Business, E-Management, and E-Learning, Osaka, Japan, 10–12 January 2020; pp. 121–124. [CrossRef]
3. Bai, Y.; Li, Z.; Zhao, F.; Jiang, Y. Strategies, methods and problems of online education in China during the epidemic. In Proceedings of the 2020 Ninth International Conference of Educational Innovation through Technology (EITT), Porto, Portugal, 13–17 December 2020; pp. 30–34. [CrossRef]
4. Chang, J. Online Instruction and Offline Classroom Teaching: A Study on Parallel Education Systems. In Proceedings of the 2021 9th International Conference on Information and Education Technology (ICIET), Okayama, Japan, 7–29 March 2021; pp. 157–160. [CrossRef]
5. Mrhar, K.; Douimi, O.; Abik, M. A dropout predictor system in MOOCs based on neural networks. *J. Autom. Mob. Robot. Intell. Syst.* **2021**, *42*, 72–80.
6. Xu, B.; Yang, D. Study partners recommendation for xMOOCs learners. *Comput. Intell. Neurosci.* **2015**, *2015*, 15.
7. Hu, Q.; Han, Z.; Lin, X.; Huang, Q.; Zhang, X. Learning peer recommendation using attention-driven CNN with interaction tripartite graph. *Inf. Sci.* **2019**, *479*, 231–249. [CrossRef]
8. Bouchet, F.; Labarthe, H.; Yacef, K.; Bachelet, R. Comparing peer recommendation strategies in a MOOC. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, Bratislava, Slovakia, 9–12 July 2017; pp. 129–134. [CrossRef]
9. Yang, D.; Wen, M.; Rose, C. Peer influence on attrition in massively open online courses. In Proceedings of the Educational data mining 2014, London, UK, 4–7 July 2014.

10. Bouchet, F.; Labarthe, H.; Bachelet, R.; Yacef, K. Who wants to chat on a MOOC? Lessons from a peer recommender system. In Proceedings of the European Conference on Massive Open Online Courses, Madrid, Spain, 22–26 May 2017; Springer: Cham, Switzerland, 2017; pp. 150–159. [CrossRef]
11. Potts, B.A.; Khosravi, H.; Reidsema, C.; Bakharia, A.; Belonogoff, M.; Fleming, M. Reciprocal peer recommendation for learning purposes. In Proceedings of the 8th international conference on learning analytics and knowledge, Tempe, AZ, USA, 4–8 March 2019; pp. 226–235. [CrossRef]
12. Prabhakar, S.; Spanakis, G.; Zaiane, O. Reciprocal recommender system for learners in massive open online courses (MOOCs). In Proceedings of the International Conference on Web-Based Learning, Cape Town, South Africa, 20–22 September 2017; Springer: Cham, Switzerland, 2017; pp. 157–167. [CrossRef]
13. Shi, C.; Li, Y.; Zhang, J.; Sun, Y.; Philip, S.Y. A survey of heterogeneous information network analysis. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 17–37. [CrossRef]
14. Xu, Y.; Zhou, D.; Ma, J. Scholar-friend recommendation in online academic communities: An approach based on heterogeneous network. *Decis. Support Syst.* **2019**, *119*, 1–13. [CrossRef]
15. Liu, X.; Wu, K.; Liu, B.; Qian, R. HNERec: Scientific collaborator recommendation model based on heterogeneous network embedding. *Inf. Process. Manag.* **2023**, *60*, 103253. [CrossRef]
16. Karnyoto, A.S.; Sun, C.; Liu, B.; Wang, X. Augmentation and heterogeneous graph neural network for AAAI2021-COVID-19 fake news detection. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 2033–2043. [CrossRef]
17. Wu, Z.; Liang, Q.; Zhan, Z. Course Recommendation Based on Enhancement of Meta-Path Embedding in Heterogeneous Graph. *Appl. Sci.* **2023**, *13*, 2404. [CrossRef]
18. Lops, P.; Gemmis, M.d.; Semeraro, G. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*; Springer, 2011; pp. 73–105. [CrossRef]
19. Huang, R.; Lu, R. Research on Content-based MOOC Recommender Model. In Proceedings of the 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 10–12 November 2018; pp. 676–681. [CrossRef]
20. Campos, R.; dos Santos, R.P.; Oliveira, J. A Recommendation System based on Knowledge Gap Identification in MOOCs Ecosystems. In Proceedings of the XVI Brazilian Symposium on Information Systems, São Bernardo do Campo, Brazil, 3–6 November 2020; pp. 1–8. [CrossRef]
21. Pang, Y.; Wang, N.; Zhang, Y.; Jin, Y.; Ji, W.; Tan, W. Prerequisite-related MOOC recommendation on learning path locating. *Comput. Social Netw.* **2019**, *6*, 1–16. [CrossRef]
22. Zhao, X.; Liu, B. Application of Personalized Recommendation Technology in MOOC System. In Proceedings of the 2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Vientiane, Laos, 11–12 January 2020; pp. 720–723. [CrossRef]
23. Çano, E.; Morisio, M. Hybrid recommender systems: A systematic literature review. *Intell. Data Anal.* **2017**, *21*, 1487–1524. [CrossRef]
24. Safarov, F.; Kutlimuratov, A.; Abdusalomov, A.B.; Nasimov, R.; Cho, Y.I. Deep Learning Recommendations of E-Education Based on Clustering and Sequence. *Electronics* **2023**, *12*, 809. [CrossRef]
25. Wu, L. Collaborative filtering recommendation algorithm for MOOC resources based on deep learning. *Complexity* **2021**, *2021*. [CrossRef]
26. Liu, J.; Yin, C.; Li, Y.; Sun, H.; Zhou, H. Deep Learning and Collaborative Filtering-Based Methods for Students' Performance Prediction and Course Recommendation. *Wirel. Commun. Mobile Comput.* **2021**, *2021*, 2157343. [CrossRef]
27. Gong, J.; Wang, S.; Wang, J.; Feng, W.; Peng, H.; Tang, J.; Yu, P.S. Attentional graph convolutional networks for knowledge concept recommendation in moocs in a heterogeneous view. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual, 25–30 July 2020; pp. 79–88. [CrossRef]
28. Uddin, I.; Imran, A.S.; Muhammad, K.; Fayyaz, N.; Sajjad, M. A systematic mapping review on MOOC recommender systems. *IEEE Access* **2021**, 9, 118379–118405. [CrossRef]
29. Urdaneta-Ponte, M.C.; Mendez-Zorrilla, A.; Oleagordia-Ruiz, I. Recommendation Systems for Education: Systematic Review. *Electronics* **2021**, *10*, 1611. [CrossRef]
30. Li, Y.; Liu, J.; Ren, J. Social recommendation model based on user interaction in complex social networks. *PLoS ONE* **2019**, *14*, e0218957. [CrossRef]
31. Ma, X.; Dong, L.; Wang, Y.; Li, Y.; Liu, Z.; Zhang, H. An enhanced attentive implicit relation embedding for social recommendation. *Data Knowl. Eng.* **2023**, *145*, 102142. [CrossRef]
32. Paleti, L.; Radha Krishna, P.; Murthy, J. User opinions driven social recommendation system. *Int. J. Knowl.-Based Intell. Eng. Syst.* **2021**, *25*, 21–31. [CrossRef]
33. Li, Y.; Wang, R.; Nan, G.; Li, D.; Li, M. A personalized paper recommendation method considering diverse user preferences. *Decis. Support Syst.* **2021**, *146*, 113546. [CrossRef]
34. Li, H.; Li, J.; Li, G.; Gao, R.; Yan, L. Expert Recommendation in Community Question Answering via Heterogeneous Content Network Embedding. *Comput. Mater. Contin.* **2023**, *75*, 1687–1709. [CrossRef]
35. Shou, Z.; Wen, Y.; Chen, P.; Zhang, H. Personalized Knowledge Map Recommendations based on Interactive Behavior Preferences. *Int. J. Perform. Eng.* **2021**, *17, 36–49.*

36. Zhu, H.; Liu, Y.; Tian, F.; Ni, Y.; Wu, K.; Chen, Y.; Zheng, Q. A cross-curriculum video recommendation algorithm based on a video-associated knowledge map. *IEEE Access* **2018**, *6*, 57562–57571. [CrossRef]
37. Shi, C.; Zhang, Z.; Ji, Y.; Wang, W.; Yu, P.S.; Shi, Z. SemRec: A personalized semantic recommendation method based on weighted heterogeneous information networks. *World Wide Web* **2019**, *22*, 153–184. [CrossRef]
38. Yang, C.; Zhao, C.; Wang, H.; Qiu, R.; Li, Y.; Mu, K. A semantic path-based similarity measure for weighted heterogeneous information networks. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Changchun, China, 17–19 August 2018; Springer: Cham, Switzerland, 2018; pp. 311–323. [CrossRef]
39. Lu, M.; Wei, X.; Ye, D.; Dai, Y. A unified link prediction framework for predicting arbitrary relations in heterogeneous academic networks. *IEEE Access* **2019**, *7*, 124967–124987. [CrossRef]
40. Vahedian, F.; Burke, R.; Mobasher, B. Multirelational recommendation in heterogeneous networks. *ACM Trans. Web* **2017**, *11*, 1–34. [CrossRef]
41. Meilian, L.; Danna, Y. HIN_DRL: A random walk based dynamic network representation learning method for heterogeneous information networks. *Expert Syst. Appl.* **2020**, *158*, 113427. [CrossRef]
42. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Schmidt-Thieme, L. BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, 18–21 June 2009; pp. 452–461.
43. Pham, T.A.N.; Li, X.; Cong, G.; Zhang, Z. A general recommendation model for heterogeneous networks. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 3140–3153. [CrossRef] [CrossRef]
44. Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; Su, Z. Arnetminer: Extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008; pp. 990–998. [CrossRef]
45. Katawazai, R. Implementing outcome-based education and student-centered learning in Afghan public universities: The current practices and challenges. *Heliyon* **2021**, *7*, e07076. [CrossRef]
46. Koren, Y.; Bell, R.; Volinsky, C. Matrix factorization techniques for recommender systems. *Computer* **2009**, *42*, 30–37. [CrossRef]
47. Xue, H.J.; Dai, X.; Zhang, J.; Huang, S.; Chen, J. Deep matrix factorization models for recommender systems. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), Melbourne, VIC, Australia, 19–25 August 2017; Volume 17, pp. 3203–3209. [CrossRef]
48. Shi, C.; Hu, B.; Zhao, W.X.; Philip, S.Y. Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 357–370. [CrossRef]