



Metacognition-driven user-to-project recommendation for online education services

Ze Zheng Wu¹ · Haonan Jiang¹ · Xinghe Cheng² · Haotian Huang¹ · Qing Yang² · Jingwei Zhang¹

Received: 9 October 2022 / Revised: 30 December 2022 / Accepted: 14 January 2023 /
Published online: 22 February 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

Various online learning platforms have accumulated a large number of users who are learning or have completed their studies. In which, online education mostly provides services for theoretical learning. A vital issue is how to bridge theoretical learning and practical projects for improving users' competence. To address this issue, we propose a novel **Metacognition-driven User-to-Project** recommendation approach for online **Education Services (MUP-ES)**. We closely model theoretical content and practical project by constructing **Heterogeneous Information Networks (HINs)** covering both user learning outcomes (learned courses, acquired knowledge concepts, etc.) and project knowledge requirements. We design an attention-based metacognitive information aggregation process to efficiently refine the structural and semantic information of HINs and match users to appropriate projects. To embed HINs, we design a meta-path based random walk strategy to generate meaningful node sequences. MUP-ES provides two major components, path filtering and information aggregation. The path filtering module filters the impurity information in the path through the attention mechanism, and enhances the weight of user learning information that is more suitable for the current project. The information aggregation module learns the neighborhood information of the nodes in the path through the multi-head self-attention mechanism, which aggregates the rich user-related information to the project nodes. Finally, MUP-ES predicts the project's match score with real users and tests the model's ability to mitigate the cold-start problem. Extensive experiments on two real-world education datasets show that MUP-ES achieves more accurate prediction results than state-of-the-art baselines.

Keywords Online education services · Recommender systems · Heterogeneous information networks · Multi-head self-attention

Ze Zheng Wu, Haonan Jiang, Xinghe Cheng, Haotian Huang, Qing Yang and Jingwei Zhang ares contributed equally to this work.

✉ Qing Yang
gtyqing@hotmail.com

Extended author information available on the last page of the article.

1 Introduction

Online education applications are developing towards personalization, intelligence and convenience. On this basis, multiple online education platforms have been developed, such as **Massive Open Online Courses (MOOCs)**, online teaching is becoming an emerging education model in the world. With the increase of users on the platform, a large amount of user interaction data helps researchers to better conduct educational data mining. Their work includes knowledge concept recommendation [1, 2], course recommendations [3, 4], student dropout or procrastination behavior [5–7], etc.

To this end, previous studies have mainly focused on providing theoretical assistance to users before or during the learning process. In order to expand the dimension of online education services and help users who have completed their studies to go to the next step. We innovatively focus on the combination of theoretical learning and practical projects, and recommend users to projects that suit them through a deep learning framework. Completing this new challenge can not only matches users to suitable projects, but also allows users to consolidate their knowledge from projects and improve learning efficiency.

In this work, we start by constructing HINs based on user learning information and project knowledge requirements. As shown in Figure 1, in the data generated by the MOOCs platform [8], we observed that there are multiple types of relationship instances. To model and exploit this heterogeneous and complex information, researchers propose to use graph data structures to represent entities and their relationships with nodes and edges. Examples include relationship diagrams in social networks [9], knowledge graphs [10, 11], recommender systems [12, 13], etc. In this field of research, information mining based on HINs is a promising direction, which can display various types of data clearly and flexibly. The basic idea of most existing HINs based recommendation methods utilizes meta-paths to capture information semantics for recommendation in HINs. [14–16]. Although the meta-path based embedding methods have achieved performance improvements to a certain extent, there are still some problems in the recommendation task. First, the connection delivery of user information and project information relies on meta-paths. However, the design of meta-paths is artificial, and it is difficult for researchers to judge what types of meta-paths are

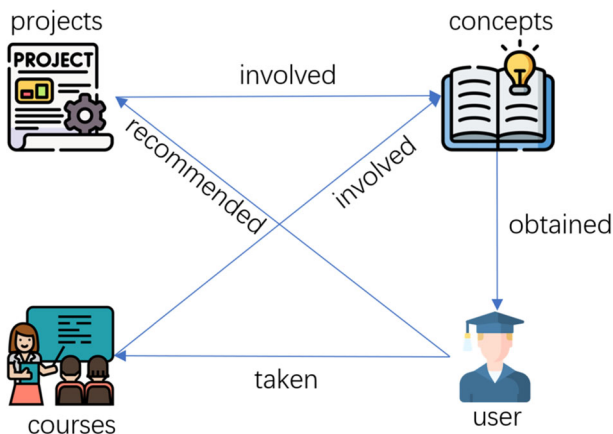


Figure 1 The infographic presents the entities and their relationships in the MOOCube dataset

paths with valuable information, so paths without valuable information are usually introduced in the work, which increases useless computation[17]. Second, the method based on meta-path rarely considers the node information features inside the path [13, 18, 19], some only consider the two endpoints of the path and ignore intermediate nodes [13], these practices reduce the model's ability to process and learn information from HINs.

To address these issues, we propose a new metacognition-driven user-to-project recommendation method. MUP-ES finds items that meet both the project requirements and the user's practice goals by deeply mining the node information in HINs. We take the project as the ontology, and continuously adjust the matching value between the project and the user through the metacognitive information aggregation process, and to match the user to the project that can help them to the greatest extent. We refer to the process of continuously aggregating target user information by the project as metacognitive information aggregation. Specifically, we design a neural network framework for learning auxiliary information in each meta-path. We assign different weights to paths through learnable weight parameters to filter out low-value information in meta-paths. Next, MUP-ES aggregates the internal node information of meta-paths through the multi-head self-attention mechanism. Transfer the information of the adjacent nodes of the meta-path, and collect auxiliary information about the user in the HINs. We aggregate information into project nodes, continuously optimize the project nodes' ability to judge users, and then select suitable users. Finally, our experiments on real-world datasets demonstrate the effectiveness of the model.

Our contributions are summarized below:

- We serve online education users from an innovative perspective of practical projects. We propose a novel metacognition-driven user-to-project recommendation method. The combination of user learning results and project knowledge requirements can not only enable users to quickly match projects, but also deepen users' application of knowledge by completing projects.
- We construct the heterogeneous network of user learning outcomes and project knowledge requirements, and design a novel neural network framework. The framework deeply mines the structural and semantic information of HINs. We extract valuable information in an attentional manner and pass the information to project nodes to match suitable users.
- We demonstrate the effectiveness of this model through extensive experiments on two real-world datasets. We compare our model to a range of baseline methods for graph embedding. Furthermore, we demonstrate the cold-start mitigation capability of the proposed model.

2 Related work

In this section, we review related work research in two areas, namely recommending in the field of education, and meta-path based HINs.

2.1 Recommending in the field of education

In the field of education, the initial research mainly focused on dropout analysis, procrastination behavior, and learning behavior [5, 6, 20, 21]. Xia Jing et al. [4] of Tsinghua university proposed a model for the MOOCs platform through the user's access behavior model to recommend courses for online users. Yanxia Pang et al.[22] added the reference

information on learner satisfaction in the recommended course task. Recently, researchers have begun to consider adding user modeling [23], which is widely used in other fields such as social media, to inject new ideas into course recommendations. J. Gong et al. [2] believe that the user's interest value and required knowledge concepts need to be added to the course recommendation for a flexible recommendation. Jibing Gong et al. [2] recommend users to the knowledge concepts they need through a meta-path method combined with an extended matrix factorization framework. In the field of education, researchers have primarily focused on serving beginners. How to combine practice with educational theory to help users who have completed their studies to take the next step is almost blank.

2.2 Meta-path based HINs

HINs [24] refers to a complex network structure composed of various types of nodes or edges. As an emerging direction, HINs can naturally model complex objects and their rich relationships in the educational domain, in which objects are of different types and links among objects represent different relations. Earlier, low-dimensional vectors of graph structure information are learned and used for various downstream tasks, such as Deepwalk [25] and Node2vec [26], these methods through two-stage training methods using unsupervised embedding to generate structural features and serve as recommendation models. Metapath2vec [17] proposes a random walk based on meta-path constraints and then learns node representations through word2vec [27]. However, the above methods do not use supervised information in the recommendation task based on graph embedding, so it is difficult to adapt to different recommendation tasks. Early graph embeddings are only suitable for isomorphic graphs, the methods mentioned above do not use the supervision information in the recommendation task in the graph embedding stage, so it is difficult to be suitable for different recommendation tasks. Early graph embeddings are only suitable for isomorphic graphs, such as graph neural network (GCN) [28], graph attention network (GAT) [29], etc. To apply graph neural networks on heterogeneous graphs, explorers combine random walk based meta-path methods with graph neural networks [14, 15, 30]. This is a method based on meta-path modeling in HINs, which can effectively capture structural and semantic information in heterogeneous graphs. For example, the ARNN [31] explores heterogeneous neighbors by meta-path based random wandering and uses RNN to model sequences, and then uses an attention mechanism to learn features in heterogeneous neighbors. However, the model's handling of manually defined meta-paths continues the previous practice, and whether the defined meta-path types are sufficiently valuable is still not well resolved.

Inspired by these pioneering research works, we construct HINs of user learning information and project knowledge requirements, and design meta-paths as carriers to obtain relevant information representations in HINs. To our knowledge, it is the first time that online education has assisted users through projects. We design an information extraction framework to provide accurate user-to-project matching through valuable information aggregation.

3 Preliminaries

In this section, we give formal definitions of basic concepts related to HINs in the educational domain.

Definition 1 (HINs in the Field of Education) HINs [24] are defined as a graph $G = (v, \varepsilon)$ associated with a node type mapping function $\phi : v \rightarrow O$ and a link type mapping function $\varphi : \varepsilon \rightarrow R$. A and R denote the predefined sets of node types and link types, respectively, with $|A| + |R| > 2$.

Example 1 Take Figure 1 as an example. This is a visualization of the dataset, which we will build as HINs using an n-part graph [32]. We represent each project with user-completed papers, and HINs consist of four categories of entities: project, user, knowledge concepts, and course. Additionally, there is a set of links that describe the relationships between these entities. In HINs, two entities can be connected via different semantic paths, which are defined as meta-paths.

Definition 2 (Meta-path) A meta-path P is defined as a path in the form of $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_n} A_{l+1}$ (abbreviated as $A_1 A_2 \dots A_{l+1}$), which describes a composite relation $R = R_1 \circ R_2 \circ \dots \circ R_n$ between node types A_1 and A_{l+1} , here "o" denotes the composition operator on relations.

Definition 3 (Meta-path Context Instance) Given a meta-path P of HINs, the context of the meta-path is defined as the aggregated instance of the relationship between nodes on the HINs.

Example 2 Take Figures 2 and 3 as an example. On the MOOCCube dataset, we design meta-paths for the project, and different meta-paths express different interaction semantics. For example, the meta-path between two projects can be defined as follows: $P \xrightarrow{need} K \xrightarrow{need^{-1}} P$, indicating that this knowledge concept is required to complete these two different projects; $P \xrightarrow{need} K \xrightarrow{exist\ in} C \xrightarrow{exist\ in^{-1}} K \xrightarrow{need^{-1}} P$, indicating that the different knowledge concepts required to complete these two different projects can be obtained from

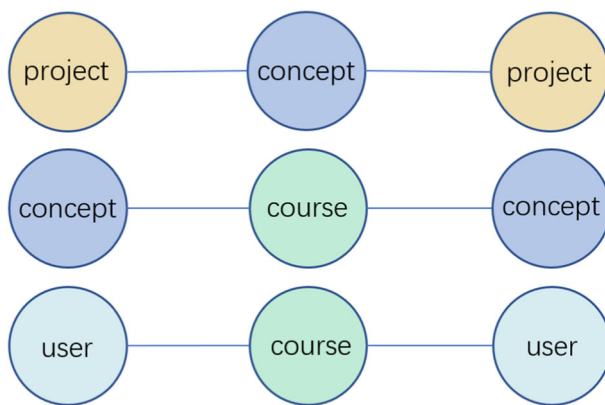


Figure 2 Meta-path types obtained from HINs

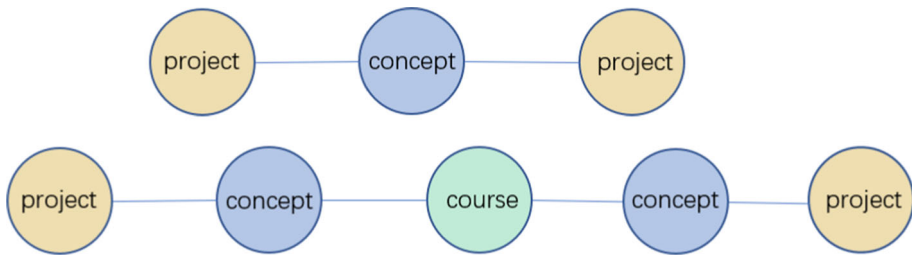


Figure 3 Project-specific meta-path examples

this obtained from course. We get the semantic instances we want by designing different meta paths.

Definition 4 (HINs based Recommendation) Given HINs $G = (v, \varepsilon)$, with node attribute matrix $X_{A_i} \in \mathbb{R}^{\varepsilon_{A_i} \times R_{A_i}}$. For recommendation tasks, we focus on nodes A_i in the network and the relationships R_{A_i} between them. Our goal is to predict match scores s_{p_i, u_i} between projects $p_i \in A_i$ and users $u_i \in A_i$.

Many studies have contributed to the embedding work based on HINs. On this basis, we propose a new method for mining HINs information. We filter the artificially defined meta-paths through an attention mechanism, and then design the multi-head self-attention method to aggregate node information between paths. We propose a new framework to address novel problems in the online education domain. Through this framework, the information on HINs can be collected efficiently and accurately, and the user learning information can be strongly correlated with the project requirements, so that the project nodes can find suitable users from the information.

4 Proposed approach

In this section, we detail our proposed approach: MUP-ES. We build a heterogeneous networks of user learning outcomes and project knowledge requirements, and drive the project to find users who are strongly related to the project through the metacognitive information aggregation process.

4.1 Overview

The basic idea of the proposed model MUP-ES is to aggregate the information of user learning outcomes for a project to better recommend users who are suitable for the project. We design two different information processing modules to better mine the information of HINs, and aggregate the auxiliary information into the project nodes.

Figure 4 shows the framework of MUP-ES. We leverage attention weights to reinforce meta-paths with valuable information, thereby addressing the problem that artificial meta-paths are not always effective. In the information aggregation module, we learn node features within the path through a multi-head self-attention mechanism. Finally, aggregate the information to project nodes.

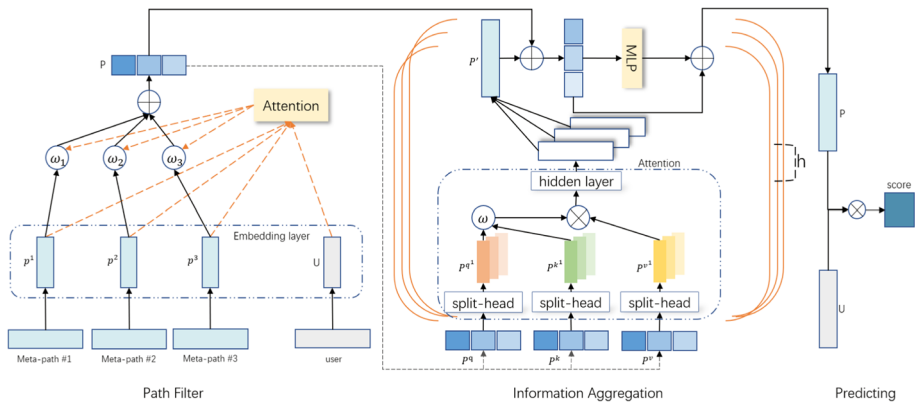


Figure 4 The architecture of MUP-ES

4.2 Meta-path filtering

As described in Section 3, when we design multiple types of meta-path instances for each project node $P_a = (p_1, p_2, \dots, p_j)$ based on HINs. We cannot determine which paths are valid. Therefore, we design an attention strategy to solve this problem. We calculate the contribution of each node $v_a \subseteq P_a$ in the path to the project (e.g., knowledge concepts and courses in the path), and calculate the relationship weights between each path and the interaction with target users. Intuitively, if there are courses or knowledge concepts in the path that the target user has taken, then those courses and knowledge concepts that can contribute to the completion of the project should have a greater impact on the project. When multiple courses or knowledge concepts in the path meet the requirements, we can believe that it is a path with valuable information, then it should be given a higher weight. On the contrary, if it is a path with low value information, then when we aggregate the information, we are reducing or even ignoring the information value of this path.

First, we parameterize each path p belongs to P_a as a matrix, embedding p to encode the heterogeneous information in the path, and embedding u to encode the attribute information of the target user, and then we parameterize $\omega(i, j)$ as attention neural network with p_j and u_i as input. And $\omega(i, j)$ is a learnable parameter that represents the influence degree of nodes on each project path to target users u_i .

$$\begin{aligned}
 y(i, j) &= h^T \text{ReLU}(v_i(p_j + u_i) + b), \\
 \omega(i, j) &= \text{softmax}(y(i, j)) = \frac{\exp y(i, j)}{\sum_{j' \in P_j} \exp y(i, j')}, \tag{1}
 \end{aligned}$$

where v_i is the weight matrix of the attention network that converts the meta-path embedding and the target user embedding into a hidden layer, where b is the bias vector of the hidden layer, and we use ReLU as the activation function of this hidden layer, and then

obtain the weight coefficient $y(i, j)$ of each meta-path by projecting the weight vector. In the last part of this module, we use softmax to normalize the results, which converts the attention network results into a probabilistic express $\omega(i, j)$ to facilitate subsequent model calculations. This is a common practice. We multiply the weight value and the initial embedding value matrix to get the new node vector.

The first part of Figure 4 shows our design for filtering multiple types of meta-path. We learn the weight of the relationship between node information and target users in the path, and strengthen the valuable information path related to the user through the weight, so that the next module aggregates more meaningful node information.

4.3 Information aggregation in meta-paths

After we obtain each project meta-path P_a with valuable information, our goal is to aggregate and learn the node structure and semantic information within the meta-path to the project nodes. We consider two problems encountered in information aggregation within meta-paths. First, is the training speed problem. Second, is the problem of capturing information over long distances of meta-paths. Inspired by the transformer-encoder part proposed by Google, we design a multi-headed attention mechanism for meta-path internal nodes to aggregate and learn the internal information of each project path and use it for recommendation tasks.

As shown in Figure 4, to learn the more realistic contribution of the auxiliary information to the project, we create three matrix vectors q, k, v for each input node in the meta-path, i.e., assign three weights, and we perform the relevant calculations on the matrix after these three linear mappings. The implementation is shown in (2). We map the query vector $p_q \in \mathbb{R}^{d_{model}d_k}$ to the same high-dimensional space as the key vector $p_k \in \mathbb{R}^{d_{model}d_k}$ to calculate the similarity. We use a matrix approach to improve the computational speed. After computing the attention matrix ω_p , in order to prevent the inner product from being too large, we perform $\sqrt{d_k}$ normalization to stabilize gradients during training. And we convert the attention score into a probability distribution through the softmax function. This score represents the value of the influence of the nodes in the path on other nodes in the current environment. Finally, we weight the value vector $p_v \in \mathbb{R}^{d_{model}d_v}$ with the obtained attention score matrix.

$$\begin{aligned}\omega_p^h &= \frac{n_h(p_q p_{k^h}^T)}{\sqrt{d_k}}, \\ \text{attention}_h &= \sum_i \frac{\exp \omega_p^h}{\sum_{h'} \exp \omega_p^{h'}} p_{v_i}, \\ x_{MH} &= \text{concat}(\text{attention}_h, h = 1, 2, \dots) W^0, \end{aligned} \quad (2)$$

To capture the attention of nodes in the path from different perspectives, we design the multi-head attention mechanism for the internal node information of the meta-path. This mechanism is a "representation subspace". It provides multiple attention layers and allows the model to give a composite score for each node's contribution to the project. As shown in

Figure 4, we will obtain h sets of weight matrices $attention_h (h = 1, 2, \dots)$, each of which will be computed independently according to the previous implementation.

$$\begin{aligned}
 x_c &= linear(ReLU(linear(\omega(i, j) * p_{v_i} + x_{MH}))), \\
 mu_t &= \frac{1}{m} \sum_{i=1}^m x_c, \\
 \sigma_j^2 &= \frac{1}{m} \sum_{i=1}^m (x_{c_i} - mu_i)^2, \\
 LN(x) &= \alpha \odot \frac{x_{ij} - mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \tag{3}
 \end{aligned}$$

We get a matrix consisting of h vectors, multiplied by an additional weight matrix W_0 , which is compressed into a matrix x_{MH} . We perform residual connections x_c on node vectors weighted by multi-head attention. To speed up model training and accelerate convergence, we normalize the hidden layers in the neural network to a standard normal distribution. We take the mean mu_t and variance σ_j^2 of the matrix, and perform a final normalization calculation to get $LN(x)$, and the implementation is shown in (3).

$$x_{out} = max(0, (LN(x) + x_{MH})W_1 + b_1)W_2 + b_2, \tag{4}$$

Finally, to enhance the expressiveness of the model, we will add a feed-forward layer and perform residual connection and normalization again. The implementation is shown in (5).

4.4 Model optimization

Since we perform the user recommendation task from the perspective of ranking, we choose the pairwise learning method to optimize the model parameters, which means that the score of users who are more suitable for the project should be higher than that of ordinary users. Specifically, we use regression-based pairwise loss, which is a common choice in the recommendation.

$$L_{user} = \sum_{(i, j, j' \in O)} (r_{ijj'} - \hat{r}_{ijj'})^2 = \sum_{(i, j, j' \in O)} (\hat{r}_{ij} - \hat{r}_{ij'} - 1)^2, \tag{5}$$

Where O represents the training set, (i, j, j') represents the interaction instance between each group of projects and users, where project p_i indicates interaction with user u_j , but no interaction with user $u_{j'}$. For the project-user pair (p_i, u_j) , we learn through feedback and calculate the prediction score r_{ij} , so that the r_{ij} gap between positive and negative samples is widened.

5 Experiments

In this section, we experimentally demonstrate the effectiveness of MUP-ES for educational recommendations by HINs.

5.1 Datasets

We adopt two widely used education-related datasets, the MOOCCube dataset [8] from the xuetangX platform, and the DBLP dataset [33] which provides metadata related to computer-based papers.

- **MOOCCube** dataset is an open data repository for large-scale online education-related knowledge graphs, data mining, and other researchers interested in online open courses. The course and user activity data in this dataset comes from the real environment of xuetangX, including 486 courses, 5,715 papers, 38,181 videos, 579 concepts, 4,723 real users of MOOCs, and interaction data of three main dimensions of user behavior, course, and concept. In this work, for the MOOCCube dataset, we used the data of the papers completed by the user, the courses and concepts that the user learned, the user id, and the interaction data between them (project-P, concept- E, course-C) as shown in Table 1.
- **DBLP** dataset is an integrated database system for storing computer literature. In this work, we use a subset of DBLP provided by [34], which contains 4,057 authors, 7,723 concepts, 14,328 papers, and 20 publication locations. Like the MOOCCube dataset, we treat papers as projects that users need to complete. Table 1 shows the interaction data between them (project-P, concept-T, location -L).

The relational data between each instance in the two datasets is shown in Table 1, for which we constructed different HINs and designed different meta-paths for the two different networks.

5.2 Evaluation metrics

To evaluate the performance of the model recommending users, we employ widely used metrics, namely the hit rate of the top-k users (HR@k) and the normalized discounted cumulative gain of the top k users (NDCG@k). Specifically, for users interacting with each project in the MOOCCube dataset, we randomly sample one as the test data, which prevents the test and training sets from overlapping. This is done because sorting all the test data takes a long time. For the DBLP dataset, due to fewer interaction data, we divide it into 67% as the training set and 33 % as the test set. For both datasets, we calculate with each group of 31 users (of which there is only one positive interaction sample and 30 negative samples) as the test set, we calculate the interaction score of the project and each user. We observe the top-k scores and compute the real interacting user score $R_u = r_u^1, r_u^2, \dots, r_u^K$ where r_u^i represents the position of these users in R_u . T_i is the project set of real interactions of user

Table 1 Entity relationship statistics and meta-path types

Dataset	Relationship	Quantity	Meta-path
MOOCCube	project-concept	5,877	PEP PECEP
	concept-course	7,799	
	course-user	24,319	
	project-user	48,979	
DBLP	project-concept	85810	PTP PLP PTPLP
	project-location	14,328	
	project-author	19,645	

u in the test data. $I(x)$ means equal to 1 if $x > 0$ otherwise equal to 0 and N is the total number of users in the test set:

$$HR@K = \frac{1}{N} \sum_u I(|R_u \cap T_i|) \quad (6)$$

We also consider the user's NDCG@k(k=5,10), i.e. top users get higher scores, where Z is a normalization factor that represents the ideal value of $DCG@K$:

$$NDCG@K = \frac{1}{Z} DCG@K = \frac{1}{Z} \sum_{j=1}^K \frac{2^{I(|r_u^j \cap T_i|)} - 1}{\log_2(j+1)} \quad (7)$$

5.3 Baselines

To demonstrate the effectiveness of the model, we compare the following baseline methods and use them for the recommendation task:

- **Deepwalk** [25] is a model that combines random walk and word embedding algorithms. We use the random walk method of deepwalk in the information network, and the isomorphic path from the walk is used in word2vec and the generated word vector is used for recommended tasks.
- **Node2vec** [26] is an extension of deepwalk. It applies a biased random walk. Similarly, we enter the walking path into the word embedding model and use the generated word vector for recommendation tasks.
- **Metapath2vec** [17]: is a method that can deal with heterogeneous networks. Different from the previous method, metapath2vec uses a random walk based on meta-paths to construct the heterogeneous field of each vertex and then uses the skip-gram model to complete the vertex embedding.
- **NCF** [35]: uses the neural network structure to replace the inner product operation, which can well realize and express those hidden features that were originally modeled by matrix decomposition. In this NCF scheme, we use project and user as model input.
- **GAT** [29]: model calculates the attention coefficient of each vertex and the domain node and aggregates the features through the attention coefficient. In this GAT scheme, we take the meta-path of each project as input and learn the node information of the project path through the attention mechanism.
- **ARNN** [31]: proposes a new method to capture neighbors using an attention mechanism. The attention mechanism is used to perform feature aggregation on meta-path-based heterogeneous neighbors, and RNN are used to model sequences. Efficiently collaborate with RNN layers in a unified manner by tuning the attention mechanism.

For traditional graph walk embedding models, including Deepwalk, Node2vec, and Metapath2vec, we set the window size to 1 and the walk length to 10. For NCF, we will observe the performance of the neural network on the subject and user in the no-path case. For the GAT, ARNN, and MUP-ES models, we set the walk window size and input path length to the same values as the traditional models. The above models are all implemented based on the Tensorflow framework. We set the batch size to 128 and the learning rate to [0.005, 0.001, 0.0005], and the corresponding epoch rounds are [0~15, 15~25, >20], and the Adam optimizer is used. All experiments were run on the same server.

5.4 Experimental results

We experiment and compare the results with our model and the baseline model on the MOOCcube and DBLP datasets, respectively. As previously introduced, for a fair comparison, we do the same for all models in the dataset preprocessing and set the same hyperparameters. The experimental results show that the MUP-ES model consistently outperforms all models. As shown in Table 2, in the traditional graph walk model, the processing of HINs by the Metapath2vec model is weaker than that of Deepwalk and Node2vec on isomorphic graphs. The results indicate that if the optimal meta-path cannot be found, it will drag down the development of the model. Other results from the table indicate that NCF and GAT are not better than graph walk models in handling heterogeneous graphs. The ARNN model performs strongly because it is not only applicable to HINs but also focuses on the information within the path, but is still lower than the model proposed in this paper.

5.5 Further analysis of MUP-ES

In this link, we conduct a series of detailed analyses on the model MUP-ES proposed in this paper, to better understand the characteristics of MUP-ES.

5.5.1 Ablation experiment

To gain further insight into the effectiveness of each component in the model, we performed some ablation experiments.

Our first ablation experiment is to test three ways of information aggregation for MUP-ES. In the process of aggregating the internal messages of nodes, we conduct $MUP - ES_{(0)}$ experiments by directly extracting the project node of the meta-path, and we conduct $MUP - ES_{sum}$ experiment by the secondary aggregation of the information aggregation module to sum other auxiliary information node vectors in the path. The $MUP - ES_{concat}$ experiment is performed by splicing the internal node vectors of the meta-path. The experimental results are shown in Table 3. Surprisingly, the method of directly extracting the subject node vector is better than the method of summing the information in the path, indicating that sometimes the method of direct addition is not very useful, but the two methods are slightly lower than the splicing method.

Table 2 Performance comparison of different methods for recommending users for projects. The best results are shown in bold

Dataset	Metrics	Model							
		@ _k	Deepwalk	Node2vec	Metapath2vec	NCF	GAT	ARNN	MUP-ES
MOOCcube	HR	k=5	0.4414	0.4468	0.4408	0.3668	0.4450	0.4795	0.5327
		k=10	0.7195	0.7208	0.7033	0.6050	0.6847	0.7441	0.7981
	NDCG	k=5	0.2733	0.2809	0.2730	0.2250	0.2764	0.2985	0.3302
		k=10	0.3694	0.3739	0.3610	0.2976	0.3518	0.3766	0.4121
DBLP	HR	k=5	0.4076	0.4432	0.4271	0.5372	0.5305	0.5108	0.6157
		k=10	0.6081	0.6416	0.6708	0.7023	0.7010	0.7232	0.7937
	NDCG	k=5	0.2858	0.2717	0.2641	0.3978	0.3903	0.3794	0.4323
		k=10	0.3222	0.3164	0.3329	0.4445	0.4455	0.4328	0.5187

Table 3 Ablation experiment of three information aggregation methods of MUP-ES

Dataset	MOOCCube				DBLP			
	k=5		k=10		k=5		k=10	
@k	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
$MUP - ES_{sum}$	0.5218	0.3218	0.7926	0.4052	0.5259	0.3764	0.7101	0.4385
$MUP - ES_{(0)}$	0.5318	0.3262	0.7937	0.4078	0.5716	0.4242	0.7478	0.4759
$MUP - ES_{concat}$	0.5327	0.3302	0.7981	0.4121	0.6046	0.4118	0.7937	0.5187

Table 4 Quantitative results of modular ablation study

Dataset	MOOCCube				DBLP			
	k=5		k=10		k=5		k=10	
@k	HR	NDCG	HR	NDCG	HR	NDCG	HR	NDCG
$MUP - ES_{filter}$	0.5218	0.3218	0.7926	0.4052	0.4752	0.3364	0.6641	0.3840
$MUP - ES_{internal}$	0.5318	0.3262	0.7937	0.4078	0.4824	0.3449	0.6501	0.4009
$MUP - ES$	0.5327	0.3302	0.7981	0.4121	0.6046	0.4118	0.7937	0.5187

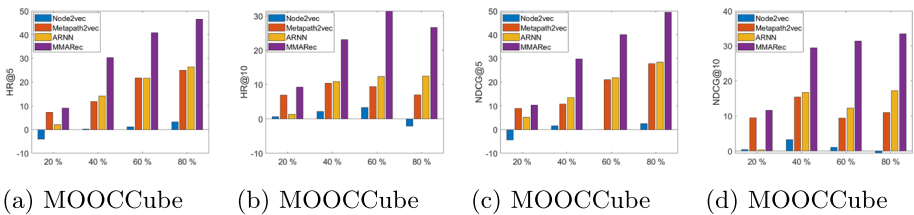


Figure 5 Performance comparison in cold-start, y-axis denotes the improvement ratio over Deepwalk

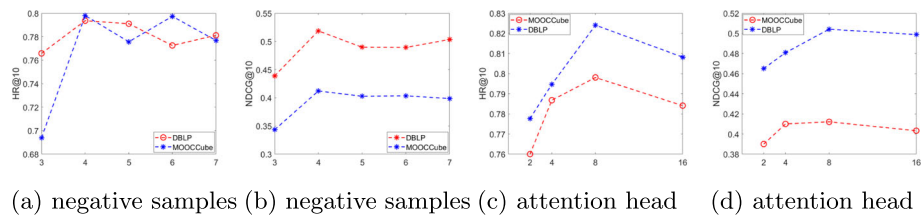


Figure 6 Parameter tuning of MUP-ES

Our second ablation experiment is to demonstrate the effectiveness of the model proposed in this paper. As shown in Table 4, for $MUP - ES_{filter}$, we first only consider the screening and filtering of multi meta-path and do not perform aggregation of node information within the meta-path. For $MUP - ES_{internal}$, we only consider the aggregation of internal information for a single meta-path. Finally, we compare the results with the complete MUP-ES model, and it turns out that the proposed meta-path filtering and meta-path internal information aggregation are effective and necessary.

5.5.2 Cold-start recommendation

We conduct experiments on the MOOCCube dataset, which is more representative of the education domain. This dataset is also sparser than DBLP, and is more suitable for observing the cold start mitigation capability of the model in the educational recommendation. We investigated the recommendation performance under different degrees of cold starts, i.e., interaction sparsity. We varied the amount of training data, using 20%, 40%, 60%, and 80% of the original training set as the training set, respectively. For comparison, we selected representative graph walking baseline models: Deepwalk, Node2vec, Metapath2vec, and ARNN. For convenience, we report the improvement rates of other comparison methods on Deepwalk. We conduct experiments on the MOOCCube dataset, as shown in Figure 5, we can see that MUP-ES yields the most improvement over Deepwalk in all the cases and the results show that our model is effective in mitigating cold start in the field of educational recommendation.

5.5.3 Parameter tuning

For the recommendation task, our model includes some important parameters that need to be tuned. We examined two parameters of the model, namely the number of negative samples and the number of heads of multiheaded attention. For the negative samples, we tested the set of numbers 3, 4, 5, 6, 7. For the number of heads of attention, we tested the set of quantities as 2, 4, 8, 16. As shown in Figure 6, the optimal number of negative samples is 4 and the optimal number of heads for the attention mechanism is 8.

6 Conclusion

In this paper, we proposed a novel metacognition-driven user-to-project recommendation method. We innovatively focus on combining theoretical learning and practice, and provide services for online education users from the perspective of practical projects. We proposed a new idea of building HINs based on user learning information and project requirements. We designed a novel deep learning framework to dig deeper into HINs information. We achieved metacognitive information aggregation through two components in the framework, i.e., path filtering and information aggregation, so that users can be accurately matched to projects that help them. We conducted extensive experiments on the MOOCCube and DBLP datasets to demonstrate the effectiveness of our model. Finally, we conducted experiments on the cold-start problem in the field of educational recommendation, and the results show that the proposed model can alleviate the cold-start problem.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No.62267002, 62167002, 61862013, U1811264), Guangxi Natural Science Foundation of China (Grant No.

2020GXNSFAA159117), Guangxi Key Laboratory of Trusted Software (No. KX202052), Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (YQ21102), and Innovation Project of GUET Graduate Education (2022YCXS077).

Author Contributions All authors contributed to the study conception and design. Zezheng Wu, wrote the main manuscript text and completed experiments. Haonan Jiang, Xinghe Cheng, Haotian Huang, prepared figures and analyze experiments. Zezheng Wu, Qing Yang, and Jingwei Zhang reviewed the manuscript.

Funding This work is supported by the National Natural Science Foundation of China (Grant No.62267002, 62167002, 61862013, U1811264), Guangxi Natural Science Foundation of China (Grant No. 2020GXNSFAA159117), Guangxi Key Laboratory of Trusted Software (No. KX202052), Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (YQ21102), and Innovation Project of GUET Graduate Education (2022YCXS077).

Data Availability All of the material is owned by the authors and no permissions are required.

Declarations

Ethics approval This declaration is "not applicable" to the content of this submission.

Conflict of Interests The authors have no relevant financial or non-financial interests to disclose.

References

- Piao, G.: Recommending knowledge concepts on mooc platforms with meta-path-based representation learning. International Educational Data Mining Society (2021)
- 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, pp. 79–88 (2020)
- Alsaad, F., Alawini, A.: Unsupervised approach for modeling content structures of moocs. International Educational Data Mining Society (2020)
- Jing, X., Tang, J.: Guess you like: course recommendation in moocs. In: Proceedings of the International Conference on Web Intelligence, pp. 783–789 (2017)
- Feng, W., Tang, J., Liu, T.X.: Understanding dropouts in moocs. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 517–524 (2019)
- Prekaj, B., Velardi, P., Distanti, D., Faralli, S.: A reproducibility study of deep and surface machine learning methods for human-related trajectory prediction. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 2169–2172 (2020)
- Gardner, J., Yang, Y., Baker, R.S., Brooks, C.: Modeling and experimental design for mooc dropout prediction: A replication perspective. International Educational Data Mining Society (2019)
- Yu, J., Luo, G., Xiao, T., Zhong, Q., Wang, Y., Feng, W., Luo, J., Wang, C., Hou, L., Li, J., et al.: Mooccube: a large-scale data repository for nlp applications in moocs. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3135–3142 (2020)
- Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017)
- Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., He, Q.: A survey on knowledge graph-based recommender systems. *IEEE Transactions on Knowledge and Data Engineering* (2020)
- Sun, Z., Deng, Z.-H., Nie, J.-Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv: [1902.10197](https://arxiv.org/abs/1902.10197) (2019)
- Zhang, J., Shi, X., Zhao, S., King, I.: Star-gcn: Stacked and reconstructed graph convolutional networks for recommender systems. arXiv: [1905.13129](https://arxiv.org/abs/1905.13129) (2019)
- Shi, C., Hu, B., Zhao, W.X., Philip, S.Y.: Heterogeneous information network embedding for recommendation. *IEEE Trans. Knowl. Data Eng.* **31**(2), 357–370 (2018)
- Xu, Y., Zhu, Y., Shen, Y., Yu, J.: Learning shared vertex representation in heterogeneous graphs with convolutional networks for recommendation. In: *IJCAI*, pp. 4620–4626 (2019)
- Wang, X., Wang, Y., Ling, Y.: Attention-guide walk model in heterogeneous information network for multi-style recommendation explanation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 6275–6282 (2020)

16. Ji, H., Zhu, J., Wang, X., Shi, C., Wang, B., Tan, X., Li, Y., He, S.: Who you would like to share with? a study of share recommendation in social e-commerce. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 232–239 (2021)
17. Dong, Y., Chawla, N.V., Swami, A.: Metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 135–144 (2017)
18. Wang, X., Ji, H., Shi, C., Wang, B., Ye, Y., Cui, P., Yu, P.S.: Heterogeneous graph attention network. In: The World Wide Web Conference, pp. 2022–2032 (2019)
19. Fu, T.-Y., Lee, W.-C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1797–1806 (2017)
20. Hajri, H., Bourda, Y., Popineau, F.: Personalized recommendation of open educational resources in moocs. In: International Conference on Computer Supported Education, pp. 166–190. Springer (2018)
21. Yao, M., Sahebi, S., Feyzi Behnagh, R.: Analyzing student procrastination in moocs: a multivariate hawkes approach. In: Proceedings of the 13th Conference on Educational Data Mining (EDM2020) (2020)
22. Pang, Y., Liao, C., Tan, W., Wu, Y., Zhou, C.: Recommendation for mooc with learner neighbors and learning series. In: International Conference on Web Information Systems Engineering, pp. 379–394. Springer (2018)
23. Zarrinkalam, F., Faralli, S., Piao, G., Bagheri, E., et al.: Extracting, mining and predicting users' interests from social media. *Foundations and Trends® in Information Retrieval* **14**(5), 445–617 (2020)
24. Sun, Y., Yu, Y., Han, J.: Ranking-based clustering of heterogeneous information networks with star network schema. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 797–806 (2009)
25. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 701–710 (2014)
26. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv: [1301.3781](https://arxiv.org/abs/1301.3781) (2013)
28. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv: [1609.02907](https://arxiv.org/abs/1609.02907) (2016)
29. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv: [1710.10903](https://arxiv.org/abs/1710.10903) (2017)
30. Lu, Y., Fang, Y., Shi, C.: Meta-learning on heterogeneous information networks for cold-start recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1563–1573 (2020)
31. Guo, Q., Sun, Z., Zhang, J., Theng, Y.-L.: An attentional recurrent neural network for personalized next location recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 83–90 (2020)
32. Data61, C.: Stellargraph Machine Learning Library GitHub (2018)
33. Ley, M.: Dblp: some lessons learned. *Proceedings of the VLDB Endowment* **2**(2), 1493–1500 (2009)
34. Ji, M., Sun, Y., Danilevsky, M., Han, J., Gao, J.: Graph regularized transductive classification on heterogeneous information networks. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 570–586. Springer (2010)
35. Yang, T., Yang, C., Zhang, L., Shi, C., Hu, M., Liu, H., Li, T., Wang, D.: Co-clustering interactions via attentive hypergraph neural network (2022)

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Affiliations

Zezheng Wu¹ · Haonan Jiang¹ · Xinghe Cheng² · Haotian Huang¹ · Qing Yang² · Jingwei Zhang¹

✉ Jingwei Zhang
gtzjw@hotmail.com

Zezheng Wu
1398820019@qq.com

Haonan Jiang
1048375125@qq.com

Xinghe Cheng
guetcxh@hotmail.com

Haotian Huang
huanghaotian42@gmail.com

¹ Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Jinji Rd., Guilin 541004, Guangxi, China

² Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Jinji Rd., Guilin 541004, Guangxi, China