Shao MM, Jiang WJ, Wu J *et al.* Improving friend recommendation for online learning with fine-grained evolving interest. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 37(6): 1444–1463 Nov. 2022. DOI 10.1007/s11390-021-2124-z

Improving Friend Recommendation for Online Learning with Fine-Grained Evolving Interest

Ming-Min Shao[1] (邵明民), Wen-Jun Jiang[1,*] (姜文君), *Senior Member, CCF*, Jie Wu[2], *Fellow, IEEE*
Yu-Qing Shi[3] (施雨晴), TakShing Yum[4] (任德盛), *Fellow, IEEE*, and Ji Zhang[5] (张　吉), *Senior Member, IEEE*

[1] *College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China*
[2] *Department of Computer and Information Sciences, Temple University, Philadelphia 19122, U.S.A.*
[3] *School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China*
[4] *Department of Information Engineering, Chinese University of Hong Kong, Hong Kong 999077, China*
[5] *Department of Mathematics and Computing, University of Southern Queensland, Brisbane, QLD 4350, Australia*

E-mail: {shaomingmin, jiangwenjun}@hnu.edu.cn; jiewu@temple.edu; shiyuqing@stumail.neu.edu.cn
tsyum@ie.cuhk.edu.hk; Ji.Zhang@usq.edu.au

Received December 31, 2021; accepted October 30, 2022.

Abstract Friend recommendation plays a key role in promoting user experience in online social networks (OSNs). However, existing studies usually neglect users' fine-grained interest as well as the evolving feature of interest, which may cause unsuitable recommendation. In particular, some OSNs, such as the online learning community, even have little work on friend recommendation. To this end, we strive to improve friend recommendation with fine-grained evolving interest in this paper. We take the online learning community as an application scenario, which is a special type of OSNs for people to learn courses online. Learning partners can help improve learners' learning effect and improve the attractiveness of platforms. We propose a learning partner recommendation framework based on the evolution of fine-grained learning interest (LPRF-E for short). We extract a sequence of learning interest tags that changes over time. Then, we explore the time feature to predict evolving learning interest. Next, we recommend learning partners by fine-grained interest similarity. We also refine the learning partner recommendation framework with users' social influence (denoted as LPRF-F for differentiation). Extensive experiments on two real datasets crawled from Chinese University MOOC and Douban Book validate that the proposed LPRF-E and LPRF-F models achieve a high accuracy (i.e., approximate 50% improvements on the precision and the recall) and can recommend learning partners with high quality (e.g., more experienced and helpful).

Keywords online social network, friend recommendation, fine-grained interest, evolving feature tag, online learning community

1 Introduction

People need friends to communicate and cooperate with, either online or offline. Hence, friend recommendation has become a key function in online social networks (OSNs). However, existing studies on friend recommendation often neglect users' fine-grained interest and their evolution, leading to some improper recommendation. Furthermore, there are very few studies on friend recommendation in some OSNs, especially for online learning. This incites us to improve friend recommendation with fine-grained evolving interest, and we take online learning as an example application scenario.

Regular Paper

This research was supported by the National Natural Science Foundation of China under Grant Nos. 62172149, 61632009, 62172159, and 62172372, the Natural Science Foundation of Hunan Province of China under Grant No. 2021JJ30137, the Open Project of ZHEJIANG LAB under Grant No. 2019KE0AB02, and the Natural Science Foundation of Zhejiang Province of China under Grant No. LZ21F030001.

*Corresponding Author

Online learning community (OLC) is a special OSN that consists of learning users and has been becoming more and more popular in recent years. By the end of 2020, there were 16.3k courses and a total of 180 million enrolling learners in MOOCs①. OLCs such as Coursera②, Edx③, Chinese University MOOC (CUM)④, and Douban Book⑤ provide a large number of courses or e-books, and people can learn online anywhere and anytime.

Sakulwichitsintu *et al.* [1] analyzed online peer learning and explored what influences students' experience, and then designed a peer learning framework for enhancing students' learning experience in OLCs. However, learners have few friends to discuss with in OLCs; moreover, it is difficult for learners to find partners by themselves, who can improve their enthusiasm and efficiency for online learning. Therefore, there is an urgent need for learning partner recommendation.

Traditional learning activities are usually carried out in classes, where teachers provide face-to-face tutoring, and the cooperation among classmates can also help improve the learning effect. However, in the open virtual learning community, massive learning resources often confuse learners. In contrast to the large-scale enrollment, the completion rate of online courses is in truth very low. One common reason why students quit learning is that they could not discuss problems with classmates [2]. During the learning process, learners often feel helpless when they encounter obstacles that are difficult to solve, and their learning passion and efficiency will decrease.

Learning partners refer to users who serve as partners for cooperation in the learning community; they usually share similar learning interest. Learning partners are important resources for the open virtual learning community, which help online learners overcome learning obstacles and improve their communication skills. Existing studies on learning partner recommendation mainly exploit learning behaviors or interest. Gong *et al.* [3] proposed a hybrid deep neural network framework to recommend social friends, considering both interactive semantics and contextual enhancement.

Although some achievements have been done, friend recommendation in OSNs, especially learning partner recommendation, still faces three main challenges: 1) the challenge of fine-grained interest: existing studies rarely pay attention to users' fine-grained learning interest; 2) the challenge of time-evolving interest: existing studies often overlook the time-evolving feature of learning interest; 3) the challenge of datasets: most datasets for online learning are about learning behaviors, and few datasets involving the interest and connections between learners.

Motivation. Keeping the friend recommendation task and the above challenges in mind, we try to: 1) analyze users' fine-grained learning interest and calculate their similarity in online learning community, 2) explore the evolution of the learning interest to enhance friend recommendation, and 3) crawl the data to exploit users' fine-grained interest for friend recommendation.

Our main contributions are threefold.

1) We construct two datasets of real users' online learning data by crawling from the China University MOOC (CUM) and Douban Book, which contain users' non-private information, learning time, and learning materials. The two datasets can help study users' fine-grained interest.

2) We propose a learning partner recommendation framework with fine-grained evolving learning interest (LPRF-E for short). It provides an effective way to model the evolving user interest and improve the recommendation accuracy and quality. We also refine the recommendation with social influence (LPRF-F for differentiation, a variant of LPRF-E). Furthermore, the proposed methods have good universality and can be applied in other OSNs for improving friend recommendation.

3) We conduct extensive experiments and the results show that the proposed LPRF-E and LPRF-F models can effectively improve the performance of learning partner recommendation with approximate 50% improvements on the precision and the recall.

The rest of this paper is organized as follows. In Section 2, related work is briefly outlined. The key problems and basic concepts are given in Section 3. The details of our framework are in Section 4. In Section 5, several experiments are described and designed to validate our framework. Finally, Section 6 illustrates our conclusions.

①https://www.class-central.com/report/moopp.c-stats-2020/, Aug. 2022.

②https://www.coursera.org, Aug. 2022.

③https://www.edx.org, Aug. 2022.

④https://www.icourse163.org, Aug. 2022.

⑤https://book.douban.com, Aug. 2022.

2 Related Work

We briefly review the literature and highlight the key differences of our work from the others.

2.1 Friend Recommendation with Interest Similarity

2.1.1 *Fine-Grained Interest*

Exploiting users' interest similarities is the main direction of friend recommendation. He *et al.* [4] proposed a topic community-based recommendation method. However, existing studies ignore that the users' interest is fine-grained [5], which leads to a lower accuracy. Fine-grained features have been widely used in personalized recommendation systems. Qi *et al.* [6] proposed a hierarchical user interest matching framework to match candidate news with different levels of user interest for more accurate user interest targeting. Wang *et al.* [7] designed a hierarchical fine-grained attention-based network to capture the users' fine-grained interest in order to better recommend the next item. Huang *et al.* [8] analyzed users' aspect preferences from reviews and improved user similarity with users' fine-grained sentiment and product relevance.

2.1.2 *Tagging System*

With the development of Web 2.0, tagging systems are growing rapidly. There are many famous datasets that include social tagging systems, such as Delicious⑥, Last.fm⑦ and Flickr⑧. Tagging systems allow users to annotate, collect and share items by assigning tags. Moreover, tags are associated with both items and users, and they can represent users' fine-grained preferences on items and be used for making personalized recommendations [9]. Inspired by the tagging system, we use it to enhance learning partner recommendation. Lima *et al.* [10] proposed a novel framework that integrates multi-level tag recommendation with external knowledge bases (KBs) to retrieve the most relevant KB articles to answer user-posted questions. Shao *et al.* [11] proposed a friend recommendation method based on fine-grained preference by the tagging system in the photography community. Existing studies using users' interest similarities to construct a graph structure can better explore the users' connections [12, 13].

2.1.3 *Sequential Recommendation*

Sequential recommenders put more emphasis on users' short-term preferences by exploiting information from their recent histories. Graph neural networks are commonly used in sequence recommendation, and Bai *et al.* [14] presented a novel graph neural network framework (GraphRec) for social recommendations. Fan *et al.* [15] provided a principled approach to jointly capture interactions and opinions in the user-item graph, and proposed the framework GraphRec for social recommendations. Gong [16] proposed a novel deep neural network named Graph Convolutional Network Transformer Recommender (GCNTRec), and GCNTRec is capable of learning effective item representation in a user's historical behaviors sequence. Guo *et al.* [17] devised a domain-aware graph convolution network to learn user-specific node representations. Tao *et al.* [18] presented a novel sequential recommendation approach dubbed TRec which learns the item trend information from the implicit user interaction history and incorporates the item trend information into the subsequent item recommendation tasks.

Most recently, pre-trained models like BERT [19] have been proposed to learn the representations of sequences. BERT is a bidirectional model, which is good at learning the representations for users' historical behavior sequences. Sun *et al.* [20] proposed a sequential recommendation model BERT4Rec, exploiting deep bidirectional self-attention to model user behavior sequences.

However, due to the sparse relationship between users in online learning community, graph neural networks are unavailable for learning partner recommendation. In addition, it is difficult to apply BERT directly in the learning partner recommendation task, because it needs a large number of corpora for pre-training when extracting text features, which is unavailable in learning community. Moreover, it is difficult for BERT to consider the time factor, in particular dealing with the irregular time intervals.

2.2 Recommendation Models with Time Evolving Interest

Collaborative filtering (CF) [21] and matrix factorization (MF) [22] are often used in recommendation. Re-

⑥http://files.grouplens.org/datasets/hetrec2011/hetrec2011-delicious-2k.zip, Aug. 2022.

⑦http://files.grouplens.org/datasets/hetrec2011/hetrec2011-lastfm-2k.zip, Aug. 2022.

⑧http://press.liacs.nl:8080/mirflickr/mirflickr25k.v3/mirflickr25k.zip, Aug. 2022.

cently, neural networks and attention mechanism [23] have been exploited.

For short-term session modeling, methods based on the recurrent neural network (RNN) have shown effective performance in sequential recommendation. The long short-term memory (LSTM) model [24] solves the long-term dependence of general RNNs. Yu *et al.* [25] proposed a dynamic recurrent basket model based on RNN, which learns a dynamic representation of users and captures global sequential features among baskets. Lv *et al.* [26] designed a new sequential deep matching model to capture users' dynamic preferences by combining short-term sessions and long-term behaviors. Zhu *et al.* [27] equipped LSTM with time gates to model time intervals. Yu *et al.* [28] improved the RNN structure with a time-aware controller and a semantic-aware controller, to exploit contextual information for the state transition control. Zhao *et al.* [29] proposed a novel short-term memory priority model, which takes into account users' current interest from the short-term memory of the last-clicks. Liu *et al.* [30] prioritized long-term and short-term information in recommendation systems by using adversarial training.

Difference. Although many advanced models have been proposed in many fields, e.g., E-commerce, they usually model the user-item interactions statically and cannot sufficiently capture the dynamic evolution in users' whole behavior sequences. Moreover, in online learning community, the time intervals of user learning behaviors are usually irregular and the time span is longer than that of shopping behaviors. Meanwhile, the semantic distances of fine-grained interest tags are usually different. Therefore, the impacts of time span and semantic distance on users' interest are also worth deeply studying.

3 Problem Definition

3.1 System Settings and Basic Concepts

In this subsection, we formulate the problem we address. We use a triple $I = (U, C, T)$ to represent the system, where U is a set of online learners, C is a set of courses/books and T is a set of learning interest tags. The notations used in this paper are described in Table 1. Some important concepts are described as follows.

Definition 1 (Fine-Grained Interest). *Fine-grained interest is the interest at the fine-grained level. It is usually represented with tags.*

Table 1. Notations

Symbol	Description
U	Set of users
C	Set of courses/books
T	Set of interest tags
R	TextRank value of words
P	PageRank value for tags
F	Social influence of users
T_0	Set of raw tags
T_1	Set of fine-grained interest tags
T_c	Set of candidate tags

For example, if a user learns the course "C++ Programming", the corresponding topic of interest is "computer", while the fine-grained interest can be "C++", "programming" or "grammar".

Definition 2 (Importance of a Tag to a User). *The importance of a tag to a user represents the degree that the tag can reflect the users' interest*[31].

In a tagging system, different tags have different importance to users. For example, user $u \in U$ has two interest tags $t_1, t_2 \in T$. If the frequency of t_1 is greater than that of t_2, then we can say the importance of t_1 is greater than that of t_2 for u.

Definition 3 (Tag Sequence). *A tag sequence is a sequence of tags that are arranged by chronological order.*

Fig.1 shows an example of tag sequence on the user level, which is generated by the courses learning process.

Definition 4 (Social Influence). *Social influence refers to the impact that one can make on others in OSNs, which is a type of peer assessment that has been adopted often in MOOCs due to its immediacy and opportunities for the diversity of feedbacks.*

In online learning community, a user's social influence can be determined by his/her learning experience (e.g., the total learning time, the number of learned courses/books, and the number of discussions).

3.2 Problem of Learning Partner Recommendation

Problem. In this paper, we study the problem of learning partner recommendation considering fine-grained learning interest and their time evolving effects. Given the input data $I = (U, C, T)$, and the target user u_t, the task is to recommend top k proper learning partners for u_t.

Solution Overview. In order to recommend proper learning partners, we propose the LPRF-E framework,

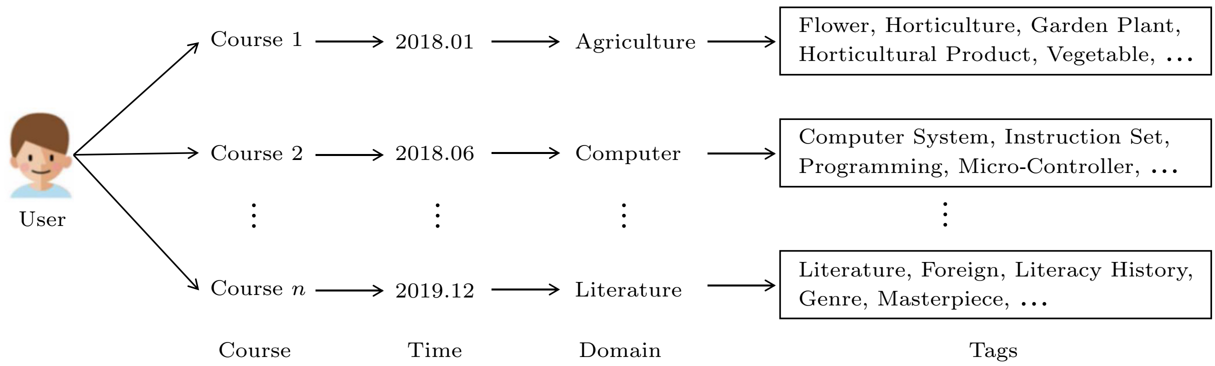


Fig.1. Example of tag sequence generated by course learning.

as shown in Fig.2. It has four main parts: 1) exploiting the introduction text of the course/book to generate the tagging system, 2) extracting the users' fine-grained interest tags from the tagging system, 3) calculating the fine-grained learning interest similarity between users to recommend learning partners, and 4) exploring the temporal evolution of the users' learning interest to generate recommendation. Furthermore, based on the LPRF-E framework, we further refine the recommended list with the social influence, and the variant framework is denoted as LPRF-F for short.

3.3 Dataset and Pre-Processing

The two datasets are crawled from Chinese University MOOC (CUM) and Douban Book, respectively. We crawled users' information and course information, and our crawling lasted for one month during 2020. Because the datasets may involve users' privacy, it is difficult to release the data without the permission. Here we provide our codes for crawling and some samples[9].

In this paper, we focus on users with many learning records, measuring their fine-grained interest similarity directly, and will not deal with cold-start users who have no learning records. Thus, we conduct three main steps for preprocessing. 1) We remove the metadata of cold-start users who have no learning records in the datasets. 2) We delete the course/book data without the introduction text, and extract keywords from the courses/books which have the description text to generate tags. The details of generating tags will be introduced in Subsection 4.1. 3) We generate a tuple of (Course/book, Time, Domain, Tags) for each

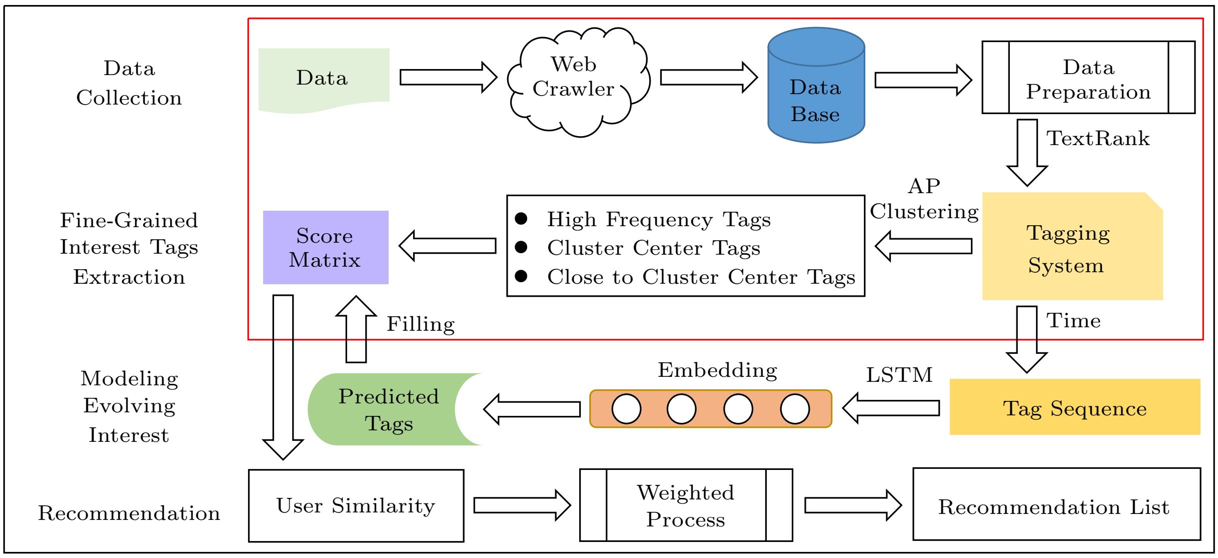


Fig.2. Overview of the LPRF-E framework.

[9]https://gitee.com/coder-wilson/mooc-and-douban-datasets-master, Aug. 2022. The whole data can be available for research use only (please contact authors).

course/book, which presents the name, the end time, the domain, and the tags of each course/book, and we arrange all these tuples in strict order of time. Table 2 shows the statistics of datasets after preprocess.

4 Frameworks for Learning Partner Recommendation

In this section, we introduce the LPRF-E framework in details. We also describe the LPRF-F framework, which refines the recommendation results of LPRF-E with social influence. We first construct the tagging system. Second, we extract fine-grained interest tags from the tagging system. Third, we model users' evolving learning interest. Finally, we select top k users as the candidate learning partners based on interest similarities, and refine the candidate list with users' social influence.

4.1 Constructing the Tagging System

In this paper, we explore the tagging system to enhance learning partner recommendation. In this subsection, we describe how to build a tagging system in the online learning community.

In the original CUM and Douban Book datasets, we can get the Chinese descriptions of the courses/books, which contain many fine-grained keywords of the courses or books. We extract keywords from course/book descriptions to build a tagging system. TextRank [32] is portable to various domains, genres, or languages, because it does not require deep linguistic knowledge, domain, or language-specific annotated corpora. Therefore, we use TextRank to extract keywords from the course/book descriptions.

Fig. 3 shows an example of extracted tags (non-repeating), and we select top m keywords using the following principles. 1) If a word occurs frequently, then it is token as a keyword; 2) if a word w with a high TextRank value is followed by a new word v, then the rank of word v will increase accordingly. The rank of a word v_i, $R(v_i)$, is calculated in an iterative fashion [31]:

$$R(v_i) = (1-d) + d \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_m \in Out(v_j)} w_{jm}} R(v_j),$$

where w_{ji} is the weight between any two points v_j and v_i, $In(v_i)$ and $Out(v_i)$ are the incoming and the outgoing neighbor set of v_i, respectively, and d is the damping coefficient.

We extract the raw tags set T_0^u from the course and book description $c \in C$ of the user $u \in U$, and $T_0^u = \{t_1, t_2, ..., t_m\}$. After that, we combine U, C and T_0 into a raw tagging system of user u. We preprocess the raw tags set T_0^u according to the following principles to get the candidate tags set T_c^u: 1) removing tags without word vectors, 2) deleting stop words⑩, and 3) removing unregistered words (not found in the dictionary, such as person names, and event names).

4.2 Extracting Fine-Grained Interest Tags

4.2.1 Word Vectorized Representation of Tags

In this subsection, we extract the tags that can represent the users' fine-grained learning interest from the

Table 2. Statistics of the Two Datasets

Dataset	Number of Users	Number of Courses/Books	Number of Domains	Number of Tags	Time Span
CUM	8 450	9 031	15	124 107	2014.09—2020.01
Douban	5 500	236 783	6	11 898 099	2007.05—2020.02

This course covers introduction to fruit types and uses, vegetable types and values, flowers and flower classification, horticultural products and safety, horticultural plant variety improvement, and garden construction and cultivation management. Session 1: Introduction to Horticulture. Session 2 : Fruit Types and Utilization. Session 3: Vegetable Types and Characteristics. Session 4: Flower and Flower Classification. Session 5: Horticultural Products and Safety. Session 6: Horticultural Plant Varieties and Breeding. Session 7: Garden construction and cultivation management.

Fig.3. Illustration of learning partners with fine-grained interest tags. The extracted keywords are backgrounded in gray.

⑩https://github.com/goto456/stopwords, Aug. 2022.

raw tagging system T_c^u. Word embedding in a vector space helps the deep learning algorithms to achieve higher efficiency and better performance. In this paper, we use word vectors to represent candidate tags set T_c. Li *et al.*[33] provided Chinese-Word-Vectors[11] trained with different representations, context features (word, n-gram, character, and more), and corpora. We use the 300-dimensional word embedding trained in Baidu Encyclopedia[12] to represent Chinese tags.

4.2.2 *Calculating Similarity Between Tags*

We use the cosine similarity of word embedding to calculate the similarity $\boldsymbol{S}(t_1, t_2)$ between any two tags t_1, $t_2 \in T_c$, and generate a similarity matrix $\boldsymbol{S}$. Let us suppose that $\boldsymbol{x}$ and $\boldsymbol{y}$ are the word vectors of tags t_1 and t_2 generated with the approach in Subsection 4.2.1, respectively. Then, the cosine similarity[34] between t_1 and t_2 is calculated as follows ($n = 300$),

$$\boldsymbol{S}(t_1, t_2) = \frac{\sum\limits_{i=1}^{n} \boldsymbol{x}_i \times \boldsymbol{y}_i}{\sqrt{\sum\limits_{i=1}^{n} (\boldsymbol{x}_i)^2} \times \sqrt{\sum\limits_{i=1}^{n} (\boldsymbol{y}_i)^2}}.$$

4.2.3 *Determining Users' Feature Interest Tags*

In this subsection, we exploit the clustering algorithm to determine the users' feature interest tags, which will be further used to represent the key features of a user's learning interest. Details are as shown in Algorithm 1: 1) clustering tags of user u (lines 1–4), and 2) selecting key tags as the feature interest tags (lines 5–15).

Common clustering algorithms include Gaussian Mixture Model (GMM)[35], KMeans[36], Affinity Propagation (AP) clustering[37], etc. Among them, AP clustering is based on the information transfer between data points, and it does not need to initialize the number of clusters. Therefore, we select the AP clustering algorithm to cluster users' candidate tag set T_c^u.

During AP clustering, the responsibility information and the availability information of the similarity data are iteratively updated, until a stable cluster center is generated. The detailed processes are as follows, where $r(p, q)$ represents the degree to which point q is suitable as the cluster center of data point p, while $a(p, q)$ represents the suitability of point p to select point q as its cluster center.

The responsibility information $r_{i+1}(p, q)$ of the $(i+1)$-th iteration is updated as follows[37],

$$r_{i+1}(p,q) = \begin{cases} \boldsymbol{S}(p,q) - \max\limits_{l \neq q} \{a_i(p,l) + r_i(p,l)\}, \\ \quad \text{if } p \neq q, \\ \boldsymbol{S}(p,q) - \max\limits_{l \neq q} S(p,l), \\ \quad \text{otherwise.} \end{cases} \tag{1}$$

Algorithm 1. Determining Interest Feature Tags

1 Tags similarity matrix $\boldsymbol{S}$, candidate tag set T_c^u of user u, fine-grained interest tags set T_1^u of user u
Initialization: responsibility information $r_1(p,q) \longleftarrow 0$, availability information $a_1(p,q) \longleftarrow 0$, $T_1^u \longleftarrow \emptyset$, $sum_{\text{frequency}} \longleftarrow 0$, $sum_{\text{distance}} \longleftarrow 0$
2 **while** $|r_{i+1}(p,q) - r_i(p,q)|, |a_{i+1}(p,q) - a_i(p,q)| < 0.000\,1$ **or** $\#$ *iterations* $<$ *maxrun* **do**
3   Update $r_{i+1}(p,q)$ using (1)
4   Update $a_{i+1}(p,q)$ using (2)
Result: tag clusters $G = \{G_1, G_2, ..., G_i\}$, supposing CG_i is the center of cluster G_i
5 **for** $G_i \in G$ **do**
6     $T_1^u \longleftarrow \{CG_i\} \cup T_1^u$
7   **for** $t \in G_i$ **do**
8         $sum_{\text{frequency}} \longleftarrow sum_{\text{frequency}} + t.\text{frequency}$
9         $sum_{\text{distance}} \longleftarrow sum_{\text{distance}} + t.\text{distance}$
10   **for** $t \in G_i$ **do**
11   **if** the frequency of tag t is less than the average value in the same cluster **then**
12             $T_1^u \longleftarrow \{t\} \cup T_1^u$
13   **if** the distance of tag t is less than the average value in the same cluster **then**
14             $T_1^u \longleftarrow \{t\} \cup T_1^u$
15 **return** T_1^u

[11] https://github.com/Embedding/Chinese-Word-Vectors, Aug. 2022.
[12] https://baike.baidu.com, Aug. 2022.

The availability information $a_{i+1}(p,q)$ of the $(i+1)$-th iteration is updated as follows[37],

$$a_{i+1}(p,q) = \begin{cases} \min\{0, r_{i+1}(q,q) + \sum\limits_{l\neq p,q} \max\{r_{i+1}(l,q),0\}\}, \\ \quad \text{if } p \neq q, \\ \sum\limits_{l\neq q} \max\{r_{i+1}(l,q),0\}, \\ \quad \text{otherwise.} \end{cases} \quad (2)$$

After performing AP clustering on the candidate tags T_c^u of user u, we obtain N clusters G ($G = G_1, G_2, ..., G_N$). Based on the clustering results, we further select the users' feature interest tags with three approaches as shown in Algorithm 1.

1) *Selecting High Frequency Tags* (H). We select tags with a frequency greater than the average value (line 11 and line 12).

2) *Selecting Cluster Center Tags* (C_1). We select the cluster center CG_i tag in the i-th cluster.

3) *Selecting Tags Close to the Cluster Center* (C_2). We select tags whose distances from the cluster center CG_i are less than the average value (line 13 and line 14).

The distance $Dist(t_1,t_2)$ between tag t_1 and cluster center t_2 is calculated by the Euclidean distance,

$$Dist(t_1,t_2) = \sqrt{\sum_{i=1}^{n}(\boldsymbol{x}_i - \boldsymbol{y}_i)^2},$$

where $n = 300$, and $\boldsymbol{x}$ and $\boldsymbol{y}$ represent the n-dimensional word vectors of the two tags t_1 and t_2 respectively. $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ represent one point in t_1 and t_2 respectively.

After selecting feature interest tags, we analyze their distribution, as shown in Fig.4. We count the number of users corresponding to the number of tags extracted by the clustering algorithm. It shows that the number of feature interest tags is mostly within 600, and the range of [1, 100] accounts for a large proportion.

In this paper, we focus on fine-grained learning interest modeling, and recommending learning partners based on users' interest similarity. We use tags to represent the users' fine-grained learning interest, and fine-grained tags change more significantly than the topic tags. Therefore, it is necessary to analyze the users' learning interest features to further extract fine-grained interest tags.

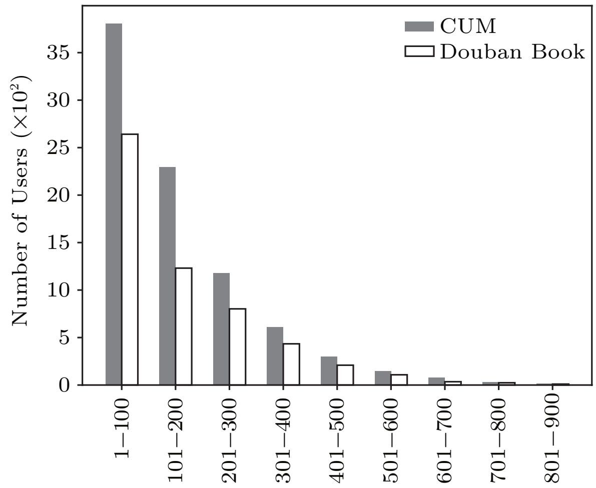


Fig.4. Number of users' feature interest tags.

4.3 Modeling Users' Evolving Learning Interest

In fact, all users' learning interest is continuously changing over a long time span, and the time intervals of user learning behaviors are irregular, as shown in Fig.5, which makes it difficult to model the learning interest over time. Moreover, users are not inclined to learn the same courses/books repeatedly in a short time. Also, there are semantic distances between the fine-grained interest tags, and many unrelated fine-grained tags lead to inaccurate prediction.

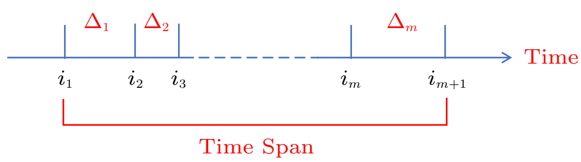
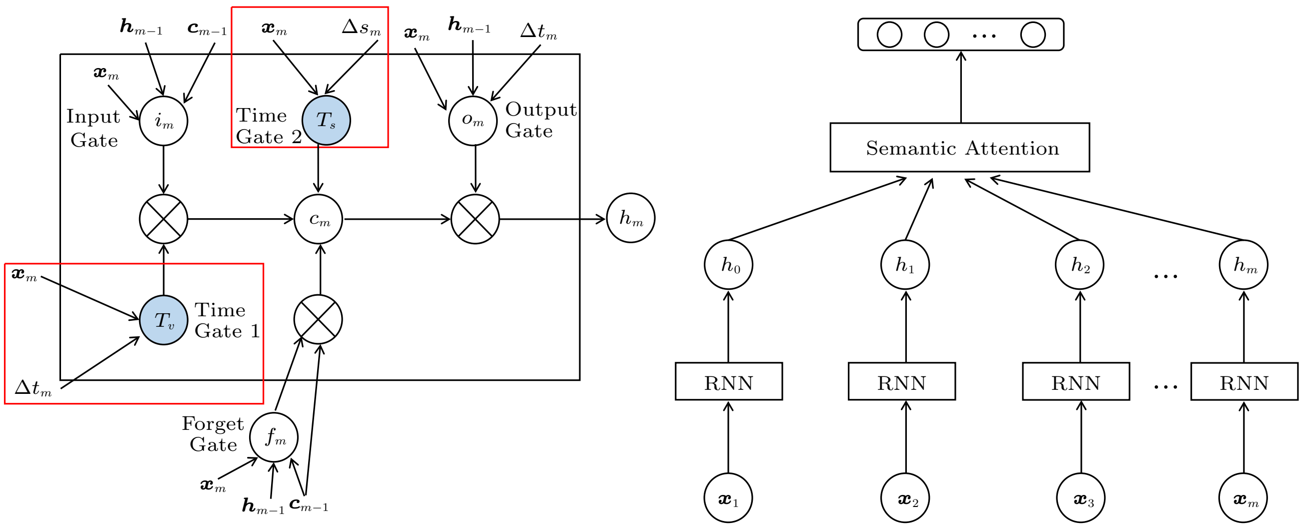


Fig.5. Time irregularity in learning sequence. i_m represents the m-th item (e.g., course and book) and $\triangle m$ is the time interval between i_m and i_{m+1}.

Therefore, we propose the TSA-LSTM model which designs two controllers (as shown in Fig.6): time-aware controller and semantic-aware controller, to model the evolving learning interest of users, and extract more abundant fine-grained interest tags.

4.3.1 *Time-Aware Controller*

LSTM is a time recurrent neural network; however, it cannot capture the relations of tag sequences by time intervals. Keeping the modeling tasks and the challenges in mind, we design a time-aware controller which modifies the logic gates in the LSTM model[27,28]. To



(a) (b)

Fig.6. TSA-LSTM model. (a) Time-aware controller. (b) Semantic-aware controller. $\boldsymbol{c}_m$ is the cell activation vector, and $\boldsymbol{h}_m$ is the hidden output vector.

tackle the problem of time irregularity in the learning process, the time-aware controller exploits two core logic gates: time-interval logic gate T_v and time-span logic gate T_s, as shown in Fig.6(a). T_s is to control the influence of the last tag on current tag learning, and T_v is to store time intervals to model users' long-term interest. The time interval feature v_{t_m} of T_v, and the time span feature s_{t_m} of T_s, are calculated as shown below.

$$\begin{aligned} v_{t_m} &= \phi(W_v \log(t_m - t_{m-1}) + b_v), \\ s_{t_m} &= \phi(W_s \log(t_p - t_m) + b_s), \\ T_v &= \sigma(x_m W_{xv} + v_{t_m} W_{tv} + b_{tv}), \\ T_s &= \sigma(x_m W_{xs} + s_{t_m} W_{ts} + b_{ts}), \end{aligned}$$

where $W_{xv}, W_{xs} \in \boldsymbol{R}^{D\times D}$ are weight parameters, $W_{tv}, W_{ts} \in \boldsymbol{R}^{D\times D}$, D indicates the dimension of the input embedding and the hidden layers, and b_v, b_s, b_{tv} and b_{ts} are the corresponding biases. The time span feature s_{t_m} encodes the absolute temporal distance between the current state t_m and the predictive state t_p, while the time interval feature v_{t_m} encodes the relative temporal distance between two consecutive states.

4.3.2 Semantic-Aware Controller

In the learning community, the sequence of a user's fine-grained learning interest tags covers a variety of semantic information, and the traditional LSTM model cannot determine which semantic features need to be retained. To address the above problems, we design a semantic-aware controller, which adopts the attention mechanisms to dynamically filter out irrelevant tags in a user's tags sequence according to the semantic-aware distance, as shown in Fig.6(b). The m-th item's attention score α_m is calculated by (3),

$$\alpha_m = \frac{\exp(\boldsymbol{x}_m W_{xs} \boldsymbol{e}_p)}{\sum_{j=1}^{|B_u|} \exp(\boldsymbol{x}_j W_{xs} \boldsymbol{e}_p)}, \tag{3}$$

where $\boldsymbol{e}_p$ represents the predicted embedding of the input embedding vector $\boldsymbol{x}_p$, and $|B_u|$ denotes the number of actions in the behavior sequence of user u.

The recommender system in the learning community also meets the challenge of sparsity. In our model, we exploit the predicted learning interest to address the challenge. Matrix filling is a method commonly used to alleviate the sparsity and improve the performance of collaborative filtering. After predicting the users' evolving learning interest, we can obtain the predicted tag (t_p) that user u is interested in. Next, we select tags from the candidate sets whose similarities with t_p are greater than a threshold (e.g., 0.5) and user u has not learned to obtain a set T_{fill}. Finally, we fill the rating of the tags in T_{fill} into the scoring matrix, so as to alleviate the sparsity.

4.4 Recommending Learning Partners

In this subsection, we generate the recommended learning partners list. We first build a user-tag scoring matrix. Second, we calculate users' similarities based on their learning interest. Third, we select top k similar users as the candidate learning partners. Finally, we refine the candidate list with users' social influence.

4.4.1 Building a Scoring Matrix for Users and Tags

In Subsection 4.2, we extract a set of fine-grained learning interest tags T^u for each user u. Before recommending learning partners, we need to establish a scoring matrix of users and feature tags to represent the users' learning interest. In the original data, it may lack the users' score for some tags. In order to better calculate the similarity of learning interest between users in online community, we need to fill the empty scores.

The PageRank algorithm [38] uses the citation number of papers to judge the importance of academic papers. Inspired by the PageRank algorithm, we use the similarity between the tags and the frequency of the tags to calculate the PageRank value $P(t_i)$, which can be taken as the score for tag t_i of user u. $P(t_i)$ is calculated as follows,

$$P(t_i) = \frac{1-d}{N} + d \sum_{t_j \in In(t_i)} \frac{P(t_j)}{Out(t_j)},$$

where $Out(t_j)$ and $In(t_i)$ are the outgoing neighbor set of tags t_j and t_i, respectively, and t_j's similarity with t_i is higher than a threshold. N is the total number of tags in the scoring matrix, and d is the damping coefficient.

4.4.2 Calculating Users' Interest Similarity

In this subsection, we describe how to calculate the similarity of learning interest between the target user (u_t) and other users by a scoring matrix. Based on this, we can select top k candidate learning partners.

The collaborative filtering (CF) algorithm is a commonly-used algorithm in recommendation systems, which can be either user-based or item-based. In this paper, we use the user-based CF for learning partner recommendation. The similarity of the learning interest between users u and v is calculated by Pearson Correlation [34], as follows,

$$\mathrm{sim}(u,v) = \frac{\sum_{i=1}^{N}(u_i-\bar{u})(v_i-\bar{v})}{\sqrt{\sum_{i=1}^{N}(u_i-\bar{u})^2}\sqrt{\sum_{i=1}^{N}(v_i-\bar{v})^2}},$$

where u_i and v_i are the score of tag i given by users u and v respectively, $\bar{u}$ and $\bar{v}$ are the average score for all tags given by user u and v respectively, and N is the total number of tags in the scoring matrix.

In the real world, users with similar learning interest are more likely to become partners. Based on the similarity with the target user u_t's learning interest, we can select the top k similar users. Supposing that sim_{tk} is the similarity of users u_t and u_k, we select and sort the candidate learning partners in descending order according to sim_{tk}. We obtain a primary learning partner recommendation list $RecList_{\mathrm{p}} = \{u_1, u_2, u_3, ..., u_k\}$, where $sim_{t1} \geqslant sim_{t2} \geqslant sim_{t3} \geqslant ... \geqslant sim_{tk}$.

4.4.3 Refining Recommendation with Social Influence

According to studies on social influence [6,39], if a user's social influence is greater, then he/she will be more likely to become friends with strangers. An online learning community is an online social network based on learning users, and the social influence between users often determines whether two users can become friends.

Wang *et al.* [40] explored the relationship between students' final grades and online learning behaviors, and found that students' online learning behaviors, such as time taken to watch videos, and participating in discussions have significant correlation with final examination performance. In this paper, we treat students' learning performance as the social influence in social networks. Thus, we model the social influence of users in online learning communities, and further explore it for refining learning partner recommendation. We denote the variant framework as LPRF-F for differentiation.

In our datasets, a user's information includes the learning time (LT), the number of courses/books (CN) and discussions (DN). In the learning community, CN and LT can represent users' learning interest and proficiency in learning, and DN can represent their activeness. In this paper, we use the above three factors to model the social influence $F_u \in (0,1)$ of user u. Let us suppose that $\widetilde{LT}$, $\widetilde{CN}$ and $\widetilde{DN}$ are the normalized value of LT, CN and DN respectively. Combining LT, CN and DN, we calculate F_u as follows,

$$F_u = w_1 \times \widetilde{LT_u} + w_2 \times \widetilde{CN_u} + w_3 \times \widetilde{DN_u},$$

where $w_1, w_2, w_3 \in (0,1)$ are the weight parameters, and $w_1 + w_2 + w_3 = 1$.

Many methods can be used to determine the weight parameters w_1, w_2, w_3. Here, we take the entropy weight method (EWM) [41] as an example. The main steps are as follows [41], *Step* 1. We normalize the user-factors matrix by the Min-Max Scaling method to obtain the value of μ_{ij},

$$\widetilde{z_{ij}} = \frac{z_{ij} - \min(\boldsymbol{Z}_i)}{\max(\boldsymbol{Z}_i) - \min(\boldsymbol{Z}_i)},$$

$$\mu_{ij} = \frac{\widetilde{z_{ij}}}{\sum\limits_{i=1}^{|U|} \widetilde{z_{ij}}},$$

where z_{ij} is the element in row i and column j of the user-factor matrix $\boldsymbol{Z}$, $\widetilde{z_{ij}}$ is the normalized value of z_{ij}, and $\boldsymbol{Z}_i$ is the set of all the elements in row i. $|U|$ is the number of total users, and μ_{ij} is the proportion of the j-th factor of the i-th user.

Step 2. We calculate the entropy of μ, as follows,

$$E_j = -\frac{\sum\limits_{i=1}^{|U|} \mu_{ij} \ln \mu_{ij}}{\ln |U|},$$

where E_j is the information entropy of the j-th factor, and $|U|$ is the number of total users.

Step 3. We define w_j based on the entropy,

$$w_j = \frac{1 - E_j}{1 - \sum\limits_{j=1}^{3} E_j},$$

where w_j is the weight parameter, $j = \{1, 2, 3\}$.

The values of weights w_1, w_2 and w_3 according to EWM are 0.26, 0.56 and 0.18 respectively, in the CUM dataset; those in the Douban Book dataset are 0.31, 0.48 and 0.21, respectively. After calculating the users' social influence, we exploit social influence to refine the primary recommendation list $RecList_{\rm p}$. We denote SF as the weighted value of F and the interest similarity sim_{tk}, and generate the friend recommendation list ($RecList_{\rm q} = \{v_1, v_2, v_3, ..., v_k\}$, where $SF_{t1} \geqslant SF_{t2} \geqslant SF_{t3} \geqslant ... \geqslant SF_{tk}$) by descending order according to SF, which is calculated as follows,

$$SF_k = sim_{tk} \times (1 - \eta) + F_k \times \eta,$$

where $sim_{tk} \in (0, 1)$ is the similarity of the target user u_t and another user k, and $F_k \in (0, 1)$ is the social influence of user k.

5 Experimental Evaluation

In this paper, we conduct personalized partner recommendation based on users' fine-grained learning interest. In order to find similar users more accurately, we propose the LPRF-E model to extract users' fine-grained interest tags and make recommendation. We also propose the LPRF-F model to refine the friend recommendation with social influence. In this section, we evaluate the proposed methods focusing on five research questions (RQs).

- *RQ*1. What is the effect of the users' feature interest tags on friend recommendation?
- *RQ*2. How does the evolving learning interest improve the recommendation?
- *RQ*3. How is the performance of our LPRF-E and LPRF-F models compared with others?
- *RQ*4. How does users' social influence affect the recommendation results of learning partners?
- *RQ*5. Do learning interest domains impact learning partner recommendation?

5.1 Effects of Users' Feature Interest Tags (RQ1)

In this subsection, we will check the effectiveness of extracting users' fine-grained interest tags from the following three perspectives: 1) the setting of the TextRank algorithm: to find out the optimal setting of the number of extracted keywords in the TextRank algorithm; 2) tags combination: to explore the impact of combination methods of fine-grained interest tags; 3) tags clustering: to validate the effect of the AP clustering algorithm.

5.1.1 *Setting of the TextRank Algorithm*

In order to explore the performance of the TextRank algorithm, we define *Rate* to evaluate the effectiveness of keywords extracted by the TextRank algorithm. It represents the proportion of the fine-grained interest tags over all the keywords, calculated as follows,

$$Rate = \frac{|T_1|}{|T_0|},$$

where $|T_1|$ is the number of fine-grained interest tags, and $|T_0|$ is the total number of keywords extracted by the TextRank algorithm for all users.

As shown in Fig.7, *Rate* decreases as the number of keywords increases, and it decreases gradually. When $T_0 > 30$, *Rate* declines insignificantly. Therefore, we set the number of extracted keywords to 30, which can ensure that important keywords are not lost and minimize the noise at the same time.

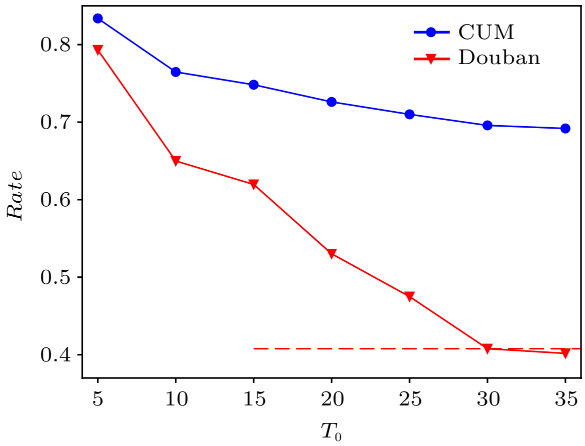


Fig.7. Performance of the TextRank algorithm with respect to the number of extracted tags.

5.1.2 *Tags Combination Evaluation*

We check the impact of tags combination (e.g., H, C_1, C_2 in Subsection 4.2.3) on the learning partner recommendation. We conduct ablation experiments on H, C_1 and C_2 to explore the effect of their seven combinations on the recommendation. As shown in Table 3, all three types (H, C_1, C_2) take effects. Moreover, the high-frequency tags have more effect on the recommendation results, and cluster center tags have the lowest effect. The reason may be that there are more high-frequency tags than the cluster center tags.

5.1.3 *Tag Clustering Evaluation*

We validate the effects of tag clustering with five commonly-used metrics.

1) *Homogeneity*. It measures how much the tags in a cluster are similar to each other regarding their categories [41]. If there is only one category in a cluster, then the homogeneity value is 1. It is calculated as follows,

$$HO = \frac{1}{N}\sum_{i=1}^{N}\frac{L(P_i == Q_i)}{L(Q_i)},$$

where HO is the homogeneity, N is the number of clusters, Q is the predicted sample and P represents the true sample. $L(Q_i)$ represents the number of predicted samples in the i-th cluster, and $L(P_i == Q_i)$ represents the number of correctly-classified samples in the i-th cluster.

2) *Completeness*. Completeness [42] refers to how tags of the same category are classified into the same cluster. If all tags of the same type are grouped into the same cluster, then the completeness is 1. It (CO) is calculated as follows,

$$CO = \frac{1}{N}\sum_{i=1}^{N}\frac{L(P_i == Q_i)}{L(P_i)},$$

where N is the number of clusters, and the main difference of the completeness from the homogeneity is that the denominator is $L(P_i)$, representing the number of true samples in the i-th cluster.

3) *V-Measure*. Considering homogeneity or completeness alone is one-sided, thus the weighted average V-Measure [41] (VM for short) of the two indicators is introduced. It is calculated as follows,

$$VM = \frac{(1+\beta)\times HO \times CO}{\beta \times HO + CO},$$

where β is the weighted parameter. If $\beta > 1$, more attention is paid to completeness (CO), and if $\beta < 1$, more attention is paid to homogeneity (HO).

4) *Mutual-Information*. Mutual-information [43] is used to measure the degree of the correlation between two random clusters' information, that is, the amount of information contained in cluster ρ and another cluster ϱ. It is calculated as follows,

$$MI(\rho, \varrho) = \sum_{i=1}^{|\rho|}\sum_{j=1}^{|\varrho|}\frac{|\rho_i \cap \varrho_j|}{N}\log\frac{N \times |\rho_i \cap \varrho_j|}{|\rho_i||\varrho_j|},$$

where $MI(\rho, \varrho)$ is the mutual-information, and N is the number of clusters. $|\rho_i|$ is the number of samples in cluster ρ_i, and $|\varrho_j|$ is the number of samples in cluster ϱ_j.

5) *Silhouette-Coefficient*. Silhouette-coefficient [44] is used to measure the reasonability of clustering. The closer the value of the silhouette-coefficient is to 1, the

Table 3. Impact of Tag Combination Methods on Recommendation Precision

Case	Top-5	Top-10	Top-15	Top-20	Top-25	Top-30	Top-35
C_1	0.161 9	0.144 8	0.120 4	0.113 5	0.101 8	0.088 2	0.081 1
C_2	0.196 2	0.173 9	0.160 9	0.153 1	0.136 8	0.131 1	0.110 3
$C_1 + C_2$	0.202 7	0.191 3	0.177 5	0.163 7	0.149 5	0.140 7	0.127 1
H	0.300 4	0.284 7	0.257 9	0.211 9	0.192 5	0.183 3	0.169 5
$H + C_1$	0.316 2	0.291 7	0.260 8	0.221 8	0.208 2	0.185 2	0.177 2
$H + C_2$	0.328 5	0.315 5	0.296 2	0.275 3	0.254 7	0.210 2	0.195 3
$H + C_1 + C_2$	0.349 3	0.331 7	0.307 2	0.280 3	0.266 1	0.220 9	0.207 7

more reasonable the clustering of tag t is. It is calculated as follows,

$$SC(t) = \frac{\gamma(t) - \varepsilon(t)}{\max\{\gamma(t), \varepsilon(t)\}},$$

where SC is the silhouette-coefficient, and $\gamma(t)$ refers to the average distance between tag t and other tags in the cluster to which t belongs. $\varepsilon(t)$ is the minimum average distance from tag t to all clusters that do not contain t.

The comparison results are shown in Fig.8. It indicates that the AP clustering performs better than KMeans and GMM for all the five metrics. As far as the CUM dataset is concerned, the five evaluation indicators of the AP clustering algorithm in extracting fine-grained interest tags are 7.20%, 17.49%, 11.96%, 8.77%, and 11.83% higher than the KMeans algorithm respectively. The possible reason is that the AP clustering algorithm does not need to set the number of clusters, and thus it can better generate stable clusters according to the similarity matrix.

5.2 Experimental Results with Evolving Learning Interest (RQ2)

In this subsection, we analyze the effect of evolving learning interest on learning partner recommendation. To verify whether our design of time-aware and semantic-aware controller is necessary and effective for short-term preference modeling, we compare our model with several variants and state-of-the-art models.

- *LSTM*[24]. The classical LSTM model is for sequential prediction.
- *T-LSTM*[27]. It is the LSTM with time gates to model time intervals, does not handle the time span, and relies on the time gates to capture short-term interest.
- *T-SeqRec*[28]. T-SeqRec is a variant of our TSA-LSTM model which only enables the time-aware controller.
- *SDM*[26]. It is a new sequential deep matching model to capture users' dynamic preferences by combining short-term sessions and long-term behaviors.
- *GraphRec*[15]. It is a novel graph neural network framework to jointly capture interactions and opinions in the user-item graph.
- *BERT4Rec*[20]. It is a novel sequential recommendation model that employs the deep bidirectional self-attention to model user behavior sequences.

Given a user's previous T fine-grained interest tags, we want to predict the $T+1$ tag. The task can be taken as a binary classification problem; thus we use AUC[45] and F-score[46] as the evaluation metrics, which can reflect the performance from different aspects.

The results are shown in Table 4. It indicates that all the other models are better than the classical model LSTM, which demonstrates the necessity of considering complex user behavior patterns. Our TSA-LSTM model performs the best, indicating that both the time-aware and the semantic-aware controllers are beneficial to model users' evolving learning interest.

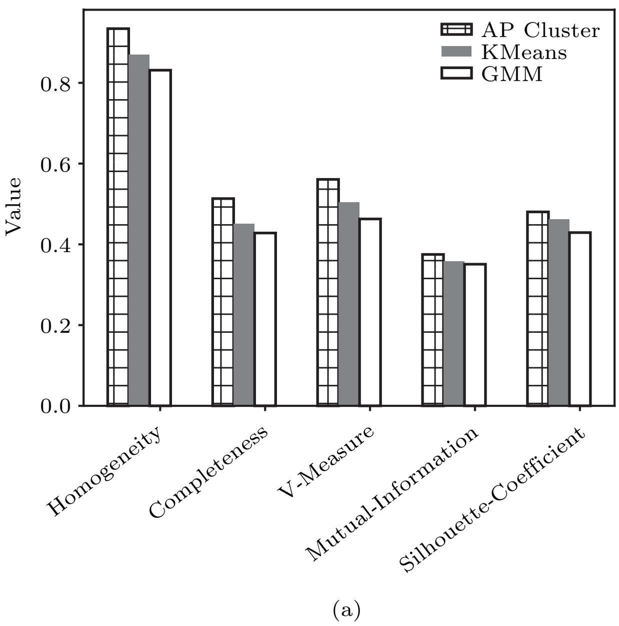


Fig.8. Evaluating the clustering effects on different datasets. (a) CUM. (b) Douban Book.

The T-SeqRec model performs the second best, except on AUC in the CUM dataset where BERT4Rec performs the second. This indicates the importance of our designed time-aware controller. In the other columns, BERT4Rec performs the third best, followed by SDM. The improvements of our model over the second best model range from 6.81% to 14.40%, as shown in Table 4.

Table 4. Comparison of Learning Interest Prediction Models

Model	AUC		*F*-Score	
	CUM	Douban	CUM	Douban
LSTM	0.5681	0.5407	0.3398	0.3090
T-LSTM	0.5713	0.5695	0.3572	0.3441
GraphRec	0.5802	0.5639	0.3791	0.3682
SDM	0.6383	0.6047	0.4725	0.4188
BERT4Rec	0.6519	0.6101	0.4992	0.4318
T-SeqRec	0.6411	0.6183	0.5227	0.4707
TSA-LSTM	0.6898	0.6604	0.5829	0.5385
Improvement	0.0759	0.0681	0.1152	0.1440

We deeply compare the performance and the characteristics of the models, and obtain several important findings. 1) In an online learning community, the time interval of the learning sequence is long, and users will not often learn the same course repeatedly in a short time. This makes the performance of BERT4Rec not so good as that of our TSA-LSTM. 2) GraphRec performs not very well. We analyze the reason and find that GraphRec needs more information of the relationships between users, which are quite sparse in the online learning scenarios.

5.3 Comparative Studies with Top-k Recommendation (RQ3)

In this subsection, we compare the performance of LPRF and LPRF-E on top-k recommendation and seven baselines. We begin with introducing the experimental setup, and then report and analyze the experimental results to answer the research question RQ3.

5.3.1 Initializing Friend Relationship

In MOOCs, there is no explicit friend recommendation function, or explicit friend relationship. In order to evaluate the performance of LPRF-E, we classify the connections between users as direct and indirect friend relationships. The direct friend relationship is the original connection relationship between users, while the indirect friend relationship is determined by the co-learning number of the same courses/books. If the co-learning number is larger than the median value of courses/books learned by all users, we will say the two users are indirect friends. We use the 10-fold cross-validation method to divide the dataset into 90% for the training set and 10% for the test set for experiments.

5.3.2 Evaluation Metrics

In this paper, we use precision and recall as the measurements to evaluate the performance. We take learning partner recommendation as a binary classification problem. Precision and recall are calculated as follows,

$$Precision = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |R(u)|},$$

$$Recall = \frac{\sum_{u \in U} |R(u) \cap T(u)|}{\sum_{u \in U} |T(u)|},$$

where U is the set of all users, $R(u)$ is the list of potential friends recommended to user u by our models, LPRF-E and LPRF-F, and $T(u)$ is the list of friends in the test data, as initialized in Subsection 5.3.1.

5.3.3 Baselines

We implement seven representative methods as the baselines, as follows.

- *RandomRec*. It randomly recommends some users as learning partners to the target user.
- *FRUG*[12]. It is the friend recommendation algorithm by the user similarity graph to find potential friends with the same interest in the social tagging system.
- *UserRec*[13]. It is a user recommendation framework for user interest modeling and interest-based user recommendation, and the similarity values between users' topic distributions are measured by the Kullback-Leibler divergence (KL-divergence) in the social tagging system.
- *NC-basedSFR*[47]. A network correlation based social friend recommendation method, NC-basedSFR, considers the effect of different social roles from different social networks, and exploits the pairwise user similarity.
- *DSFR*[48]. A two-stage method is applied to the unlabeled data in social networks to model users' interest and activities, and recommends friends with similar social behavior patterns.
- *FPAC*[49]. It is collaborative filtering based on the combination of interest and cognition to improve the effect of friend recommendation.

• *IACFM*[50]. It is a new hybrid technology based on FP-growth and the ant colony optimization algorithm to improve friend recommendation performance in the social tagging system.

The number of learning partners recommended for a target user (u_t) is 10, 20 and 30 respectively, and the experimental results on the CUM and Douban Book datasets are shown in Table 5 and Table 6, respectively.

We compare the performance of LPRF-F and FRUG, and the improvements are shown in the last row. It shows that, our models LPRF, LPRF-E and LPRF-F perform better than the other baselines. For example, in terms of Precision@10, the LPRF-F model is 84.09% and 51.51% higher than the FRUG model on the CUM and Douban Book datasets, respectively. Moreover, LPRF-F which considers time-evolving fine-grained interest and social influence achieves the best performance. For instance, the improvement of LPRF-F over LPRF-E on Precision@10 is 39.29% on CUM, and 20.03% on Douban Book.

We further analyze the possible reasons, that why randomly recommending (RandomRec) learning partners for u_t has the worst performance followed by NC-basedSFR, UserRec, and FRUG. NC-basedSFR recommends friends based on the correlation between the user network and the tag network, and considers the users' social roles. Its performance is poor because the user network in the learning community is sparse. UserRec exploits user interest topics to calculate the interest similarity between users. However, it is a coarse-grained method, which leads to poor performance. FRUG uses the LDA algorithm to extract the users' interest topics, and uses the graph structure to model the users' interest similarity, which has better performance than the other baselines. It indicates that the graph structure can improve friend recommendation.

Furthermore, we analyze the statistical significance of the proposed model on Precision@10 and Recall@10. As shown in Fig.9, we calculate the average and variance of the results after running 10–50 iterations. It shows that the mean value of the recommended results of the LPRF-F model fluctuates within a reasonable range and becomes stable after 40 iterations.

Table 5. Comparison of Recommended Performance on CUM

Method	Precision@10	Recall@10	Precision@20	Recall@20	Precision@30	Recall@30
RandomRec	0.120 6	0.054 1	0.101 9	0.059 2	0.079 4	0.047 7
NC-basedSFR	0.180 7	0.071 3	0.159 1	0.115 3	0.130 6	0.149 1
DSFR	0.188 3	0.097 4	0.168 8	0.120 4	0.153 7	0.139 8
FPAC	0.190 7	0.090 1	0.172 5	0.113 8	0.159 1	0.121 9
IACFM	0.217 2	0.118 3	0.207 4	0.120 6	0.166 3	0.142 7
UserRec	0.223 8	0.109 4	0.182 7	0.126 9	0.156 8	0.157 1
FRUG	0.250 9	0.141 1	0.210 6	0.157 5	0.182 6	0.171 4
LPRF	0.311 7	0.177 1	0.280 3	0.229 3	0.241 9	0.230 9
LPRF-E	0.331 6	0.216 3	0.319 3	0.248 2	0.284 7	0.261 1
LPRF-F	0.461 9	0.290 8	0.356 6	0.346 3	0.302 8	0.369 2
Improvement	0.840 9	1.060 9	0.693 3	1.198 7	0.858 2	1.154 0

Table 6. Comparison of Recommended Performance on Douban Book

Method	Precision@10	Recall@10	Precision@20	Recall@20	Precision@30	Recall@30
RandomRec	0.157 7	0.068 8	0.140 1	0.072 2	0.048 8	0.093 8
NC-basedSFR	0.212 7	0.099 5	0.190 2	0.117 4	0.135 1	0.144 9
DSFR	0.225 9	0.108 3	0.217 7	0.120 4	0.158 5	0.139 3
FPAC	0.228 6	0.115 7	0.205 3	0.133 8	0.173 4	0.148 4
IACFM	0.230 7	0.121 6	0.209 7	0.147 2	0.180 3	0.158 2
UserRec	0.249 1	0.127 9	0.227 3	0.155 3	0.179 2	0.162 9
FRUG	0.297 4	0.159 5	0.250 9	0.173 7	0.221 8	0.196 2
LPRF	0.340 3	0.192 2	0.311 1	0.228 8	0.275 3	0.247 1
LPRF-E	0.375 4	0.249 1	0.338 6	0.265 9	0.305 2	0.281 4
LPRF-F	0.450 6	0.307 2	0.359 2	0.351 1	0.384 1	0.330 7
Improvement	0.515 1	0.926 0	0.431 6	1.021 3	0.731 7	0.685 5

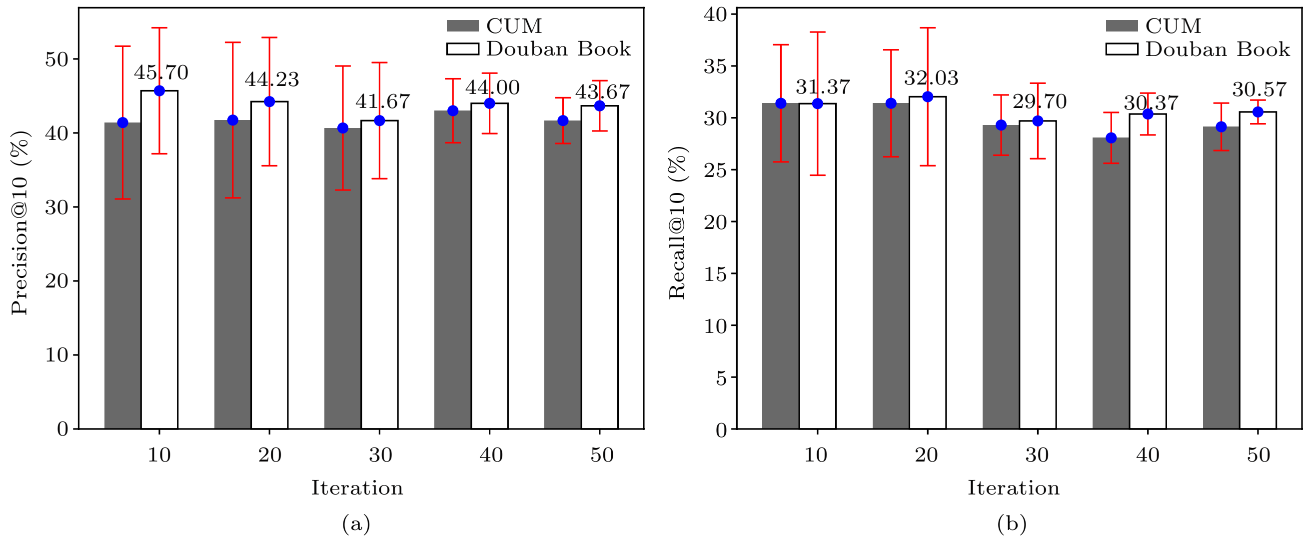


Fig.9. Average and variance of the performance of LPRF-F. (a) Precision@10. (b) Recall@10.

In summary, the comparison study validates the importance of considering time fine-grained evolving interest and social influence.

5.4 Experimental Results with Social Influence (RQ4)

In order to explore the impact of social influence SF on the recommendation results, we use the Jaccard similarity coefficient (J) to calculate the similarity of the recommendation list generated by the two methods of LPRF-E and LPRF-F (which recommend the learning partners list by SF). It is calculated as follows,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|},$$

where A is the recommendation list generated by using SF weighting generated in Subsection 4.2.3, and B is the original learning partner recommendation list generated in Subsection 4.2.2.

As shown in Fig. 10, we recommend top-k ($k \in [5, 35]$) learning partners list by LPRF-E and LPRF-F. We can see that $J \in [0.2, 0.6]$, which indicates F_u has a great influence on the recommendation list.

In Subsection 4.4.3, we define SF to integrate the fine-grained interest and social influence, where parameter η determines the importance of social influence in learning partner recommendation. Here, we set up a comparative experiment with $\eta \in [0, 0.8]$ to explore how parameter η affects the learning partner recommendation. The results are shown in Fig.11.

We can see that when $\eta \in [0, 0.5]$, users' influence can greatly improve the precision of learning partners' recommendation. For instance, in the CUM dataset, LPRF-F ($\eta = 0.4$) is 39.29% higher than LPRF-E on Precision@10, and the improvement of LPRF-F is 20.03% in Douban Book. However, when $\eta > 0.5$, the precision of learning partner recommendation will be reduced. Moreover, with the increase of η, the precision decreases gradually. The possible reason is that a large η will overlook the importance of fine-grained user interest. Therefore, when conducting learning partner recommendation with social influence, η should not be too large.

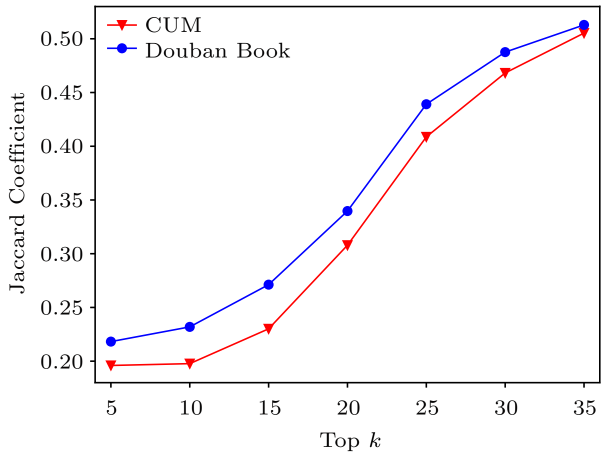
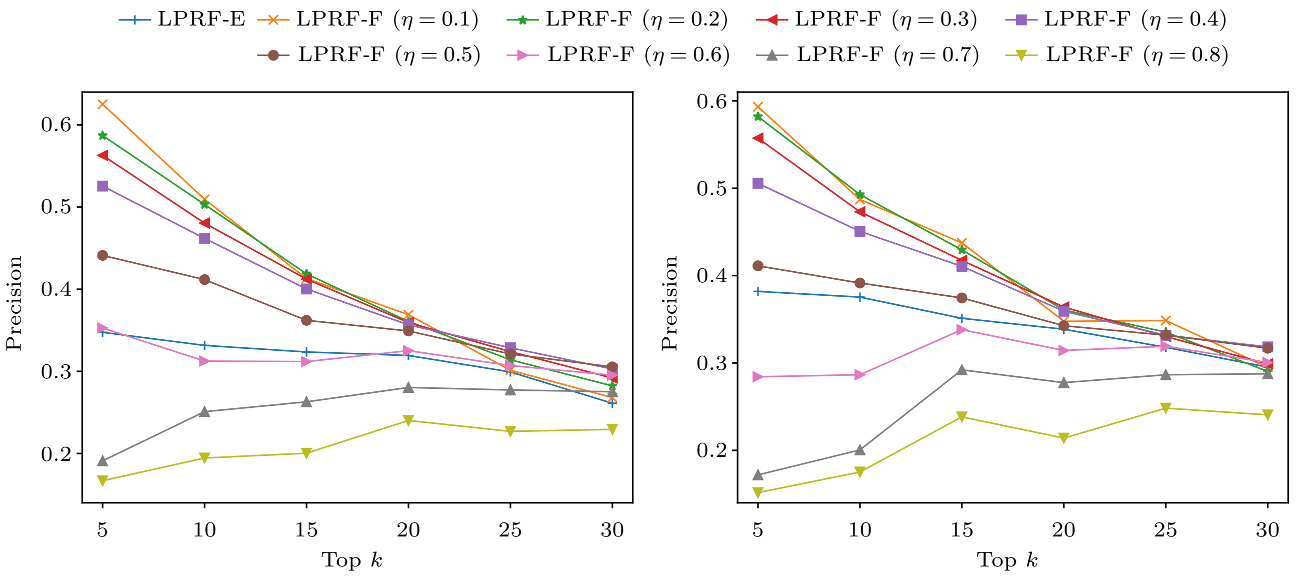


Fig.10. Analysis of the Jaccard coefficient on the learning partner recommendation list.

The above comparison validates that social influence can improve the accuracy of learning partner recommendation. In addition, when two candidate partners have the equal interest similarity to the target user



(a) (b)

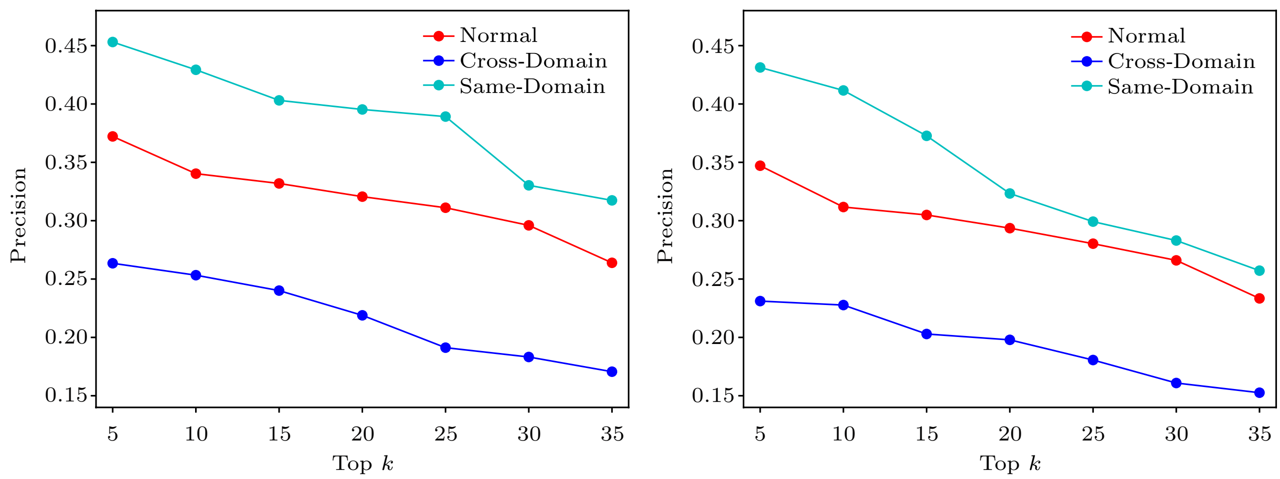
Fig.11. Exploring the impact of η on the precision of learning partners recommendation. (a) CUM. (b) Douban Book.

u_t, the one with higher social influence is usually more helpful. Therefore, he/she should have a higher priority to be recommended as a potential learning partner.

5.5 Experimental Results with Interest Domains (RQ5)

In this subsection, we check the effects of interest domains on learning partner recommendation. We divide users according to the interest domains in which they learn the courses/books the most frequently.

According to whether the potential friend has the same interest domain as the target user, the learning partners can be divided into the same-domain, cross-domain and normal ones. Based on LPRF-E, the proposed model, we recommend learning partners for the target user (u_t) in the same-domain and the cross-domain. We use normal recommendation (i.e., LPRF-E) as a baseline to explore the impact of the learning domain on precision, and the results are shown in Fig. 12. It shows that for the precision on the CUM and Douban Book datasets, the recommendation in the same-domain is the highest, and the cross-domain is the lowest. From the experimental results, we can clearly see that the learning interest domain has a significant



(a) (b)

Fig.12. Impact of the same-domain and the cross-domain on the precision of learning partners recommendation. (a) CUM. (b) Douban Book.

impact on friend recommendation.

On the one hand, it has been verified again that the users' learning interest usually falls in limited domains (that is, only learning in a few domains). On the other hand, learning partner recommendation should expand users' vision as much as possible to avoid the phenomenon of "Information Cocoons" (that is, users prefer to stay within the scope of familiar knowledge). Meanwhile, the proposed LPRF-E model comprehensively considers the users' multiple fine-grained interest features, which are extracted from the users' historical learning records in multiple domains. Therefore, it can take into account the recommendation of learning partners both in the same-domain and in the cross-domain.

6 Conclusions

In this paper, we proposed a comprehensive framework to explore and exploit fine-grained evolving interest, to improve friend recommendation in OSNs. We first extracted the users' fine-grained interest tags with AP clustering, and modeled the users' evolving learning interest with an improved TSA-LSTM model. Then we generated candidate friends list and refined it with social influence. Extensive experiments validated that considering fine-grained evolving interest helps to improve both the accuracy and the quality of friend recommendation. To be specific, the improvements are approximately 50% on precision and recall; while the recommended friends are more experienced and more helpful. We also tested the effects of social influence and cross-domain interest, and gained some interesting findings: 1) social influence can improve the accuracy of learning partners recommendation (e.g., 39.29% higher on precision than that of LPRF-E in the CUM dataset and 20.03% higher in the Douban Book dataset), 2) the cross-domain interest helps to expand users' vision, and our method can recommend both in-domain and cross-domain friends. In future work, we are interested in exploring fine-grained evolving interest in other personalized recommendation scenarios. It is also an interesting direction to integrate the users' behaviors and recommend similar and complementary friends.

References

[1] Sakulwichitsintu S, Colbeck D, Ellis L *et al.* A peer learning framework for enhancing students' learning experiences in online environments. In *Proc. the 18th IEEE International Conference on Advanced Learning Technologies*, July 2018, pp.168-169. DOI: 10.1109/ICALT.2018.00123.

[2] Zewail-Foote M. Pivoting an upper-level, project-based biochemistry laboratory class to online learning during COVID-19: Enhancing research skills and using community outreach to engage undergraduate students. *Journal of Chemical Education*, 2020, 97(9): 2727-2732. DOI: 10.1021/acs.jchemed.0c00543.

[3] Gong J B, Zhao Y, Chen S *et al.* Hybrid deep neural networks for friend recommendations in edge computing environment. *IEEE Access*, 2020, 8: 10693-10706. DOI: 10.1109/ACCESS.2019.2958599.

[4] He C B, Li H C, Fei X *et al.* A topic community-based method for friend recommendation in large-scale online social networks. *Concurrency and Computation: Practice and Experience*, 2017, 29(6): Article No. e3924. DOI: 10.1002/cpe.3924.

[5] Cheng Z Y, Ding Y, Zhu L *et al.* Aspect-aware latent factor model: Rating prediction with ratings and reviews. In *Proc. the 2018 World Wide Web Conference*, April 2018, pp.639-648. DOI: 10.1145/3178876.3186145.

[6] Qi T, Wu F Z, Wu C H *et al.* HieRec: Hierarchical user interest modeling for personalized news recommendation. In *Proc. the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, August 2021, pp.5446-5456. DOI: 10.18653/v1/2021.acl-long.423.

[7] Wang H Z, Liu G F, Liu A *et al.* DMRAN: A hierarchical fine-grained attention based network for recommendation. In *Proc. the 28th International Joint Conferences on Artificial Intelligence Organization*, August 2019, pp.3698-3704. DOI: 10.24963/ijcai.2019/513.

[8] Huang C L, Jiang W J, Wu J *et al.* Personalized review recommendation based on users' aspect sentiment. *ACM Transactions on Internet Technology*, 2020, 20(4): Article No. 42. DOI: 10.1145/3414841.

[9] Jiang W J, Chen J, Ding X F *et al.* Review summary generation in online systems: Frameworks for supervised and unsupervised scenarios. *ACM Transactions on the Web*, 2021, 15(3): Article No. 13. DOI: 10.1145/3448015.

[10] Lima E, Shi W S, Liu X M *et al.* Integrating multilevel tag recommendation with external knowledge bases for automatic question answering. *ACM Transactions on Internet Technology*, 2019, 19(3): Article No. 34. DOI: 10.1145/3319528.

[11] Shao M M, Jiang W J, Zhang L. FRFP: A friend recommendation method based on fine-grained preference. In *Proc. the 7th International Conference on Smart City and Informatization*, November 2019, pp.35-48. DOI: 10.1007/978-981-15-1301-5_4.

[12] Wu B X, Xiao J, Chen J M. Friend recommendation by user similarity graph based on interest in social tagging systems. In *Proc. the 11th International Conference on Intelligent Computing*, August 2015, pp.375-386. DOI: 10.1007/978-3-319-22053-6_41.

[13] Zhou T C, Ma H, Lyu M R, King I. UserRec: A user recommendation framework in social tagging systems. In *Proc. the 24th AAAI Conference on Artificial Intelligence*, July 2010, pp.1486-1491. DOI: 10.1609/aaai.v24i1.7524.

[14] Bai T, Zhang Y, Wu B *et al.* Temporal graph neural networks for social recommendation. In *Proc. the 2020 IEEE International Conference on Big Data*, December 2020, pp.898-903. DOI: 10.1109/BigData50022.2020.9378444.

[15] Fan W Q, Ma Y, Li Q *et al.* Graph neural networks for social recommendation. In *Proc. the 2019 World Wide Web Conference*, May 2019, pp.417-426. DOI: 10.1145/3308558.3313488.

[16] Gong J B. Sequential recommendation through graph neural networks and transformer encoder with degree encoding. *Algorithms*, 2021, 14(9): Article No. 263. DOI: 10.3390/a14090263.

[17] Guo L, Tang L, Chen T *et al.* DA-GCN: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. In *Proc. the 30th International Joint Conference on Artificial Intelligence*, August 2021, pp.2483-2489. DOI: 10.24963/ijcai.2021/342.

[18] Tao Y, Wang C, Yao L *et al.* Item trend learning for sequential recommendation system using gated graph neural network. *Neural Computing and Applications*, 2021. DOI: 10.1007/s00521-021-05723-2.

[19] Devlin J, Chang M W, Lee K *et al.* BERT: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/pdf/1810.04805.pdf, Nov. 2022.

[20] Sun F, Liu J, Wu J *et al.* BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proc. the 28th ACM International Conference on Information and Knowledge Management*, November 2019, pp.1441-1450. DOI: 10.1145/3357384.3357895.

[21] Sarwar B, Karypis G, Konstan J *et al.* Item-based collaborative filtering recommendation algorithms. In *Proc. the 10th International Conference on World Wide Web*, May 2001, pp.285-295. DOI: 10.1145/371920.372071.

[22] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer*, 2009, 42(8): 30-37. DOI: 10.1109/MC.2009.263.

[23] He X N, Liao L Z, Zhang H W *et al.* Neural collaborative filtering. In *Proc. the 26th International Conference on World Wide Web*, April 2017, pp.173-182. DOI: 10.1145/3038912.3052569.

[24] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735-1780. DOI: 10.1162/neco.1997.9.8.1735.

[25] Yu F, Liu Q, Wu S *et al.* A dynamic recurrent model for next basket recommendation. In *Proc. the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 2016, pp.729-732. DOI: 10.1145/2911451.2914683.

[26] Lv F Y, Jin T W, Yu C L *et al.* SDM: Sequential deep matching model for online large-scale recommender system. In *Proc. the 28th ACM International Conference on Information and Knowledge Management*, November 2019, pp.2635-2643. DOI: 10.1145/3357384.3357818.

[27] Zhu Y, Li H, Liao Y K *et al.* What to do next: Modeling user behaviors by time-LSTM. In *Proc. the 26th International Joint Conference on Artificial Intelligence*, August 2017, pp.3602-3608. DOI: 10.24963/ijcai.2017/504.

[28] Yu Z P, Lian J X, Mahmoody A *et al.* Adaptive user modeling with long and short-term preferences for personalized recommendation. In *Proc. the 28th International Joint Conference on Artificial Intelligence*, August 2019, pp.4213-4219. DOI: 10.24963/ijcai.2019/585.

[29] Zhao W, Wang B Y, Ye J B *et al.* PLASTIC: Prioritize long and short-term information in top-n recommendation using adversarial training. In *Proc. the 27th International Joint Conference on Artificial Intelligence*, July 2018, pp.3676-3682. DOI: 10.24963/ijcai.2018/511.

[30] Liu Q, Zeng Y F, Mokhosi R *et al.* STAMP: Short-term attention/memory priority model for session-based recommendation. In *Proc. the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2018, pp.1831-1839. DOI: 10.1145/3219819.3219950.

[31] Lei K, Fu Q A, Yang M *et al.* Tag recommendation by text classification with attention-based capsule network. *Neurocomputing*, 2020, 391: 65-73. DOI: 10.1016/j.neucom.2020.01.091.

[32] Mihalcea R, Tarau P. TextRank: Bringing order into texts. In *Proc. the 2004 Conference on Empirical Methods in Natural Language Processing*, July 2004, pp.404-411.

[33] Li S, Zhao Z, Hu R F *et al.* Analogical reasoning on Chinese morphological and semantic relations. In *Proc. the 56th Annual Meeting of the Association for Computational Linguistics*, July 2018, pp.138-143. DOI: 10.18653/v1/P18-2023.

[34] Nahler G. Pearson correlation coefficient. In *Dictionary of Pharmaceutical Medicine*, Nahler G (ed.), Springer, 2009, pp.132. DOI: 10.1007/978-3-211-89836-9_1025.

[35] Nir F, Stuart R. Image segmentation in video sequences: A probabilistic approach. In *Proc. the 13th Conference on Uncertainty in Artificial Intelligence*, August 1997, pp.175-181.

[36] Hartigan J A, Wong M A. A K-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 1979, 28(1): 100-108. DOI: 10.2307/2346830.

[37] Frey B J, Dueck D. Clustering by passing messages between data points. *Science*, 2007, 315(5814): 972-976. DOI: 10.1126/science.1136800.

[38] Page L, Brin S, Motwani R *et al.* The PageRank citation ranking: Bringing order to the web. Technical Report, Stanford InfoLab, 1998. http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf, Jul. 2022.

[39] Huo Y F, Chen B L, Tang J *et al.* Privacy-preserving point-of-interest recommendation based on geographical and social influence. *Information Sciences*, 2021, 543: 202-218. DOI: 10.1016/j.ins.2020.07.046.

[40] Wang J, Che Y, Li D. Research on the relation-ship between students' final grade and online learning behavior in blended learning model: Taking business Ethics and CSR course as an example. In *Proc. the 2nd International Conference on Education, Knowledge and Information Management*, January 2021, pp.159-166. DOI: 10.1109/ICEKIM52309.2021.00043.

[41] Wang T C, Lee H D. Developing a fuzzy TOPSIS approach based on subjective weights and objective weights. *Expert Systems with Applications*, 2009, 36(5): 8980-8985. DOI: 10.1016/j.eswa.2008.11.035.

[42] Rosenberg A, Hirschberg J. V-Measure: A conditional entropy-based external cluster evaluation measure. In *Proc. the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, June 2007, pp.410-420.

[43] Vinh N X, Epps J, Bailey J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 2010, 11: 2837-2854.

[44] Rousseeuw P J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987, 20: 53-65. DOI: 10.1016/0377-0427(87)90125-7.

[45] Lobo J, Jiménez-Valverde A, Real R. AUC: A misleading measure of the predictive distribution models. *Global Ecology and Biogeography*, 2008, 17(2): 145-151. DOI: 10.1111/j.1466-8238.2007.00358.x.

[46] Goutte C, Gaussier R. A probabilistic interpretation of precision, recall and *F*-score, with implication for evaluation. In *Proc. the 27th European Conference on Information Retrieval Research*, March 2005, pp.345-359. DOI: 10.1007/978-3-540-31865-1_25.

[47] Huang S R, Zhang J, Wang L *et al*. Social friend recommendation based on multiple network correlation. *IEEE Transactions on Multimedia*, 2016, 18(2): 287-299. DOI: 10.1109/TMM.2015.2510333.

[48] Qader S A, Abbas A R. Dual-stage social friend recommendation system based on user interests. *Iraqi Journal of Science*, 2020, 61(7): 1759-1772. DOI: 10.24996/ijs.2020.61.7.25.

[49] Yin Y, Feng X. Friend recommendation algorithm based on interest and cognition combined with feedback mechanism. In *Proc. the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, August 2019, pp.1025-1030. DOI: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00199.

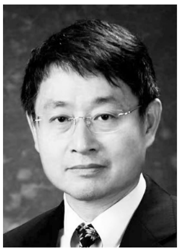
[50] Usman U B, Umar K. Toward a hybrid technique for friends recommendation system in social Tagging. In *Proc. the 3rd International Engineering Conference*, September 2019, pp.404-410.

Ming-Min Shao got his Bachelor's degree in e-commerce from Jiangxi Agricultural University, Nanchang, in 2018, and his Master's degree in computer technology from Hunan University, Changsha, in 2021. His main research directions are online social network analysis and data mining.

Wen-Jun Jiang received her Bachelor's degree in computer science from Hunan University, Changsha, in 2004, her Master's and Ph.D. degrees in computer software and theory from the Huazhong University of Science and Technology, Wuhan, and the Central South University, Changsha, in 2007 and 2014, respectively. Currently, she is a professor and Ph.D. supervisor of Hunan University, Changsha, and a senior member of CCF. Her main research interests include social network analysis and mining, recommendation system, and smart education and learning.

Jie Wu is an IEEE/AAAS Fellow, and a Laura H Carnell Professor of Temple University, Philadelphia. His current research interests include network trust and security, and mobile and wireless networks. He is the recipient of the 2011 CCF Overseas Outstanding Achievement Award.

Yu-Qing Shi got her Master's degree in computer technology from Northeastern University, Shenyang, in 2021. Her main research directions are online social network analysis, video classification and data mining.



TakShing Yum is an IEEE Fellow, and a professor of the Chinese University of Hong Kong, Hong Kong. His main research interests include communication networks, smart education, and learning optimization.

Ji Zhang is a professor of the University of Southern Queensland (USQ), Brisbane, and is an IEEE senior member. His main research interests include knowledge discovery, data mining and big data analytics.