

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A group-oriented recommendation algorithm based on similarities of personal learning generative networks

Haiping Zhu^{1,2,3}, Member, IEEE, Yifu Ni^{1,2}, Feng Tian^{2,4*}, Member, IEEE, Pei Feng^{2,3}, Yan Chen^{1,2,3}, Member, IEEE, Qinghua Zheng^{1,2,3}, Member, IEEE

¹Department of Computer Science and Technology, Xi'an Jiaotong University, 710049, China

²MOEKLINNS Lab, Department of Computer Science and Technology, Xi'an Jiaotong University, 710049, China

³School of Electronic and Information Engineering, Xi'an Jiaotong University, 710049, China

⁴Systems Engineering Institute, Xi'an Jiaotong University, 710049, China

Corresponding author: Feng Tian (e-mail: fengtian@mail.xjtu.edu.cn).

This work was supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004500; the National Science Foundation of China under Grant No. 61472315; Innovative Research Group of the National Natural Science Foundation of China(61721002); Innovation Research Team of Ministry of Education (IRT_17R86); project of China Knowledge Centre for Engineering Science and Technology.

ABSTRACT To solve the lack of consideration of the learning time sequence and knowledge dependencies in group-based recommendation, we proposed a novel group-oriented recommendation algorithm based on learning generative networks, which are characterized by mapping the user's learning log to a personal learning generative network (PLGN) based on a knowledge map. In this paper, we first provide calculation methods of similarity and temporal correlation between knowledge points, where we provide the construction method of the PLGN. Second, we propose a method for measuring the similarities between any two PLGNs. According to the similarities, we perform the CURE clustering algorithm to obtain learning groups. Third, based on the group clustering, we generate the group learning generative network (GLGN) using a graph overlay method. We calculate the importance of the vertices on the different learning needs and propose a group-oriented recommendation algorithm. Finally, we compare the effect of the proposed recommendation to that of a group-based collaborative filtering recommendation for the aspects of precision rate, recall rate, normalized discounted cumulative gain (NDCG) and average accuracy of parameters (MAP). The experimental results show that the group-oriented learning recommendation based on the learning generated network outperforms the group recommendation-based collaborative filtering when the amount of data is large enough.

INDEX TERMS Graph similarity, Group recommendation, Knowledge map, Learning generative network

I. INTRODUCTION

Most recommendation systems focus on recommendations for a single user. However, in many applications, it is necessary to make recommendations for groups. In recent years, the group recommendation system has gradually become one of the hot topics in the field of recommendation systems. It is the main task of group recommendation to integrate group members' preferences[1,2,3]. In particular, the users' interest preferences will change during the learning process. It is necessary to update the corresponding recommendations in time. However, unlike movies, books, and music recommendations, learning resources contain rich semantic and cognitive relationships. Therefore, the

recommended resource sequence directly affects the user's cognitive learning. Although the most widely used collaborative filtering algorithm can work on learning resource recommendations, it mainly uses score information but ignores the temporal and relational information of the recommended items, which can be of equal importance[4]. In recent years, research on recommender systems has shown that knowledge maps are beneficial for hybrid recommender systems, which concerning the semantic relationships of the resources. Therefore, how to use the time sequence information and resource semantic relationships effectively and how to improve the accuracy of the group-oriented recommendation algorithm are the core issues of this paper.

In this paper, we propose a group-oriented recommendation algorithm based on a novel learning generative network that is characterized by mapping the users' learning logs to a personal learning generative network (PLGN), which provides the retention of learning time sequences and knowledge dependencies based on a knowledge map. Then, we construct the group learning generative network (GLGN) based on group clustering, which will duly capture group learners' interests, to provide flexible and accurate recommendations for the group.

A. PROBLEM PRESENTATION

The basic idea of the learning generative network is to transform learning logs into a graph model, in which the vertices are learning resources and the edges are learning relationships between resources. By analyzing the time slice of the learning logs on a Java course, we extract the user vertices and the resource sequences and then construct the corresponding learning generative network (see the fourth chapter of the specific construction method). In this paper, the learning resources represent the knowledge points. Fig. 1 shows the different PLGNs of the same user on different time slices. Fig. 2 shows different PLGNs of different users at the same time slice. Among these figures, the redder vertices represent earlier learning times.

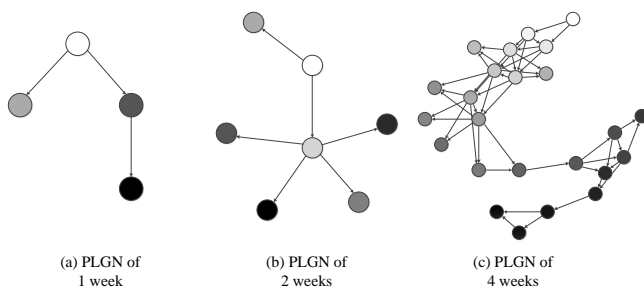


FIGURE 1. An example of constructing the personal learning generative network (PLGN) in different time slices

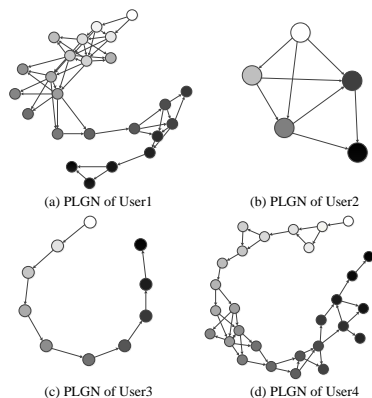


FIGURE 2. An example of constructing the personal learning generative network (PLGN) of different users in the same time slice

From Fig. 1 and Fig. 2, it can be seen that the learning generative networks of the same user on different time slices are different in structure. On the same time slice, the learning generative networks of different users are also not the same at all, but it is obvious that some personal learning generative network structures are similar. This paper mainly examines how to make group learning subgraph recommendations based on an analysis of the group learning characteristics and group learning requirements with a reasonable time slice.

B. INNOVATION

The innovation of this study mainly includes the following:

The main idea of the PLGN is to map the temporal learning logs into a computable network model that takes full account of the learning time sequences and diverse users' behavior. First, learning logs are divided by different data quantities to generate the PLGNs. This approach will reflect the users' dynamic learning processes. Second, different from the existing personal learning networks [5][6][7], which contain only user vertices and edges based on social relationships, the vertices in PLGNs represent the knowledge points. By extracting the users' learning behavior related to the knowledge points, the edge weights between the knowledge points and the sequence relationships of the edges are calculated, in such a way as to embody the diverse learning processes of different users in the PLGNs.

A similarity calculation method based on PLGN content and structure is proposed by means of graph kernel theory[8], which allows user similarities to be calculated. The users are divided into learning groups by using the CURE clustering method. Most of the existing grouping methods rely on the users' explicit or implicit scores on the learning resources, but they seldom consider the users' temporal resource access preferences during the learning process. In this paper, a clustering method based on graph similarities of different time slices is proposed to solve this problem.

The group learning generative network is constructed to represent the common learning characteristics of the user group. A multi-constraint model for vertex importance is used for pertinent subgraph recommendations. The key point is to obtain the common preferences of the group members, to alleviate the preference conflicts among group members, and, finally, to make the recommendation results satisfactory for as many group members as possible. Based on the graph fusion, this study mines the learning characteristics and learning requirements of most users in the group, gives the importance of nodes in the GLGN on different learning needs, and builds a multi-constraint model for vertex importance. This paper uses the GLGN to recommend learning subgraphs, which breaks through the monotony of the linear ordering approach that appears in traditional recommendation systems.

In summary, this paper focuses on the construction and similarity computation method based on time-sliced learning generative networks and proposes a group-oriented recommendation algorithm based on a graph similarity calculation. In the second chapter, we introduce the related research on group division and group recommendations. In the third chapter, we give the basic definitions of the user-knowledge-point heterogeneous network (UKN) and two types of learning generative networks (LGN). The fourth chapter proposes the generation method of PLGN and gives the similarity calculation method between PLGNs while considering both the PLGNs' contents and structures. Then, the users are divided into groups according to the PLGN similarities. In the fifth chapter, based on the results of the group division, the construction method of GLGN and the group-oriented learning subgraph recommendation algorithm based on the GLGN are given. Then, the experiment data, experimental design and evaluating indicator are introduced, and the experimental results are analyzed. The sixth chapter is the summary and prospects.

II. RELATED WORK

Group recommendation methods are usually divided into two categories: fusion based on individual recommendation models and fusion based on individual recommendation result[9,10,11]. The former method is to aggregate the user preferences into a group preference and then recommend a preference service to the group users. The more similar the user preferences are in the group, the better the group recommendations. The latter method obtains each user's personalized recommendation list and fuses all of the lists into the same list as the group recommendation result [12]. The following is summarized from the existing group division and group recommendation research.

A. GROUP DIVISION

A group division method is the most important component for group recommendations[13]. Wang[9] proposed a hidden factor graph model, which used a variety of implicit and explicit social recognition and text information on the user groups and trained a user group recognition model. Chen[14] made a quantitative analysis of user characteristics based on their interests, course information, knowledge level, and so on and then used genetic algorithms to generate the learning groups. L Boratto[15] evaluated group modeling strategies in a group recommendation scenario in which the groups were detected by clustering the users. Once the users were clustered, strategies were tested to find the strategy that obtained the best accuracy. Jin Tao[16] proposed to use locality sensitive hashing technology (LSH) to achieve groups rapidly. Tan[17] presented the typical user group (TUG) concept, which compares the contribution of new typical users to typical user groups and expands the scale of the typical user groups one by one, to ultimately achieve higher recommendation coverage and accuracy. Zhang C[18]

proposed a clustering algorithm for user group recommendations that uses the singular value decomposition algorithm to extract the characteristics of the main users and simplifies the calculation by reducing the dimension of the user feature vectors. Young-Duk Seo[19] proposed an upward leveling (UL) aggregation method that considers deviations for group recommendations. J Castro[20] proposed to keep all group member preferences by using hesitant fuzzy sets (HFSs) and retaining information such as the group hesitation about their preferences, which will be used in the group recommendation process.

It can be found from the above literature that user feature selection is of great importance in the group divisions, and it is usually required to combine the users' static and dynamic characteristics before establishing the best group model. In the field of e-learning, a learning group usually refers to users with similar learning interests, including the users' accesses to the same learning resources. Users' explicit scores on learning resources and the implicit attributes of access to resources, such as learning time and learning frequency, are often used together to calculate the similarities between users. However, most of the existing methods lack consideration of the users' cognitive processes. For example, two users who have learned the same two knowledge points but at different times can be considered dissimilar according to the forgetting curve proposed by H. Ebbinghaus. Thus, it is not sufficient to divide the group by only judging whether the users have access to the resources.

B. GROUP RECOMMENDATION

Group recommendation refers to the recommendation of information or projects that could meet the needs or interests of the group users. Roy[21] examined the problem of enabling the flexibility of updating one's preferences in group recommendations. According to the preference vector, state dynamic adjustment is made to recommend a suitable resource set for the group users. Zapata[22] used a collaborative search method to select, evaluate and recommend learning objects and recommended a series of learning objects to the learning group. Yuan[5] proposed a probabilistic model called COM (COnsensus Model) to model the generative process of group activities and make group recommendations. Wang Haiyan[12] proposed a latent group recommendation (LGR) based on a dynamic probabilistic matrix factorization model integrated with a convolutional neural network (DPMFM-CNN). He M[23] considered the interactions among group members and reconstructed the dynamic decision-making process within a group based on the inspiration of Pareto Improvement. Sacharidis[24] proposed two group recommendation models and attempted to solve the discrepancy between individual and group behavior evaluations. Villavicencio[25] presented a multi-agent approach for group recommendation called PUMAS-GR. This approach leverages on negotiation techniques to integrate the recommendations obtained for

each group member into a list for the group. Bok K[26] proposed a group recommendation scheme using users' profiles and collaborative filtering over a social network. Their approach uses the profiles from other users in the collaborative filtering group to recommend the user groups. Ke Ji [27] proposed a topic-based probabilistic model called GIST to infer group activities and make group recommendations. This approach jointly considers individual members' choices and subgroups' choices for group recommendations. Liu[28] investigated the group recommendation problem from a novel approach, which is to attempt to maximize the satisfaction of each group member while minimizing the unfairness between them. Wang Jing[29] proposed a group recommendation based on user topical influence analysis. Four user-factors are introduced to calculate the user social influence on the topical sub-groups. Wei Wang[30] presented a hierarchy visualization method for the group recommender (HVGR) systems to provide a visual presentation and intuitive explanation. L Boratto[13] analyzed the influence of predicting the ratings to accurately rate a group recommender system and used it to avoid data sparsity.

In summary, the research of the existing group recommendation mainly focused on group preference acquisition, group recommendation visualization, group recommendation utility evaluation, and so on. However, these recommendation results show that few studies have considered the sequences and dependencies of the recommended resources. For e-learning, the sequences and dependencies of the recommendation resources will directly affect the user learning. In recent years, research on recommender systems has shown that a knowledge map is beneficial for hybrid recommender systems to build learning dependencies. In this paper, the proposed group-oriented recommendation algorithm based on a learning generating network is characterized by providing a recommendation subgraph that is based on a knowledge map, which is more conducive to the construction of users' cognitive processes.

III. CORRELATION DEFINITION

A. User-knowledge-point Heterogeneous Network (UKN)

The user-knowledge-point heterogeneous network is a network topology map that is generated from user learning logs, user information, knowledge point information and knowledge maps. The vertices are composed of users and knowledge points, while the edges are composed of the relationships between the users, the relationships between the knowledge points and the learning relationships between the users and knowledge points. Its formal representation is shown in (1):

$$\begin{cases} UKN = G(V, E, T_1, T_2) \\ V = \{V_u, V_p\} \\ E = \{E_u, E_p, E_{up}\} \end{cases} \quad (1)$$

where UKN is the user-knowledge-point association network from time T_1 to time T_2 . Network vertices are composed of a user set V_u and knowledge point set V_p . Network edges are composed of the user relationships E_u , knowledge point relationships E_p and user learning knowledge point relationships E_{up} . The user-knowledge-point association network indicates the correlations between the users and knowledge points. The user-knowledge-point related network ignores the differences between types for the three types of relationships between the users and knowledge points, and it unifies them as edge weights instead.

B. LEARNING GENERATIVE NETWORK (LGN)

The Learning Generative Network is a network topology graph that is dynamically generated based on the knowledge point relationship of the user learning logs and user similarities. The learning generative network is a directed graph that is composed of learning resources viewed by all users in the learning logs as vertices and learning dependencies between resources as edges. Learning generative networks can be divided into personal learning generative networks (PLGN) and group learning generative networks (GLGN). Its formal representation is shown in (2):

$$\begin{cases} PLGN_{u_x} = G(V_{u_x}, E_{u_x}, T_1, T_2) \\ GLGN_{g_a} = G(V_{g_a}, E_{g_a}, T_1, T_2) \\ V_{u_x} \subseteq V, V_{g_a} \subseteq V \\ E_{u_x} \subset V \times V, E_{g_a} \subset V \times V \end{cases} \quad (2)$$

where $PLGN_{u_x}$ refers to the learning generative network of user u_x from time T_1 to time T_2 . V_{u_x} refers to the resource set that user u_x ever viewed from time T_1 to time T_2 . Each resource represents a vertex in the learning generative network and has a timestamp. E_{u_x} is the set of relationships between the resources that user u_x learned from time T_1 to time T_2 . The group learning generative network $GLGN_{g_a}$ has a similar structure to the $PLGN_{u_x}$ where g_a refers to a group of users.

C. Knowledge Map (KM)

The knowledge map of a course is composed of all knowledge points and learning orders between the knowledge points. The formal representation of the knowledge maps is shown in Eq. (3):

$$\begin{cases} KM = (V_{kg}, E_{kg}) \\ V_{kg} = \{v | v \in KG\} = V \\ E_{kg} = \{(v_i, v_j) | v_i \in KG, v_j \in KG, (v_i, v_j) \in KG\} \subset V \times V \end{cases} \quad (3)$$

where the vertex set V_{kg} equals the knowledge point set V . The edge set E_{kg} is composed of the learning dependencies between the knowledge points, which given by the experts in the related field.

IV. CONSTRUCTION OF THE PLGN AND ITS CHARACTERISTIC ANALYSIS

The group-oriented learning subgraph recommendation algorithm based on GLGN includes three steps: the construction of the PLGN, the grouping of the PLGN, the construction of the GLGN and the group recommendation. The overall process is shown in Fig. 3.

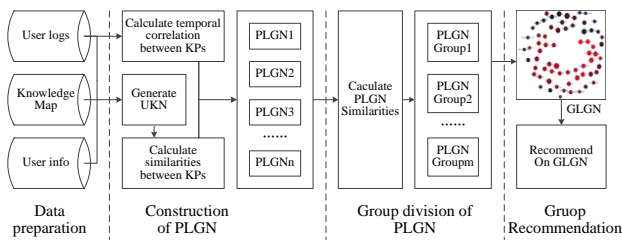


FIGURE 3. RESEARCH FRAMEWORK

The construction of the PLGN requires the user information, knowledge point information and user logs. Within a time slice (T_1, T_2) a user's knowledge-point learning sequence, such as ' $S_1-S_2-S_3-S_4-S_5$ ', is extracted from his/her user log to build the learning generative network. In this study, a learning generative network's generative algorithm is based on the knowledge points' similarities, and their sequence relationships are used to transform the learning sequences of the knowledge points that were extracted from the user logs into PLGNs. To determine the number of new vertices and edges that must be generated, the 'matching degree' mat_k^i is proposed to quantify the matching degree of the $(i+1)th$ vertex and the existing k th vertex. The matching degree depends on the similarity of the knowledge points and the temporal correlation of the knowledge points that the user learned. To divide the PLGNs into groups, a graph similarity measurement method based on the content and structure is proposed.

A. SIMILARITY OF KNOWLEDGE POINTS

To calculate the similarities between the knowledge points, this study first constructs the relationships between the users based on whether they are in the same profession, in the same batch, or have the same level of education, and it constructs the relationships between the knowledge points based on whether they have a sequential relationship and

whether they belong to the same chapter. It constructs the user-knowledge-point relationships based on the cumulative learning length and number of times. Then, the three types of relationships are quantified and normalized into edge weights to form the user-knowledge-point heterogeneous network. Finally, a random walk method proposed by Fouss et al.[31] is used to calculate the similarities dis_{ij} between any two knowledge points i and j in the heterogeneous network.

B. TEMPORAL CORRELATION OF KNOWLEDGE POINTS

The temporal correlation of knowledge points indicates the learning sequential correlation and the learning time correlation between any two knowledge points that the user learns. It is necessary to give a certain temporal threshold when calculating the temporal correlations of the knowledge points. Researchers at the University of Missouri-Columbia found that the brain can remember up to 3-4 things in a short period of time[32]. At the same time, according to the forgetting curve proposed by Ebbinghaus, the learning dependence of a certain vertex in a PLGN only generates from the vertex he/she learned in a week, which is in line with people's cognitive laws. Therefore, this study calculates only the temporal correlations between one vertex and at most the previous 4 vertices that a user learned in a week. In other words, this paper takes the sequential threshold $\gamma = 4$ and the time threshold $\delta = 7 * 24h$.

When there are already i vertices in the PLGN, the newly added vertex is the $(i+1)th$. The formalization of the temporal correlation is expressed as

$$\begin{cases} tpl_{i+1,k}^{time} = \frac{2 * tpl_{i+1,k}^{seq} * tpl_{i+1,k}^{time}}{tpl_{i+1,k}^{seq} + tpl_{i+1,k}^{time}} \\ tpl_{i+1,k}^{seq} = \frac{i+1-k}{\gamma} \\ tpl_{i+1,k}^{time} = \frac{T_{i+1}-T_k}{\delta} \end{cases} \quad (4)$$

where γ and δ are normalized coefficients. The temporal correlation $tpl_{i+1,k}$ between the $(i+1)th$ vertex and the k th vertex is determined by the harmonic mean of the order correlation $tpl_{i+1,k}^{seq}$ and the time correlation $tpl_{i+1,k}^{time}$. $T_{i+1}-T_k$ represents the time distance between user the learning knowledge point $i+1$ and the knowledge point k .

C. CONSTRUCTION OF THE PLGN

As the similarities between knowledge points and the temporal correlation of user learning knowledge points are obtained, the matching degree between two knowledge points from the user's learning sequence can be calculated to generate the learning dependencies and build the PLGN. The formula of the matching degree between the $(i+1)th$ knowledge point and the k th knowledge point is

$$mat_{i+1,k} = \alpha * dis_{i+1,k} + \beta * tpl_{i+1,k} \quad (5)$$

where α and β are the weight coefficients. In this study, $\alpha = \beta = 0.5$.

Using the matching degree, $mat_{i+1,k}$ allows us to synthetically consider the similarities between the knowledge points and the temporal correlations of the user learning knowledge points. The smaller the value of $mat_{i+1,k}$ is, the more reasonable the addition of the edges $(k, i+1)$ appears to be. Thus, it is required to calculate and sort all of the $mat_{i+1,k}$ values to determine which edge or edges should be added to the PLGN when the $(i+1)$ vertex is added to it. Table 1 shows the algorithm for generating the PLGN.

TABLE I
THE ALGORITHM FOR GENERATING THE PLGN

Input:	user logs Log ; id of u_x ; Time interval (T_1, T_2) ; Temporal threshold Δt ;
Output:	Personal learning generative network.
1.	Form the learning sequence $List(id, Log, T_1, T_2) = \{s_i\}$ of user u_x from user logs Log within time interval (T_1, T_2) accord to id;
2.	Add the first knowledge point to the $PLGN_{u_x}$;
3.	Starting from the second knowledge point, do the following:
4.	For the certain knowledge point s_i , find its learning dependence set $Candidate(s_i)$ according to the temporal threshold Δt ;
5.	For each $s_k \in Candidate(s_i)$, calculate the matching degree $mat_{i,k}$;
6.	Find $mat_{i,m} = \min(mat_{i,k})$, add the learning dependence from s_m to s_i ;
7.	For each $s_j \in Candidate(s_i)$ that satisfies $mat_{i,j} * 80\% < mat_{i,m}$, add the learning dependence from s_j to s_i

D. GROUP DIVISION BASED ON THE SIMILARITIES OF THE PLGNS

The group division of the PLGN is achieved by measuring the similarities of the PLGNS. The composition of the graph similarity indicators used in this study is as follows:

Similarity of PLGNS	Content similarity
	Structural similarity
	Vertex number
	Edge number
	Average vertex degree
	Average vertex strength
	Round number
	Average round size

The content similarity of $PLGN_{u_x}$ and $PLGN_{u_y}$ can be calculated using the graph kernel theory[8]. The principle is to calculate the proportion of the same paths between any

pairs of vertices in two PLGNS and to take the average to obtain the content similarity value between the two PLGNS. The normalized value $PLGNCntDis_{u_x, u_y}$ is represented as the following formula:

$$PLGNCntDis_{u_x, u_y} = \frac{1}{N(N-1)} \sum_{i=0}^{i < N, i \neq j} \sum_{j=0}^{j < N} ppDis_{u_x, u_y}^{i, j} \quad (6)$$

$$ppDis_{u_x, u_y}^{i, j} = \begin{cases} \frac{(sc_{u_x, u_y}^{i, j})^2}{tc_{u_x}^{i, j} * tc_{u_y}^{i, j}}, & \text{if } tc_{u_x}^{i, j} > 0 \text{ and } tc_{u_y}^{i, j} > 0 \\ 0.5, & \text{if } tc_{u_x}^{i, j} = 0 \text{ and } tc_{u_y}^{i, j} = 0 \\ 0, & \text{otherwise} \end{cases}$$

where N is the total number of knowledge points of the course to which the PLGN belongs, and i and j represent the index of the knowledge points in the course. Here, $ppDis_{u_x, u_y}^{i, j}$ is the similarity between the knowledge points i and j of $PLGN_{u_x}$ and $PLGN_{u_y}$. Additionally, $tc_{u_x}^{i, j}$ means the number of paths that exist between knowledge points i and j of $PLGN_{u_x}$ and $PLGN_{u_y}$, and $sc_{u_x, u_y}^{i, j}$ means the number of paths that exist in both $PLGN_{u_x}$ and $PLGN_{u_y}$.

The structural similarities of the PLGNS are reflected in the vertex number, edge number, average vertex degree, average vertex strength, round number and average round size of the PLGNS. The value $PLGNCntDis_{u_x, u_y}$ can be calculated by the weighted sum of the normalized differences of the six indexes as the following formula:

$$PLGNStrDis_{u_x, u_y} = \sum_{i=1}^6 dis_{u_x, u_y}^{k_i} * imp^{k_i} \quad (7)$$

$$dis_{u_x, u_y}^{k_i} = \frac{norm(value_{u_x}^{k_i} - value_{u_y}^{k_i})}{Max(\{|value\})}$$

$$k_i \in \{ver, edg, deg, str, rnd, rSize\}$$

where $dis_{u_x, u_y}^{k_i}$ is the similarity value between $PLGN_{u_x}$ and $PLGN_{u_y}$ on attribute k_i . Here imp^{k_i} means the weight of attribute k_i and k_i means one of the six aspects that measure the similarity. dis_{u_x, u_y}^{ver} is the vertex number distance that is calculated by normalizing the difference between the vertex numbers of $PLGN_{u_x}$ and $PLGN_{u_y}$. Similarly, dis_{u_x, u_y}^{edg} is the edge number distance, dis_{u_x, u_y}^{dge} is the average vertex degree distance,

dis_{u_x, u_y}^{str} is the average vertex strength distance, dis_{u_x, u_y}^{rnd} is the round number distance, and dis_{u_x, u_y}^{rSize} is the average round size distance. Additionally, $norm(value)$ is the normalizing function. In this study, $imp^{ver} = imp^{deg} = 0.1$, $imp^{deg} = imp^{str} = imp^{rnd} = imp^{rSize} = 0.2$.

Finally, a weighted method is used to derive the overall similarities $PLGNDis_{u_x, u_y}$ between $PLGN_{u_x}$ and $PLGN_{u_y}$.

$$PLGNDis_{u_x, u_y} = \alpha * PLGNCntDis_{u_x, u_y} + \beta * PLGNStrDis_{u_x, u_y} \quad (8)$$

where α and β are the weights. In this study, $\alpha = 0.7$ and $\beta = 0.3$.

After completing the PLGN similarity calculation, this study uses the Cure hierarchical clustering algorithm to divide the PLGNs into groups.

V. GROUP-ORIENTED RECOMMENDATION BASED ON THE GLGN

The group recommendation system fuses the preference of each group member to obtain the group preference and generates the group recommendation according to the group preference. In this study, a group learning generative network (GLGN) that embodies the learning trajectory of group network learning behaviors is generated based on group division, to realize resource recommendations for a variety of purposes. To better retain the dependencies between the knowledge points, this paper presents the recommendations in the form of graphs.

A. GENERATION OF THE GLGN

An LGN corresponds to a matrix of edge weights between vertices. Therefore, a GLGN can be generated by the superposition of the PLGNs from one group. In other words, all matrices that the PLGNs correspond with are calculated first; then, all of the matrices are added together to form a group matrix, and finally, the GLGN is restored from the group matrix. The formal representation is as follows:

$$\begin{cases} GLGN_{g_a} = getLGN(LW_u) \\ LW_u = \sum W_{u_x} \\ W_{u_x} = getW(PLGN_{u_x}) \\ PLGN_{u_x} \in Cluster_{g_a} \end{cases} \quad (9)$$

where $Cluster_{g_a}$ represents one group of PLGNs. W_{u_x} is the corresponding user matrix of $PLGN_{u_x}$. LW_u is the superposition of all of the user matrices that belong to $Cluster_{g_a}$.

B. GROUP-ORIENTED LEARNING SUBGRAPH RECOMMENDATION ALGORITHM BASED ON THE GLGN

The GLGN can reflect the knowledge points and learning dependencies that most users learned and used in the group. For a group of users, different vertices in the GLGN can have

different degrees of importance and could be suitable for users to learn in different scenarios. Therefore, this study referred to a multi-constraint learning path recommendation algorithm[33] and conducted the construction of multi-constraint models based on the importance of the vertices; then, it used the constructed models to design a group-oriented learning subgraph recommendation algorithm.

1) CONSTRUCTION OF A MULTI-CONSTRAINT MODEL FOR VERTEX IMPORTANCE

The group-oriented learning subgraph recommendation algorithm must focus on three types of weights of vertices, which are the extensional-learning weight, key-learning weight and leak-filling-learning weight. This study transforms the vertex into a three-dimensional vector. For the vertex v_i , the vector form is expressed as

$$\mathbf{w}_i = (w_1, w_2, w_3) \quad (10)$$

where w_1, w_2, w_3 respectively represent the importance of vertex v_j in extensional learning, key learning and leak-filling learning.

The group-oriented learning subgraph recommendation algorithm must focus on three types of vertices. The first vertex is the extensional-learning vertex, which is the vertex that the group users ignore but need to learn. If there is a learning dependency between vertex v_i and vertex v_j on the knowledge map, vertex v_j exists in $GLGN_{g_a}$ and vertex v_i does not exist in $GLGN_{g_a}$; then, vertex v_i is an extensional-learning vertex. The similarity of the knowledge points in IV A is used as its extensional-learning factor w_1 . The extended GLGN is denoted $GLGN'_{g_a}$. The second vertex is the key-learning vertex. This vertex computes the vertex strength (the sum of the weights of the connected vertex edges) for each vertex in $GLGN'_{g_a}$ as the key-learning factor w_2 . The last vertex is the leak-filling-learning vertex. If the total learning time length of a user on a knowledge point is less than 90% of its intrinsic time length, then it is considered that the knowledge point is not effectively learned, and leak-filling learning is required. If more than 50% of the users need leak-filling learning of a vertex, the strength of this vertex is calculated and normalized as the leak-filling-learning factor w_3 .

To be able to recommend knowledge points under different learning requirements, we use the constraint $\Phi = (\alpha, \beta, \gamma)$ to adjust the importance of vertex v_j . In other words, we have the following:

$$\mathbf{w}_i^\Phi = (\alpha * w_1, \beta * w_2, \gamma * w_3) \quad (11)$$

where α, β, γ are the given constants in the range of $[0,1]$. For those who want to focus on extensional learning, key learning or leak-filling learning, it is proper to use the learning factor $\Phi_1 = (1,0,0)$, $\Phi_2 = (0,1,0)$ or $\Phi_3 = (0,0,1)$. For those who want to perform key learning and leak-filling learning, $\Phi_4 = (0,0.5,0.5)$ can be used. The specific

importance value is the modulus of the importance vector, which is the following:

$$w_i^\Phi = \|\mathbf{w}_i^\Phi\| = \sqrt{(\alpha * w_1)^2 + (\beta * w_2)^2 + (\gamma * w_3)^2} \quad (12)$$

2) LEARNING SUBGRAPH RECOMMENDATION ALGORITHM

Under a certain learning requirement with the given constraint Φ , the importance of the vertices can be calculated according to the model of the vertex importance to recommend the learning subgraphs. When recommending learning subgraphs for users, the recommendation system should consider not only the importance of the vertex but also the learning dependencies of the vertices in the GLGN. This study separately calculates the importance of each vertex in the extended GLGN and marks the top 15% of the vertices with the largest importance on the GLGN. Then, all of the connected subgraphs are found from these top-15% vertices, and their average vertex importance can be calculated. The subgraph with the top average vertex importance is recommended to the group. The algorithm is shown in Table II.

TABLE II
THE LEARNING SUBGRAPH RECOMMENDATION ALGORITHM BASED ON THE
VERTEX IMPORTANCE MODEL

Input:	Group learning generative network $GLGN_{g_a}$; Personal learning generative network set $\{PLGN\}_{g_a}$; User logs $\{Log\}_{g_a}$; Knowledge map KM ; Constraint condition $\Phi = (\alpha, \beta, \gamma)$;
Output:	Recommended learning subgraph $G_{g_a}^{sub}$.
1.	Generate the expanded group learning generative network $GLGN'_{g_a}$;
2.	foreach vertex v_i in $GLGN'_{g_a}$ and not in $GLGN_{g_a}$:
3.	Calculate the extensional-learning factor w_1 of v_i ;
4.	end for
5.	foreach vertex v_i in $GLGN'_{g_a}$:
6.	Calculate the key-learning factor w_2 of v_i ;
7.	end for
8.	Use $\{Log\}_{g_a}$ and mark all of the knowledge points that have not been effectively learned on $GLGN_{g_a}$;
9.	foreach vertex v_i in marked vertex set:
10.	Calculate the leak-filling-learning factor w_3 of v_i ;
11.	end for
12.	foreach vertex v_i in $GLGN'_{g_a}$:
13.	Form its vector representation $\mathbf{w}_i = (w_1, w_2, w_3)$;
14.	Calculate the importance of vertex v_i according to Φ ;
15.	end for

16. Sort the vertices using their importance degree in descending order;
17. Mark the top 15% vertices;
18. Find all the connected subgraphs and record them in $\{G_{g_a}^{sub}\}$;
19. **foreach** learning graph G_j^{sub} **in** $\{G_{g_a}^{sub}\}$:
20. Calculate the average vertex importance of G_j^{sub} ;
21. **end for**
22. Search $\{G_{g_a}^{sub}\}$ for the one with the top average vertex importance $G_{g_a}^{sub}$;
23. **return** recommended learning subgraph $G_{g_a}^{sub}$.

C. Experimental results and analysis

Unlike traditional recommendation systems, the data sets required for group recommendation must include group information. However, there are few data sets that contain group information. The commonly used method is to construct groups from these data sets according to the group characteristics needed for the experiment[34][35][36]. Based on the existing e-learning logs and knowledge map, this study constructed PLGNs and GLGNs to obtain group division and recommendation results. For group-based recommendations, its accuracy rate, recall rate, normalized discounted cumulative gain (NDCG), and mean average precision (MAP) were calculated.

1) EXPERIMENTAL DATA SET

This research used the log data of the Java course from May 9, 2014 to November 30, 2017, including 11 batches of 247 learners. The knowledge map of the Java course contains 70 knowledge points and 79 edges. A total of 156 learners were selected to perform the group recommendation and verify the recommendation effect.

2) EXPERIMENTAL CONTRASTED ALGORITHM

This study used the above methods to recommend learning subgraph for groups. At the same time, a group-based K-nearest neighbor collaborative filtering recommendation algorithm was synchronously used for contrast. The recommendations made in this study were group-specific recommendations, and experimental schemes were designed to verify the pros and cons of their recommendations.

The key and leak-filling learning knowledge points recommendation results are used as the recommendation result for a group and compared the results with the recommendation results of the group-based collaborative filtering algorithm. The formal representation of the group-based collaborative filtering algorithm is as follows:

$$\left\{ \begin{aligned} rat(g_a, p_i) &= \frac{\sum_{u_x \in g_a} w(u_x, g_a) * rat(u_x, p_i)}{\sum_{u_x \in g_a} w(u_x, g_a)} \\ pred(g_a, p_i) &= \frac{\sum_{g_b \in |g|} sim(g_a, g_b) * rat(g_b, p_i)}{\sum_{g_b \in |g|} sim(g_a, g_b)} \\ w(u_x, g_a) &= \sum_{u_y \in g_a} PLGNDis_{u_x, u_y} \\ sim(g_a, g_b) &= \sum_{u_x \in g_a} \sum_{u_y \in g_b} PLGNDis_{u_x, u_y} \end{aligned} \right. \quad (13)$$

where $rat(g_a, p_i)$ is the score of group g_a on knowledge point p_i . Here, $rat(u_x, p_i)$ is the score of user u_x on knowledge point p_i . $w(u_x, g_a)$ is the weight of user u_x in group g_a , the value of which is the summation of the similarities between u_x and all of the other users in group g_a . $sim(g_a, g_b)$ represents the similarity between group g_a and group g_b , the value of which is the summation of all of the similarities between the users from group g_a and the users from group g_b . $PLGNDis_{u_x, u_y}$ is the similarity between user u_x and user u_y .

3) EVALUATION INDICATOR

This study provides the corresponding evaluation indexes from four aspects, namely, the accuracy rate, recall rate, normalized discounted cumulative gain (NDCG), and mean average precision (MAP). The final evaluation results take the average of the evaluation results for all of the users.

a. ACCURACY RATE AND RECALL RATE

The formal description of the accuracy rate and recall rate based on the list-form recommendation for a single user is as follows:

$$\left\{ \begin{aligned} P_p^{u_x} &= P(pred(u_x), real(u_x)) = \frac{|pred(u_x) \cap real(u_x)|}{|pred(u_x)|} \\ R_p^{u_x} &= R(pred(u_x), real(u_x)) = \frac{|pred(u_x) \cap real(u_x)|}{|real(u_x)|} \end{aligned} \right. \quad (14)$$

where $p = pred(u_x)$ represents the set of knowledge points recommended to user u_x , and $real(u_x)$ represents the set of knowledge points that were actually learned by user u_x .

The accuracy rate and the recall rate of the collaborative filtering algorithm recommendation results can be calculated using the above method. Since the recommendation form of the algorithm proposed in this study is the learning subgraph, this study proposed a method for calculating the accuracy rate and recall rate of the learning subgraph.

This approach requires the recommendation sets and test sets to calculate the accuracy and recall rate. For the collaborative filtering algorithm, both are lists of knowledge points, and as a result, there is no problem in the computation. For the group-oriented recommendation algorithm proposed in this paper, the recommendation sets contain not only knowledge points but also learning dependencies. Thus, the corresponding test sets must be converted into the same structure. For example, if the sequence of knowledge points

actually learned by the user is ' S_1 - S_2 - S_3 - S_4 - S_5 ', then the converted test set of he/she should be $real(u_x) = \{S_1, S_2, S_3, S_4, S_5, (S_1, S_2), (S_2, S_3), (S_3, S_4), (S_4, S_5)\}$. Therefore, for the group-oriented learning subgraph recommendation algorithm proposed in this study, its accuracy rate and recall rate can be calculated from the collective accuracy rate and recall rate of the recommended knowledge points and learning dependencies. At the same time, the accuracy rate and recall rate of a traditional collaborative filtering recommendation algorithm can be calculated using the original method. Therefore, this study has unified the accuracy rate and recall rate for the recommended graphs and lists.

In addition, for a single group g_a recommendation, the accuracy rate and recall rate can be reflected in those of the recommendation results on a single user within the group, which is formally expressed as

$$\left\{ \begin{aligned} P_p^{g_a} &= \frac{\sum_{u_x \in g_a} w(u_x) * P_p^{u_x}}{\sum_{u_x \in g_a} w(u_x)} \\ R_p^{g_a} &= \frac{\sum_{u_x \in g_a} w(u_x) * R_p^{u_x}}{\sum_{u_x \in g_a} w(u_x)} \\ w(u_x, g_a) &= \sum_{u_y \in g_a} PLGNDis_{u_x, u_y} \end{aligned} \right. \quad (15)$$

where $w(u_x)$ indicates the importance of user u_x in group g_a . If a user has a large degree of similarity with the majority of users in the group, the user will have a large influence on the group recommendation. Therefore, this study uses the average of the similarity $PLGNDis_{u_x, u_y}$ between the user u_x and other users in the group as the importance of user u_x .

b. MEAN AVERAGE PRECISION (MAP)

The mean average precision of a recommended knowledge point list is the average of the accuracy rate of each recommended knowledge point. For a recommended knowledge point list $p = \{p_1, p_2, p_3, \dots\}$ and for a list that actually needs to be recommended $p^{true} = \{p_1^{true}, p_2^{true}, p_3^{true}, \dots\}$ (which has size N), if there exist n knowledge points that are in both list p and list p^{true} and if its sequence number in list p is $rank_p = \{rank_1, rank_2, \dots, rank_n\}$, the mean average precision of the recommended knowledge point list is as follows:

$$MAP_p = \frac{1}{N} * \sum_{i=1}^n \frac{i}{rank_i} \quad (16)$$

where the range of values for MAP_p is $[0, 1]$.

c. NORMALIZED DISCOUNTED CUMULATIVE GAIN (NDCG)

The normalized discounted cumulative gain can be used to evaluate recommendation results in the list forms, and the higher the ranking of items in the list, the greater the impact on the results of the evaluation. For a recommended knowledge point list $p = \{p_1, p_2, p_3, \dots\}$, if the score list is $rel_p = \{real_{p_1}, real_{p_2}, real_{p_3}, \dots\}$, the discounted cumulative gain DCG is as follows:

$$DCG_p = \sum_i \frac{2^{rel_{p_i}} - 1}{\log_2^{(i+1)}} \quad (17)$$

If the score list is sorted in descending order, in other words, for a recommended knowledge point list $p' = \{p'_1, p'_2, p'_3, \dots\}$, if the score list is

$$rel_{p'} = \{real_{p'_1}, real_{p'_2}, real_{p'_3}, \dots\} \quad \text{and} \quad \text{satisfies}$$

$real_{p'_i} \geq real_{p'_j}, \forall i < j$, then the DCG value is the ideal discounted cumulative gain:

$$IDCG_p = \sum_i \frac{2^{rel_{p'_i}} - 1}{\log_2^{(i+1)}} \quad (18)$$

Then, by normalizing the discounted cumulative gain DCG with the ideal discounted cumulative gain INCG, the normalized discounted cumulative gain is calculated as follows:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \quad (19)$$

where the range of values for $NDCG_p$ is $[0,1]$.

3) EVALUATION RESULTS

In the contrasted group-based collaborative filtering algorithm, the number of groups used is 5, and the number of neighbors used in K-nearest neighbor collaborative filtering is 10. Under the different time slices, the group learning generative network generated by this study is different; thus, the recommended number of knowledge points is also different. In this regard, the number of knowledge points recommended by the collaborative filtering algorithm must be kept consistent with this research method.

The experiment used the user logs on the Java course after September 1, 2015. It started at the time slice of 2 weeks and took an increment of 2 weeks at each time. Finally, the GLGN was generated and used on recommendations 9 times. The GLGN recommendation result is learning subgraphs whose sizes are not fixed. Therefore, for each learning group, it is necessary to ensure that the recommended list size of the CF recommendation algorithm stays in step with that of the GLGN recommendation algorithm. Table III and Table IV compare the effectiveness of the two recommendation algorithms:

TABLE III
COMPARISON I OF THE RESULTS OF TWO RECOMMENDATION ALGORITHMS

Data quantity/ Week	Basic		Precision rate		Recall rate	
	User Num- bers	Rec size	This study	Group CF	This study	Group CF
2	13	4	0.058	0.173	0.058	0.112
4	35	8	0.050	0.166	0.041	0.242
6	59	9	0.049	0.158	0.045	0.236
8	79	9	0.119	0.136	0.250	0.232
10	94	9	0.143	0.133	0.347	0.239
12	103	9	0.148	0.130	0.384	0.249
14	130	9	0.182	0.133	0.420	0.251
16	145	10	0.166	0.138	0.427	0.333
18	150	10	0.186	0.140	0.483	0.344

TABLE IV
COMPARISON II OF THE RESULTS OF TWO RECOMMENDATION ALGORITHMS

Data quantity/ Week	Basic		MAP		NDCG	
	User Num- bers	Rec size	This study	Group CF	This study	Group CF
2	13	4	0.058	0.160	0.154	0.346
4	35	8	0.040	0.110	0.094	0.417
6	59	9	0.033	0.114	0.087	0.397
8	79	9	0.069	0.085	0.364	0.360
10	94	9	0.090	0.079	0.478	0.382
12	103	9	0.094	0.076	0.517	0.385
14	130	9	0.081	0.074	0.461	0.378
16	145	10	0.071	0.070	0.459	0.386
18	150	10	0.077	0.070	0.506	0.387

It is not difficult to determine from the data that when the data quantity is not large, the scale of the GLGN is not either, and the size of the recommended list is unable to reach the expected number (which is approximately 10). However, since the 5th experiment (data of 10 weeks), the amount of data is relatively sufficient, and thus, the GLGN recommendation effect is relatively stable. Its precision rate and recall rate are better than the group-based CF recommendation algorithm. At the same time, its MAP and NDCG are also better, which indicates that the recommended learning subgraphs are reasonable.

VI. CONCLUSION

Aiming at group recommendation in online learning, this paper proposed combining user learning logs and curriculum knowledge maps to generate a personal learning generation network to represent the user learning process. Then, a learning group division method based on the similarities of the PLGNs is given to generate the GLGN. On this basis, a group-oriented learning subgraph recommendation algorithm is designed and implemented. To measure the effect of the recommendations, four indexes for learning subgraph recommendations are used, which are the accuracy, recall rate, NDCG and MAP. Compared with the traditional group-based recommendation algorithm based on collaborative filtering, the proposed group-oriented recommendation algorithm can better adapt to large-scale network learning

environment such as MOOC when the group size is larger. Considering that there are some differences between the users' learning behavior characteristics for different courses and majors, the next group recommendation can be applied to performance comparisons and algorithm optimization in different majors and courses.

REFERENCES

- [1] Y. J. Zhang, Y. L. Du and X. W. Meng, "Research on group recommender systems and their applications," *Chinese Journal of Computers*, 2016, vol. 39, no. 4, pp. 745-764.
- [2] P. De Meo, F. Messina, D. Rosaci, G. Sarné, "Combining trust and skills evaluation to form e-Learning classes in online social networks," *Information Science*, 2017, vol. 405, pp. 107-122.
- [3] Recalde, Lorena. "A social framework for set recommendation in group recommender systems," European Conference on Information Retrieval. Springer, Cham, 2017, pp. 777.
- [4] M. K. Najafabadi, M. N. Mahrin and S. Chuprat, "Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data," *Computers in Human Behavior*, 2017, vol. 67, no. C, pp. 113-128.
- [5] Q. Yuan, G. Conga and C. Y. Lin, "COM: a generative model for group recommendation," in *ACM-SIGKDD*, 2014, pp. 163-172.
- [6] L. D. Camiel, J. D. Goldman-Levine and M. D. Kostka-Rokosz, "Twitter as a medium for pharmacy students' personal learning network development," *Currents in Pharmacy Teaching & Learning*, 2014, vol. 6, no. 4, pp. 463-470.
- [7] N. A. Cooke, "Professional development 2.0 for librarians: developing an online personal learning network (PLN)," *Library Hi Tech News*, 2012, vol. 29, no. 3, pp. 1-9.
- [8] M. A. Alvarez, X. Qi and C. Yan, "A shortest-path graph kernel for estimating gene product semantic similarity," *J Biomed Semantics*, 2011, vol. 2, no. 1, pp. 1-9.
- [9] Z. Q. Wang, S. S. Li and G. D. Zhou, "Personal group recommendation via textual and social information," *Ruan Jian Xue Bao/Journal of Software*, 2017, vol. 28, no. 9, pp. 2468-2480 (in Chinese). [Online] Available: <http://www.jos.org.cn/1000-9825/5267.htm>
- [10] Liu, Yujia, et al. "A trust induced recommendation mechanism for reaching consensus in group decision making," *Knowledge-Based Systems*, 2016, vol. 119, pp: 221-231.
- [11] Hong, Minsung, J. J. Jung, and D. Camacho. "GRSAT: A Novel Method on Group Recommendation by Social Affinity and Trustworthiness." *Journal of Cybernetics*, 2017, vol. 48, no. 3, pp: 140-161.
- [12] H. Y. Wang and M. W. Dong, "Latent group recommendation based on dynamic probabilistic matrix factorization model integrated with CNN," *Journal of computer research and development*. 2017, vol. 54, no. 8, pp. 1853-1863
- [13] L. Boratto, S. Carta and G. Fenu, "Influence of Rating Prediction on the Accuracy of a Group Recommender System that Detects Groups," *IEEE Intelligent Systems*, 2017, no. 99, pp. 1-1
- [14] L. Chen and Q. H. Yang, "A group division method based on collaborative learning elements," in *Chinese Control and Decision Conference*, 2014, pp. 1071-1075.
- [15] L. Boratt and S. Carta, "Modeling the Preferences of a Group of Users Detected by Clustering: a Group Recommendation Case-Study," in *ACM-WIMS*, 2014, pp. 1-7
- [16] T. Jin, J. Xie and Z. Yang, "Research on Fast Build Group and Group Recommendation in Social Networks," *Journal of Chinese Computer Systems*, 2017, vol. 38, no. 3, pp. 483-488.
- [17] C. Tan, Q. Liu and L. Wu. "Finding and Applying Typical User Group in Recommender Systems," *Pattern Recognition and Artificial Intelligence*, 2015, vol. 28, no. 5, pp. 462-471.
- [18] C. Zhang, J. Zhou, W. Xie, "A Users Clustering Algorithm for Group Recommendation," in *ACIT-CSII-BCD*, 2017, pp. 352-356.
- [19] Y. D. Se, Y. G. Kim and E. Lee, "An Enhanced Aggregation Method Considering Deviations for a Group Recommendation," *Expert Systems with Applications*, 2018, vol. 93, pp. 299-312.
- [20] J. Castro, M. J. Barranco and R. M. Rodríguez, "Dealing with diversity and novelty in group recommendations using Hesitant fuzzy sets," in *FUZZ-IEEE*, 2017, pp. 1-6.
- [21] S. B. Roy, S. Thirumuruganathan and S. Amer-Yahia, "Exploiting group recommendation functions for flexible preferences," in *IEEE-ICEBE*, 2015, pp. 412-423.
- [22] A. Zapata, V. H. Menéndez and M. E. Prieto, "Evaluation and selection of group recommendation strategies for collaborative searching of learning objects," *International Journal of Human-Computer Studies*, 2015, vol. 76, no. C, pp. 22-39.
- [23] M. He, W. Gu and Y. Kong. "Group recommendation: by mining users' check-in behaviors," in *ACM-UBICOMP and ACM-ISWC*, 2017, pp. 65-68.
- [24] D. Sacharidis, "Group Recommendations by Learning Rating Behavior," in *ACM-UMAP*, 2017, pp. 174-182.
- [25] C. P. Villavicencio, S. Schiaffino and A. D. Pace, "A Group Recommendation System for Movies based on MAS," *Advances in Distributed Computing & Artificial Intelligence Journal*, 2016, vol. 5, no. 3, pp. 1.
- [26] K. Bok, H. Yang and J. Yoo, "Social group recommendation based on dynamic profiles and collaborative filtering," *Neurocomputing*, 2016, vol. 209, no. C, pp. 3-13.
- [27] K. Ji, Z. Chen and R. Sun, "GIST: A Generative Model with Individual and Subgroup-based Topics For Group Recommendation," *Expert Systems with Applications*, 2018, vol. 94, pp. 81-93.
- [28] L. Xiao, Z. Min and Z. Yongfeng, "Fairness-Aware Group Recommendation with Pareto-Efficiency," in *ACM-RecSys*, 2017, pp. 107-115.
- [29] W. Jing, Z. Hui and Z. Liu, "Exploring User Influence for Topical Group Recommendation," *Chinese Journal of Electronics*, 2017, vol. 26, no. 1, pp.106-111.
- [30] W. Wang, G. Zhang and J. Lu, "Hierarchy Visualization for Group Recommender Systems," *IEEE Transactions on Systems Man & Cybernetics Systems*, 2017, no. 99, pp. 1-12.
- [31] F. Fouss, A. Pirotte and J. M. Renders, "Random-Walk Computation of Similarities between Learning Generative Networks of a Graph with Application to Collaborative Recommendation," *IEEE Transactions on Knowledge & Data Engineering*, 2007, vol. 19, no. 3, pp. 355-369.
- [32] http://www.sohu.com/a/144194865_559441
- [33] H. P. Zhu, F. Tian and K. Wu, "A Multi-Constraint Learning Path Recommendation Algorithm Based on Knowledge Map," *Knowledge-Based Systems*, 2018, vol.143, pp. 102-114.
- [34] T. D. Pessemer, S. Dooms and L. Martens, "Comparison of group recommendation algorithms," *Multimedia Tools & Applications*, 2014, vol. 72, no. 3, pp. 2497-2541.
- [35] M. Kompan and M. Bielikova, "Group Recommendations: Survey and Perspectives," *Computing & Informatics*, 2014, vol. 33, no. 2, pp. 1-31.

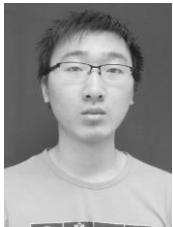
- [36] Felemban, Samah, M. Gardner, and V. Callaghan. "Towards Recognising Learning Evidence in Collaborative Virtual Environments: A Mixed Agents Approach." *Computers*, 2017, vol. 6, no. 3, pp.1-11



HAIPING ZHU received the B.S. degree in electronic science from Northwest University in 1996, the M.S. degree in information and communication engineer from Chinese Academy of Sciences in 1999, and the Ph.D. degree in computer science from Xi'an Jiaotong University, China.

She is currently a Lecturer and graduate supervisor with the Department of Computer Science and Technology, Xi'an Jiaotong University. She is a member of National Engineering Lab of Big Data Analytics, the innovation team of the Ministry of Education and the Satellite-Terrestrial Network Technology R&D Key Laboratory, Shaanxi Province. Her research interests include educational data mining, learning analytics and personalized recommendation.

Dr. Zhu won 1st prize of Science and Technology Progress Award of ShaanXi institution of higher learning, China. She is a member of the IEEE.



YIFU NI received the B.S. degree in computer science and technology from Xi'an Jiaotong University, China, in 2016.

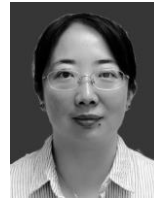
He is currently pursuing the M.S. degree in computer science and technology with Xi'an Jiaotong University, China. His current research interests include learning analytics and personalized recommendation.



FENG TIAN was born in Xi'an, ShaanXi, China in 1972. He received the B.S. degree in industrial automation and the M.S. degree in computer science and technology from the Xi'an University of Architecture and Technology, Xi'an, China, in 1995 and 2000, respectively, and the Ph.D. degree in control theory and application from Xi'an Jiaotong University, Xi'an, in 2003.

He has been with Xi'an Jiaotong University since 2004, where he is currently with National Engineering Lab of Big Data Analytics and also with the Systems Engineering Institute, as a Professor. He is a member of the Satellite-Terrestrial Network Technology R&D Key Laboratory, Shaanxi Province. His research interests include big data analytics, learning analytics, system modelling and analysis, and cloud computing.

Dr. Tian won 2nd class prize of National Science and Technology Progress Award of China (4th position, 2017), 1st class Price of MOE Science and Technology Progress award of China (4th position, 2015), 1st class prize of Science and Technology Progress award of Chinese Electrical Society (5th position, 2013), and TOP Grade Prize of Teaching Achievement Award of ShaanXi Province award, China (2nd Position, 2017). He is a member of the IEEE.



YAN CHEN received the B.S. degree in computer science and technology from Xi'an Jiaotong University, in 1993, the M.S. degree in computer science and technology from Xi'an Jiaotong University, in 1996, and the Ph.D. degree in computer science from Xi'an Jiaotong University, China in 2005.

She is currently an Associate Professor and graduate supervisor with the Department of Computer Science and Technology, Xi'an Jiaotong University. She is a member of National Engineering Lab of Big Data Analytics, the innovation team of the Ministry of Education and the Satellite-Terrestrial Network Technology R&D Key Laboratory, Shaanxi Province. Her research interests include educational data mining and learning analytics.



QINGHUA ZHENG received the B.S. degree in computer software, the M.S. degree in computer organization and architecture, and the Ph.D. degree in system engineering from Xi'an Jiaotong University, China, in 1990, 1993, and 1997, respectively.

He did post-doctoral research at Harvard University in 2002 and was a Visiting Professor of Research with the Hong Kong University from 2004 to 2005. He is currently a Professor with the Department of Computer Science and Technology, Xi'an Jiaotong University, where he serves as the Vice President. His research interests include intelligent e-learning theory and algorithm, computer network, and trusted software.

Dr. Zheng received the First Prize for National Teaching Achievement, State Education Ministry in 2005 and the First Prize for Scientific and Technological Development of Shanghai City and Shaanxi Province, in 2004 and 2003, respectively.



PEI FENG received the B.S. degree in Mathematics and applied mathematics from HeFei University of Technology, China, in 2017.

She is currently pursuing the M.S. degree in applied statistics with Xi'an Jiaotong University, China. Her current research interests include machine learning and learning analytics.