Knowledge-Based Systems 143 (2018) 102–114

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A multi-constraint learning path recommendation algorithm based on knowledge map

Haiping Zhu[a,b], Feng Tian[b,*], Ke Wu[a,c], Nazaraf Shah[d], Yan Chen[a,b], Yifu Ni[a,b], Xinhui Zhang[a], Kuo-Ming Chao[d], Qinghua Zheng[a,b]

[a] Department of Computer Science and Technology, Xi'an Jiaotong University, China
[b] National Engineering Lab of Big Data Analytics, Xi'an Jiaotong University 710049, China
[c] Shaanxi Province Key Lab. of Satellite and Terrestrial Network Tech. R&D, Xi'an Jiaotong University, Xi'an 710049, China
[d] Department of Computing, Coventry University CV12JH, UK

ARTICLE INFO

Article history:
Received 17 November 2016
Revised 15 October 2017
Accepted 8 December 2017
Available online 21 December 2017

Keywords:
E-learning
Knowledge map
Learning scenario
Learning path recommendation

ABSTRACT

It is difficult for e-learners to make decisions on how to learn when they are facing with a large amount of learning resources, especially when they have to balance available limited learning time and multiple learning objectives in various learning scenarios. This research presented in this paper addresses this challenge by proposing a new multi-constraint learning path recommendation algorithm based on knowledge map. The main contributions of the paper are as follows. Firstly, two hypotheses on e-learners' different learning path preferences for four different learning scenarios (initial learning, usual review, pre-exam learning and pre-exam review) are verified through questionnaire-based statistical analysis. Secondly, according to learning behavior characteristics of four types of the learning scenarios, a multi-constraint learning path recommendation model is proposed, in which the variables and their weighted coefficients considers different learning path preferences of the learners in different learning scenarios as well as learning resource organization and fragmented time. Thirdly, based on the proposed model and knowledge map, the design and implementation of a multi-constraint learning path recommendation algorithm is described. Finally, it is shown that the questionnaire results from over 110 e-learners verify the effectiveness of the proposed algorithm and show the similarity between the learners' self-organized learning paths and the recommended learning paths.

1. Introduction

The advances in learning technologies, rapid changing society and environment put the traditional learning methods or approaches into test, as learners want more customized, concise and selective materials with different scenarios in their learning process to maximize the learning outcomes.

Current e-learners often learn new knowledge with fragmented time through their mobile devices rather than a continuous period. Mobile learning (M-learning), with different learning scenarios, is characterized both by fragmented learning time and fragmented learning resources. For example, the analysis of 12,600 students' learning log of 40 courses from the Distance Learning College (DLC) of Xi'an Jiaotong University shows that more than 80% of continuous learning durations of the students are less than 3 min. At the same time, the utilization of fragmented time is highly related to learning scenarios and learner's learning behaviors. For example, at the beginning of a term, the learners want to learn the contents thoroughly and wish to spend more time on them, while before each examination, they wish to review more knowledge with less time. Therefore, the organization of learning resources directly affects the way of recommending the learning path, as learning time fragments require that a recommended learning path should be composed of many fine-grained resources in different scenarios. This brings a new dimension to the problem of learning path recommendation.

Selecting appropriate learning resources to compose a suitable learning path for e-learners is a complex task [24,31], and is an active research topic in personalized recommendation [11,12,17,30]. Most of the research on learning resource recommendation methods have ignored the inherent dependency between human cognitive characteristics and knowledge units. Although a few of them have considered this dependency but they have ignored other key factors such as different learning scenarios, fine-grained learning resources and limited time. Moreover, existing studies

* Corresponding author.
E-mail address: fengtian@mail.xjtu.edu.cn (F. Tian).

https://doi.org/10.1016/j.knosys.2017.12.011

on scenario based-learning (SBL) [7,18] and goal based scenario (GBS) [16] have shown influence on the e-learners. However, this research strand lacks the consideration of the learner's different learning path preferences in different learning scenarios.

Aiming to accommodate the different learning path preference of e-learners in different learning scenarios and fragmented learning time, this paper presents a new multi-constraint learning path recommendation algorithm to meet the diverse needs of learners in different learning scenarios. The main contributions of this paper are as follow.

(1) Based on the collected questionnaires, we have obtained learners' self-organized learning paths in four basic learning scenarios (initial learning, usual review, pre-exam learning and pre-exam review) and their ratings on the recommended learning paths. The analysis of the results based on these collected questionnaires proves following two hypotheses:
   - Hypothesis I: A learner has different learning path requirements in different learning scenarios.
   - Hypothesis II: Different learners have similar learning path requirements in a same learning scenario.

   E-learners' different learning path preferences for four different learning scenarios are verified via the proof of the two hypotheses.
(2) A multi-constraint learning path recommendation model based on a linear weighted formula is proposed to meet the learners' different learning path preferences in different learning scenarios and fragmented learning time with eight kinds of learning paths (complemented learning path, shortest learning path, shortest duration learning path, critical learning path, easy learning path, complete learning path, more-hotspot learning path and quick learning path) and their constraint factors for the proposed model. This model solves the problem of how to recommend a suitable path for an e-learner when he/she is learning in a specific scenario.
(3) We propose a multi-constraint learning path recommendation algorithm based on the proposed model and knowledge map [15], as well as other key information including the features of learners' behaviors, such as learning duration, learning frequency, learning interval, attention degree and learning centrality of knowledge unit (KU). The feature of learning centrality of KU is based on our team's previous research [14], in which the inheritance and development of knowledge is seen as a stochastic dynamic migration process of the semantic information in knowledge map.

The above contributions will help engineers and researchers in this community to develop more effective recommendation algorithms, especially for M-learning.

The rest of the paper is organized as follows. Section 2 presents relevant research on learning path recommendation. Section 3 provides the related definitions and describes the multi-constraint learning path model. Section 4 implements the multi-constraint learning path recommendation algorithm. Section 5 presents experimental result and analysis. Questionnaire and subjective evaluation are used for verifying hypothesis mentioned above and for the effectiveness of the recommended learning paths for different learning scenarios. Section 6 presents conclusion and our future work.

2. Related work

The existing representative research on learning path recommendation can be divided into three categories, based on learner characteristics, semantics relations and cognitive relations between knowledge units.

2.1. Learning path recommendation based on learner characteristics

A learning path recommendation method is based on learner characteristics to generate a customized learning path by analyzing characteristics of the learning behavior observed during learning process. Yang and Dong [30] proposed a learning path model that allows learning activities and the assessment criteria of their learning outcomes to be explicitly formulated by the Bloom's Taxonomy. Salehi and Kamalabadi [22] proposed a new material recommender system framework based on sequential pattern mining and multidimensional attribute-based collaborative filtering (CF). Lin et al. [14] presented a fusion of personalized learning and game based learning. They developed a personalized innovative learning system based on decision tree to provide personalized learning path for learners. Dwivedi et al. [5] present an effective learning path recommendation system (LPRS) for e-learners through a variable length genetic algorithm (VLGA) by considering learners' learning styles and knowledge levels. Basu et al. [2] presented a user model based system which takes into account a number of parameters such as learner's preference, previous performance, requirement of credit points, and availability of time to recommend a learning path. Bendahmane et al. [3] presented a competence based approach (CBA) derived from learning data, learner's characteristics and their expectations. In this approach, learners were clustered and traced, and finally proper learning paths were presented. Salehi and Kamalabadi [21] introduced learner preference tree (LPT) in which the multidimensional attributes of material, rating of learners, ordered and sequential patterns of the learner's accessed material were put together into the model. The model uses mixed, weighted, and cascade hybrid methods to form the final recommended learning paths.

The above learner-based methods mainly recommend learning paths from the learner's point of view. They are based on the parameters such as a learner's preference, his/her previous performance, learning targets, and learning abilities etc. There is a tendency that e-learners often use time fragments to learn and it led to their fragmented and inconsecutive learning behavior. This increases the difficulty of mastering the knowledge. So a consecutive learning path is preferred if the continuity of the knowledge elements is considered. The learner-based modeling methods have partly taken into account the order of the knowledge elements in the learning process, yet it is impossible to ensure that the knowledge units are provided continuously and systematically.

2.2. Learning path recommendation based on semantic relations

Learning path recommendation is essentially different from the general merchandise recommendation and the movie recommendation. The key factors that affect the results of commercial product recommendation mainly depend on the user's evaluation, rating, browsing, or collection which has limited relationship with the user's understanding and cognition. On the other hand, the massive educational resources have some inherent semantic structure characteristics, which determines that the knowledge elements have logical sequences [15]. According to the theory of connectivism, learning is a process of constantly connecting knowledge nodes or resources. The internal relationship between knowledge elements has an important role in learning process.

Chu et al. [4] proposed a learning path generation algorithm based on ontology. In this method, the ontology base is established based on the relationships between knowledge elements, and then a learning path is recommended according to the relationship between them in the ontology. Ouf et al. [20] proposed the framework for smart e-learning ecosystem using ontology and SWRL. Tam et al. [25] presented an explicit semantic analysis, followed by enhancing the ontology analysis through concept clustering, and applied an optimizer to find an optimal learning path of involved concepts or modules. Shmelev et al. [23] combined the learner's model with the domain knowledge ontology of the learning resources for matching, and realized the personalized recommendation through the semantic recommendation algorithm. Al-Hassan et al. [1] proposed a hybrid semantic enhanced recommendation approach by combining the new Inferential Ontology-based Semantic Similarity (IOBSS) measure and the standard item-based CF approach. Tseng et al. [27] constructed the concept map for adaptive learning and provided educational recommender for individual students.

The key underpinning concepts of the above methods are to perform the semantic analysis to form a concept map among knowledge units. The advantage of the recommendation methods based on concept map is that they consider the inherent relation between knowledge units, however, due to the lack of considering the learning scenarios, the recommended results of the above methods cannot precisely match the e-learners' needs. Therefore, there is a need to create an effective recommendation approach that considers not only the semantic relationship between the knowledge units, but also the different learning scenarios.

2.3. Learning path recommendation based on cognitive relations

Most of the above learning path generation algorithms seldom consider the target knowledge units, and they ignore the impact of the cognitive relationship between knowledge units. The massive and heterogeneous learning resources lead to knowledge disorientation and cognitive overload for the learners. To overcome this difficulty, knowledge map is put forward as a novel learning resource organization mode. In a knowledge map, knowledge units in a course or a subject are organized as a big graph [13].

The knowledge map has been applied in knowledge management, storage, learning navigation, and so on. Hanewald [9] showed how to develop lifelong learning skills by creating a digital cooperation knowledge map. Guillaume et al. [6] proposed a learning path recommendation system based on graph theory, and a greedy algorithm is used to find local optimal solution for the shortest path. Wan and Niu [28] paid attention to learners' emotions and proposed a learner oriented recommendation approach based on mixed concept mapping and immune algorithm (IA). He modeled the learner oriented recommendation as a constraint satisfaction problem (CSP) which aims to minimize the penalty function of unsatisfied indexes. Zhao et al. [32] proposed an approach of recommending micro-learning path based on improved ant colony optimization algorithm. During the process of micro-learning, the proposed algorithm can detect learner's learning transitions of knowledge level, knowledge area and learning goal according to the operation of learner.

These learning path recommendation methods mentioned above are mainly based on knowledge map according to cognitive order. The sequences of the knowledge units have been well considered when the knowledge maps are formed. Therefore, the recommended learning paths will meet the general needs of the learners when they acquire new knowledge. Their work also requires further improvement when considering the diversity of learning scenarios, as the learners will put forward different requirements under different learning scenarios. An ideal recommendation algorithm should consider learners' learning process and learning abilities etc., and then provides diversified learning paths to different learning scenarios.

However, after conducting experiment, we have found that learners have a great randomness when they were learning through knowledge map. In this experiment, GSP algorithm [10] is used to find the frequent sequential pattern of knowledge elements based on single learning duration. The raw data consists of 2150 log items from 156 learners and 57 knowledge units. The mining results are $\{Seq_i\}=\{<\#1,\#2>, <\#1,\#8>, <\#2,\#3>, <\#1,\#3>, <\#1,\#6>, <\#2,\#4>, <\#2,\#7>, <\#6,\#8>, <\#8,\#5>, <\#3,\#6,\#8,\#9>, <\#3,\#4,\#6,\#9>, <\#5,\#6,\#8,\#9>, <\#1,\#4,\#6,\#10,>\ldots\}$ when setting the value of support [8] as 0.1. Obviously, we can find that there exist absent elements between a start knowledge unit to end knowledge unit of many sequence in $\{Seq_i\}$, when comparing one Seq_i with a corresponding partial path with the same start knowledge unit and the same end knowledge unit in knowledge map (as shown in Fig. 1). This can cause a great difficulty for learners to master the knowledge, especially when they cognitively make newly-learned knowledge units connected to the knowledge units they already mastered. To recommend appropriate learning path for learners to solve the problems mentioned above is one of the core research in this paper.

Based on learner's prior knowledge and learning goals, using knowledge map as unit of knowledge structure description tool, this paper proposes a multi-constraint learning path recommendation algorithm, which offers learning path recommendation according to the current learning scenarios and learning goals.

3. Definitions and requirement analysis

In this section, firstly, we give five definitions we will use in the following sections. Secondly, we explain two indexes of knowledge unit which represent attention degree and the learning centrality of KU. Thirdly, the multi-constraint model of learning path is proposed.

3.1. Related definitions and requirements analysis

Definition 1: Knowledge unit(KU): KU is a complete knowledge expression that cannot be further divided. For example, 'Java constant' is a knowledge unit of the course Java Programming.

Definition 2: Target knowledge unit: A target Knowledge Unit is a knowledge unit that a leaner wants to learn in a learning process.

Definition 3: Learning dependency: It refers to a kind of necessary dependency relation between knowledge units during a learning process. Fig. 2 shows an example of learning dependency between three knowledge units.

Definition 4: Knowledge map (KM): Knowledge map is a directed graph that regards knowledge units as nodes and the relationships between knowledge units as edges. Knowledge map is defined as follows:

$$KM = (KU, KE) \tag{1}$$

in which:

KU is a set of all knowledge units in a knowledge map, KE is a set of all relations between knowledge units in a knowledge map.

The adjacency matrix of knowledge map KM is referred as $C = (c_{ij})_{n*n},\ 0 < i \le n,\ 0 \le j < n,\ i \ne j$ and C_{ij} satisfies the follows.

$$\begin{cases} c_{ij} = 1\ (ku_i, ku_j) \in KE \\ c_{ij} = 0\ (ku_i, ku_j) \notin KE \end{cases} \tag{2}$$

$ku_i \in KU$, is a specific knowledge unit of the knowledge map, which is the basic knowledge unit with complete expression ability.

$ke_i \in KE$, is a learning dependency relation in the knowledge map, which refers to a relationship among prerequisite order, causality and case relation.

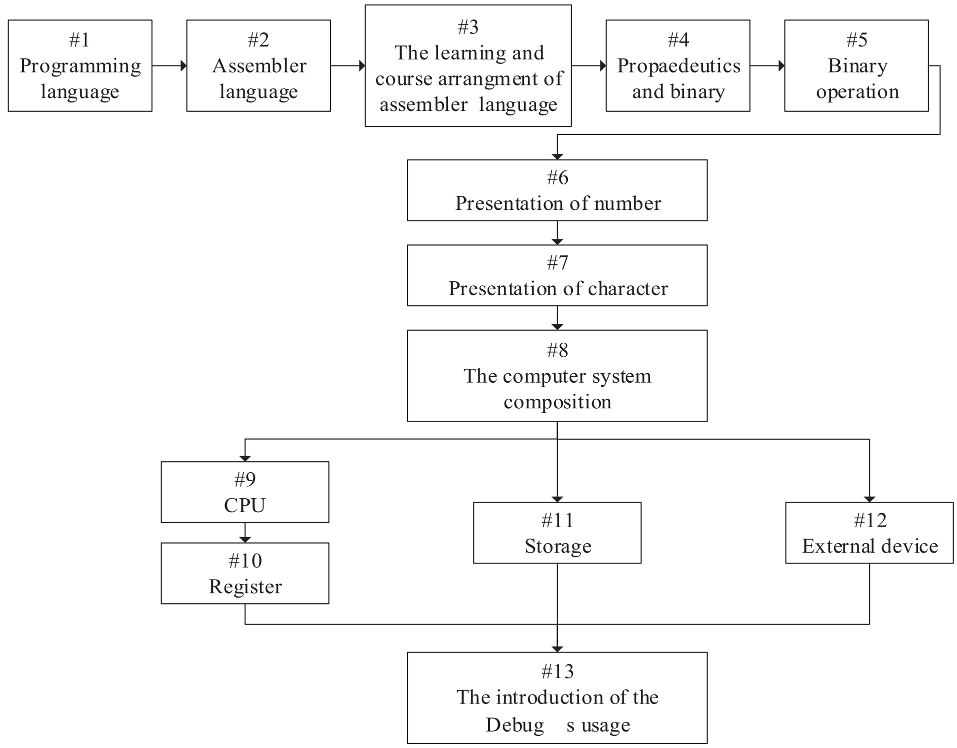


Fig. 1. Sub KM of the course Assembly language.

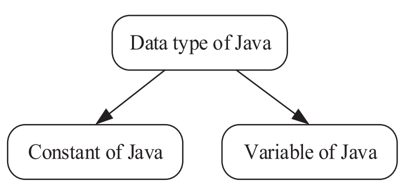


Fig. 2. Example of learning dependency between knowledge units.

Definition 5: Learning path: Learning path is a sequence of multiple knowledge units which is determined by the target knowledge unit, and denoted by $p = \{ku_i, ku_j \ldots, ku_m\}$, in which $ku_i, ku_j, \ldots, ku_m \in KU$.

In a knowledge map, there would be multiple learning paths from the initial knowledge unit to the target knowledge unit. The goal of this paper is to recommend learning path satisfying multiple constraints based on the learner's learning log.

3.2. Attention degree of KU

Taking Assembly language course as an example, we have selected 156 learners and have analyzed the total learning frequency and total learning duration for 57 KUs. The result is shown in Fig. 3.

As shown in Fig. 3, some KUs' total learning frequency and total learning duration is significantly higher than other KUs. Therefore, we define attention degree as a measure of the learner's interest in a KU. For a KU, its attention degree is actually the ratio between all learners' average learning duration and the original duration of the KU itself.

$$h_i = \frac{T_{isum}}{F_{isum} * t_{i0}} \tag{3}$$

T_{isum} is the total learning duration of ku_i, F_{isum} is the total learning frequency of ku_i and t_{i0} is the original duration of ku_i.

3.3. Learning centrality of KU

From the view of KUs' semantic inheritance, some KUs play a core role in the learning process, and they have high semantic contribution ability to the other KUs' in a learning process. So when a critical learning path recommendation is proposed, the key KUs should be taken into consideration. Based on our team's previous research [29], an absorbing state Markov model for the semantic migration of KM is constructed, and then the KU's learning centrality d is used to describe the statistical feature of the degree of the importance of KU's.

To calculate KU's learning centrality, we set the current KU as ku_i. q_{ij} is define as the probability of the semantic information migrated to its cognitive consequent ku_j. The KU's development potential shows its degree of being semantically inherited, which is defined as the KU's out-degree e_i^{out}. For ku_i, the larger out-degree means more cognitive consequents, meanwhile, probability of each candidate cognitive consequent to be chosen is smaller. In the KM, e_i^{in} is defined as the KU's in-degree. The larger in-degree means more cognitive antecedents, which leads to hard to learn that KU. After the calculating probability of the semantic migration, q_{ij}, we obtain the absorbing state Markov model for the semantic migration of KM. Thereafter the KU's learning centrality d is calculated.

3.4. Multi-constraint model of learning path

It is imperative that not only the learners' targets but also the learning scenarios should be considered during a learning process [26]. In this paper, we proposed four basic learning scenarios based on the time when the learner starts learning and the learning status of the target knowledge unit.

a. Initial learning: First time for studying the target KU in ordinary study means that the learners study the target KU first time, and there is an adequate time for studying it.

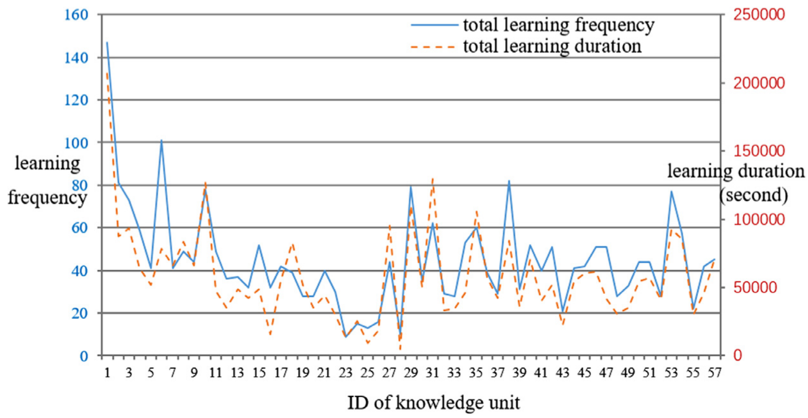


Fig. 3. Total learning frequency and total learning duration for each KU of Assembly language.

b. Usual review: Reviewing the target KU in ordinary study means that the learners' ordinary review of the knowledge that they have already studied, and there is adequate time for reviewing it.
c. Pre-exam learning: First time for studying the target KU just before the exam means the learners study the target KU first time, and there is not much time for studying it.
d. Pre-exam review: Reviewing the target KU just before the exam means that the learners review those knowledge units they had already studied, and there is not much time for studying it.

3.4.1. Requirement analysis of multi-constraint learning path

Based on the learners' log and the four kinds of learning scenarios presented above, we proposed seven kinds of scenario-oriented requirements for the learning path according to characteristics of learners and of knowledge map.

(1) Complemented learning path
It is a learning path in which the user's goal graph contains most KUs which have not been effectively learned. Here the user's goal graph means knowledge sub-graph which is generated by the paths from the starting KU to the target KU. Note that, a KU is considered to be learned effectively, if and only if its total learning duration is more than 80% of the its original duration.
(2) Shortest learning path
A learning path in the user's goal graph contains the least number of KUs.
(3) Shortest duration learning path
A learning path in the user's goal graph contains the shortest total duration of KUs.
(4) Critical learning path
A learning path in the user's goal graph contains maximum learning centrality of KUs.
(5) Easy learning path
A learning path in the user's goal graph contains the KUs of highest learning frequency. The higher learning frequency of a KU, the more familiar a learner is with the KU. Total frequency of the KUs is used as the measurement to determine whether the learning path is easy to learn or not.
(6) Complete learning path
A learning path in the user's goal graph has more KUs that have not been learned.
(7) More-hotspot learning path
A learning path in the user's goal graph contains the KUs which have the highest attention degree.

In these seven paths, the complemented learning path, easy learning path, complete learning path and more-hotspot learning path, are proposed based on the learner's characteristics in learning log. The learning path with shortest duration, shortest learning path and critical learning path are based on the inherent attributes of the course's knowledge map.

3.4.2. Construction of a multi-constraint learning path recommendation model based on knowledge map

According to the seven kinds of learning path requirements, formal representations of these learning paths' constraints factors are shown in Table 1.

To satisfy the learner's diversity of learning scenario, the multi-constraint model of learning path recommendation is constructed as shown in Eq. (4). $min_{i=0\ldots k} f(p_i)$ outputs the recommended learning path according to specific constraint conditions.

$$\begin{aligned} min_{i=0\ldots k} f(p_i) = min_{i=0\ldots k}(\alpha * g(f_1(p_i)) + \beta * g(f_2(p_i)) \\ + \chi * g(f_3(p_i)) + \delta * g(f_4(p_i)) + \gamma * g(f_5(p_i)) \\ + \lambda * g(f_6(p_i)) + \zeta * g(f_7(p_i)) \end{aligned} \quad (4)$$

In Eq. (4), p_i is a specific path in user's goal graph. k is the total number of these paths. g is a function for calculating min-max standardization of the seven learning paths' constraint factors. α, β, χ, δ, γ, λ, ζ are the constraints for recommending learning paths for learners, which are the weighted coefficients of the constraint factors. By adjusting the weighted coefficients, we can gain different learning path in different constraint conditions. For example, when $\alpha = 1$, $\beta = 0$, $\chi = 0$, $\delta = 0$, $\gamma = 0$, $\lambda = 0$, $\zeta = 0$, the number of KUs which are not effectively learned in the learning path are less, and the value of f is smaller, and this situation is suitable for the complemented learning path. Similarly remaining six paths can be calculated. When $\alpha = 0$, $\beta = 0.5$, $\chi = 0$, $\delta = 0$, $\gamma = 0.5$, $\lambda = 0$, $\zeta = 0$, the learning path is shorter, the total learning frequency of the KUs in the path is bigger, and the value of f is smaller, and this situation is suitable for the shorter length and easier learning path, which is defined as a quick learning path that satisfies two constraint factors. We use Φ to denote the constraint conditions, $\Phi = \{\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta\}$.

4. Multi-constraint learning path recommendation algorithm

According to the model proposed in Section 3.4.2, we design and realize the multi-constraint learning path recommendation algorithm as shown in Algorithm 1 in the following subsection.

Table 1
Learning paths' constraints factors.

Learning path	Constraint factor	Parameters
Complemented learning path	$f_1(p_i) = \frac{n_r}{m_r(p_i)+1}$	n_r is the number of KUs which are not effectively learned in the learner goal graph.$m_r(p_i)$ is the number of KUs which are not effectively learned in a typical learning path p_i.
Shortest learning path	$f_2(p_i) = \frac{l(p_i)}{l_M}$	$l(p_i)$ represents number of KU of a learning path p_i, which defines path length.l_M is the longest path length of the user's goal graph.
Shortest learning duration path	$f_3(p_i) = \frac{l_t(p_i)}{l_{tM}}$	$l_t(p_i)$ is the duration of a path, namely, the total video duration of KUs in a path.l_{tM} is the longest duration of a path in a user's goal graph.
Critical learning path	$f_4(p_i) = \frac{l_{dM}}{l_d(p_i)}$	$l_d(p_i)$ is the sum of KUs' learning centrality of a path p_i.l_{dM} is the biggest $l_d(p_i)$ of the path in a user's goal graph.
Easy learning path	$f_5(p_i) = \frac{l_{WM}}{l_W(p_i)}$	$l_W(p_i)$ is the total learning frequency of KUs in the path p_i.l_{WM} is the biggest $l_W(p_i)$ of the path in a user's goal graph.
Complete learning path	$f_6(p_i) = \frac{n_u}{m_u(p_i)+1}$	n_u is he number of KUs which are not learned in the learner goal graph.$m_u(p_i)$ is the number of KUs which are not learned in the learning path p_i.
More-hotspot learning path	$f_7(p_i) = \frac{l_{hM}}{l_h(p_i)}$	$l_h(p_i)$ is the sum of the KUs' attention degree in a learning path p_i.l_{hM} is the biggest $l_h(p_i)$ of the learning path in the user's goal graph.

Algorithm 1 Multi-constraint learning path recommendation algorithm based on KM. $p = RecommendPath(C, \{UserLog\}, S, E, \Phi)$.

Input:
The adjacent matrix of the KM $C = [C_{ij}]_{n*n}$
the user's learning log $\{UserLog\}$
the start KU S
the target KU E
the constraint condition $\Phi = (\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta)$
Output:
The constrained learning path recommendation p
Use adjacent matrix to represent a course's KM;
Apply $G(id, t) = Genleagra(C, \{UserLog\}, id)$ to generate user's learned graph;
Apply $G'(id, t, S, E) = FormAllPath(C, G(id, t), S, E)$ to get all paths between S and E;
Apply $p = RecommendPath(G'(id, t, S, E), \Phi)$ to find the recommended learning path under the constraint condition $\Phi = (\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta)$.

4.1. Design of multi-constraint learning path recommendation algorithm based on KM

The architecture of the multi-constraint learning path recommendation algorithm is shown in Fig. 4.

The description of the algorithm is shown in Algorithm 1.

(1) Use adjacent matrix to represent a course's KM
The KM's adjacent matrix shows the learning dependency between the KUs. For a directed KM which includes n KUs, matrix $C = (c_{ij})_{n*n}$ is defined. If C satisfies Eq. (2), we call it as adjacent matrix of the course's KM.

(2) Generate user's learned graph
According to the user's learning log, we mark the studied knowledge units in knowledge map and get the user's learned graph, denoted as $G(id, t)$. A weight adjacent matrix is used to represent $G(id, t)$. The weights include the learned label, the original duration of the KU video, the learning centrality of KU, the learning frequency of KU and the attention degree of KU.

$$G(id, t) = \{ku, ke, (r, t_{i0}, d, w, u, h)\} \tag{5}$$

ku represents knowledge unit in KM, $ku \in KU$. ke represents the dependency relationship between ku, $ke \in KE$. r represents learned mark of ku. $r = 1$ means ku has already been effectively finished while $r = 0$ means ku has not been effectively finished. t_{i0} is the original video duration of ku. d is the learning centrality of ku. w is the learner's learning frequency. u is the learner's learning label of ku, $u = 1$ means ku has already been learned, while $u = 0$ means ku has not been learned. h is the attention degree of ku. The generation algorithm of the user's learned graph is shown in Algorithm 2.

Algorithm 2 Generation of user's learned graph. $G(id, t) = Genleagra(C, \{UserLog\}, id)$.

Input:
The adjacent matrix of the KM $C = [c_{ij}]_{n*n}$ (the relationships between any pairs of ku_i and ku_j. C_{ij}=1 if $(ku_i, ku_j) \in KE$ while C_{ij}=0 if $(ku_i, ku_j) \notin KE$)user's learning log$\{UserLog\}$ (each of which records the learner's student number, the number of the ku he learnt, the length of the learning time, it's original video duration and some other information.)
Output:
User's learned graph $G(id, t)$ (consists of ku_i. Each ku_i in the graph has the following attributes: r (the if-effectively-learnt label of ku_i), t_{i0} (the original video duration of ku_i), d (learning centrality of ku_i), w (learning frequency of ku_i), u (if-ever-learnt label of ku_i), h (the attention degree of ku_i))
for all $ku_i \in KU$ **do**
  select all the records of ku_i from $\{UserLog\}$ and form a subset $\{UserLog\}_{ku_i}$;
  $r = getR()$; {check if ku_i has been efficiently learned}
  $t_{i0} = getT()$; {get the original video duration of ku_i}
  $d = calculateD()$; {calculate the learning centrality of ku_i}
  $w = calculateW()$; {calculate the learning frequency of ku_i}
  $u = getU()$; {check if ku_i has already been learned}
  $h = calculateH()$; {calculate the attention degree of ku_i}
  record r, t_{i0}, d, w, u, h in ku_i and update ku_i in $G(id, t)$;
end for
return the user's learned graph $G(id, t)$

(3) Constructing user's goal graph
All paths from the start KU S to the end KU E in $G(id, t)$ are produced based on the depth-first traversal algorithm. We can construct the user's goal graph $G'(id, t, S, E)$ as shown in Algorithm 3.

(4) Generating constraint learning path
According to the constrained condition $\Phi = \{\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta\}$ and the multi-constraint model in Eq. (4), all values f of all the learning paths in the user's goal graph $G'(id, t, S, E)$ are calculated. Then we get the learning path recommendation p which satisfies all constrained conditions. The generation algorithm of the constrained learning path is shown in Algorithm 4.

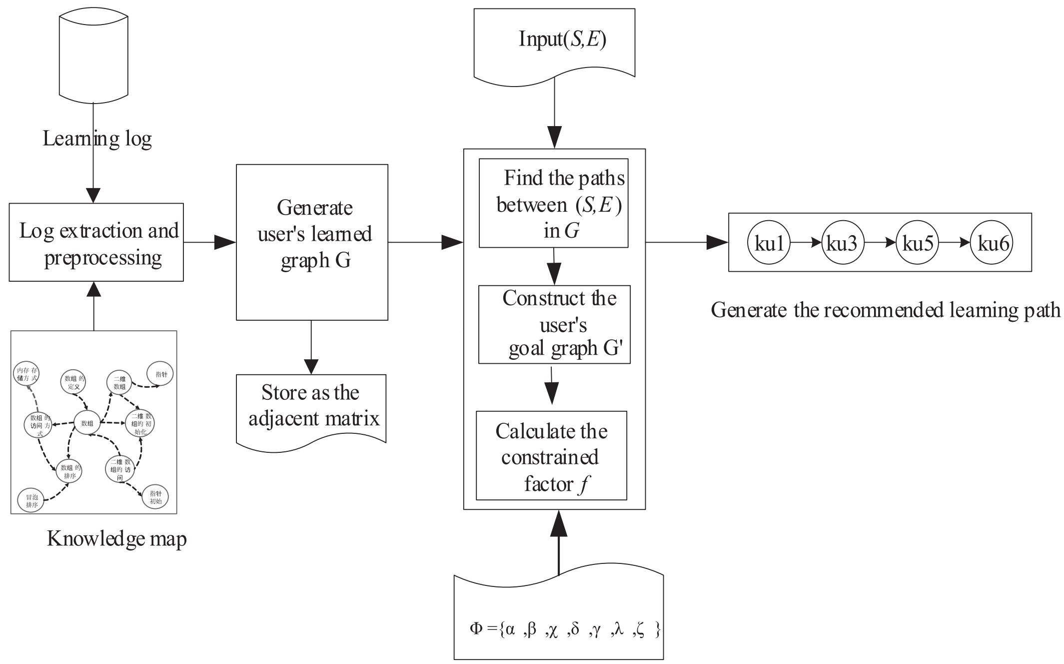


Fig. 4. Architecture of multi-constraint learning path recommendation algorithm based on KM.

Algorithm 3 Generation of user's goal graph. $G'(id, t, S, E) = FormAllPath(C, G(id, t), S, E)$.

Input:

The adjacent matrix of the KM $C = [c_{ij}]_{n*n}$
the user's learned graph $G(id, t)$
the start KU S
the target KU E

Output:

$G'(id, t, S, E)$ (all paths between (S, E))

initialize a Boolean array *visit* with the length n and the initial value *false*
$pathf(s, e)$ {
mark s visited in *visit* array (set the corresponding value to *true*)
record s in *path*;
if $s == e$ **then**
  return *path*;
else
  for all w connect to s **do**
    if w is not visited **then**
      recursively call the function $pathf(w, e)$;
    end if
  end for
end if
}
return $G'(id, t, S, E) = allpath$

Algorithm 4 Generation of constrained learning path. $p = RecommendPath(G'(id, t, S, E), \Phi(\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta))$.

Input:

The user's goal graph $G'(id, t, S, E)$
constrained condition $\Phi(\alpha, \beta, \chi, \delta, \gamma, \lambda, \zeta)$

Output:

The constrained learning path recommendation p

for *all* $path p_i, i = 0 \cdots k$ **do**
  $f_1(p_i) = n_r/(m_r(p_i) + 1)$; {calculate the constraint factor of the complemented learning path}
  $f_2(p_i) = l(p_i)/l_M$; {calculate the constraint factor of the shortest learning path}
  $f_3(p_i) = l_t(p_i)/l_{tM}$; {calculate the constraint factor of the shortest duration learning path}
  $f_4(p_i) = l_{dM}/l_d(p_i)$; {calculate the constraint factor of the critical learning path}
  $f_5(p_i) = l_{wM}/l_w(p_i)$; {calculate the constraint factor of the easy learning path}
  $f_6(p_i) = n_u/(m_u(p_i) + 1)$; {calculate the constraint factor of the complete learning path}
  $f_7(p_i) = l_{hM}/(l_h(p_i) + 1)$; {calculate the constraint factor of the more-hotspot learning path}
  $f(p_i) = \alpha * g(f_1(p_i)) + \beta * g(f_2(p_i)) + \chi * g(f_3(p_i)) + \delta * g(f_4(p_i)) + \gamma * g(f_5(p_i)) + \lambda * g(f_6(p_i)) + \zeta * g(f_7(p_i))$;
end for
return $p = min_{i=0 \cdots k} f(p_i)$

4.2. Realization of the multi-constraint learning path recommendation algorithm

This paper uses the KM of Assembly language course from the DLC of our university. Table A.1 shows the original video duration, the calculation results of the attention degree and the learning centrality of each knowledge unit in Assembly language course. Let 'binary operation' be the start KU S, and 'pseudo-operation' be the target KU E. The learned KUs' weight information in (S, E) of the No. 0035 learner is shown in Table 2.

According to the KM of Assembly language and the user's learned graph, the user's goal graph can be constructed. Finally, we generated multi-constraint learning paths for No.0035 learner, which are shown in Table A.2.

5. Experiment result and analysis

This section describes our experiment, its design and results to verify the hypotheses we have proposed and evaluate the learners' preference of the recommended learning paths.

Table 2
Characteristic values of the learned KUs in (*S*, *E*) of No. 0035 learner.

Sequence number of KU i	Effectively learned r	Original video duration t_{i0}(second)	Learning centrality d	Learning frequency w	Learning label u	Attention degree h
#5	1	603	2.2457	6	1	1.0963
#8	1	997	2.9754	11	1	1.7119
#9	1	1298	3.4665	3	1	1.1591
#10	0	1674	4.6880	2	1	0.9719
#11	0	1317	1.4234	2	1	0.7323
#14	1	619	1.5728	11	1	2.0063
#15	1	719	1.6554	3	1	1.3100
#19	1	1040	1.9008	4	1	1.7901
#22	0	1251	2.4341	5	1	0.7913
#23	1	1382	2.6987	7	1	1.0757

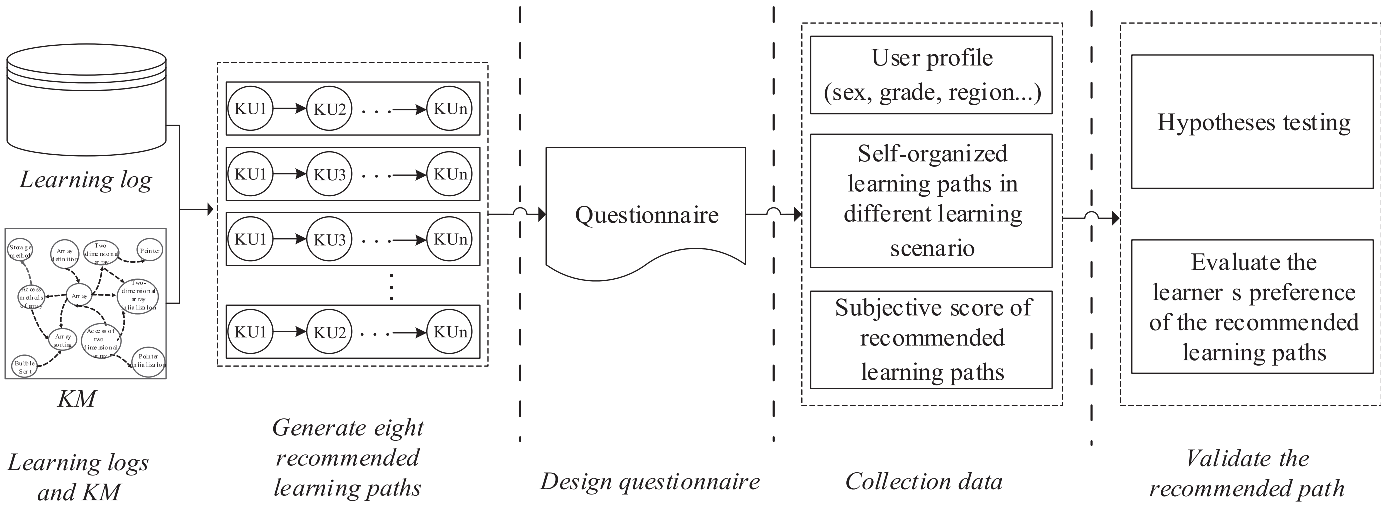


Fig. 5. Experiment process for verifying the hypotheses and the recommendation algorithm of learning paths.

Table 3
Learners' self-organized learning paths in four different scenarios.

Learning scenario	Self-organized learning path
1 Initial learning (*E* has not been learned)	
2 Usual review (*E* has been learned)	
3 Pre-exam learning (*E* has not been learned)	
4 Pre-exam review (*E* has been learned)	

5.1. Experiment process and questionnaire

This section describes the experiment process and the questionnaires results that we used to verify the hypotheses. An experiment process for verifying the hypotheses and the recommendation algorithm of learning paths is shown in Fig. 5.

As shown in Fig. 5, firstly, using the KM and learning logs, we generate different recommended learning paths in various learning scenarios, which will be used in the questionnaire. Secondly, we design a questionnaire, in which learners' basic information, self-organized learning paths in different learning scenarios and subjective score of the eight recommended learning paths in each learning scenario can be collected. Thirdly, we send out questionnaires and then collect the data. Finally, we verify the hypotheses we proposed and evaluate the learners' preference of the recommended learning paths.

The questionnaire includes three parts. The first part is used to collect some basic information about the learners. The second part enables learners to give their self-organized learning paths from the start KU *S* #5, to the target KU *E* #30, in different learning scenarios, and the content of the second part is shown in Table 3. The third part enables the learners to select their subjective score of the eight recommended learning paths in each learning scenario. We use five-point scale. Five points represent very satisfied, four points means satisfied, three points means general, two points means unsatisfied, as well as one point means very unsatisfied. Table 5 presents an example of learners' subjective scoring to the complemented learning path.

The graph in Table 4 is a partial knowledge map of a course Assembly language, whose detailed information is as follows: Set the start KU *S* as #5 (the operation of binary) and target KU *E* as #30 (pseudo-operation 2). For example, a learning record from *S* to *E* is represented as $\{\#5(2), \#8(11), \#9(3), \#10(2), \#11(2), \#13(14), \#14(11), \#19(4), \#22(5), \#23(7)\}$ in which #5(2) means the KU ID is 5, and the learning frequency is 2.

5.2. Experiment result

In this part, firstly, we analyze some basic information about 110 learners. Secondly, two hypotheses we proposed are verified. Then, we evaluate the learners' preference of the recommended learning paths. Based on the evaluation results, we conclude the strategy of the recommended learning path.

5.2.1. Hypothesis testing

The experiment carried out received 110 questionnaires responses of which 105 responses were valid. Figs. 6–Fig. 8 show some of information of 105 DLC-learners. The Batch number in Fig. 7 means DLC-learner's admission date.

We verify the two hypotheses we proposed in the introduction section. The first one is that a learner has different learning path requirements in different learning scenarios and the second is that

Table 4
Learners' subjective scoring of the recommended learning paths in four learning scenarios.

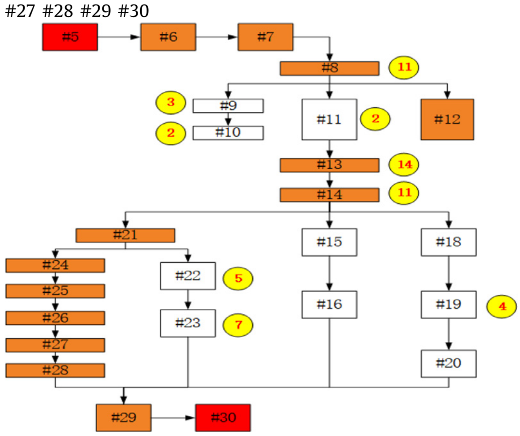
Complemented learning path	Initial learning score(1–5)	Usual review score(1–5)	Pre-exam learning score(1–5)	Pre-exam review score(1–5)
#5 #6 #7 #8 #12 #13 #14 #21 #24 #25 #26 #27 #28 #29 #30 				

Table 5
Edit distances between a learner's four self-organized learning paths and the eight recommended learning paths.

	Self-organized learning path in initial learning	Self-organized learning path in usual review	Self-organized learning path in pre-exam learning	Self-organized learning path in pre-exam review
Complemented learning path	2	5	5	5
Shortest learning path	5	4	5	4
Shortest duration learning path	4	1	5	2
Critical learning path	5	2	4	0
Easy learning path	3	0	5	2
Complete learning path	0	5	5	5
More-hotspot learning path	3	4	2	2
Quick learning path	2	4	3	1

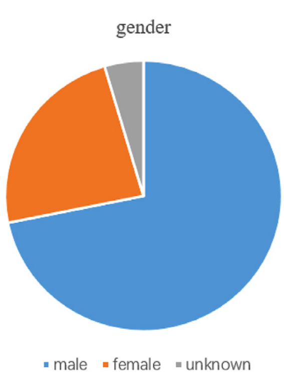


Fig. 6. Learners' gender information.

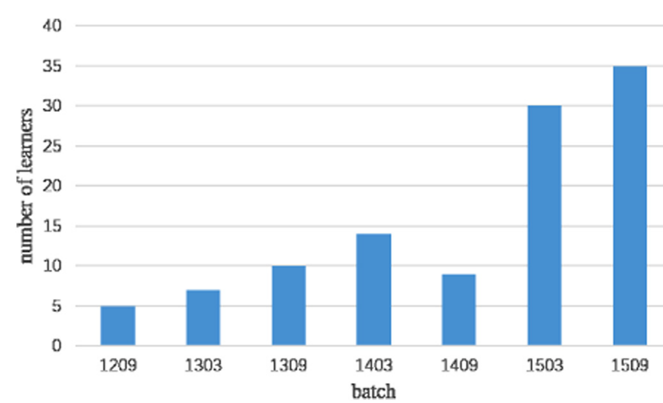


Fig. 7. Learners' batch information.

different learners have similar learning path requirements in the same learning scenario.

Edit distance algorithm [19] (see algorithm in Appendix B) is used to calculate the similarity between the learners' self-organized learning paths and the eight recommended learning paths. If the value of edit distance between learner's self-organized learning path and one of the eight recommended learning paths is zero, it means that the two paths are identical. A bigger value of the edit distance of the two paths means a larger difference between them.

For each learner, we calculate all the edit distances between his/her four self-organized learning paths and the eight recommended learning paths. Table 5 shows the calculation result of a learner.

(1) Verification of Hypothesis I
According to the result shown above, for each learner, we take one of the eight recommended learning paths, which has the smallest edit distance, as his/her required learning path in one specific learning scenario. Here are some examples of learners' requirements in different learning scenarios. In Table 6, P1–P8 refer to eight kinds of constraint learning paths.

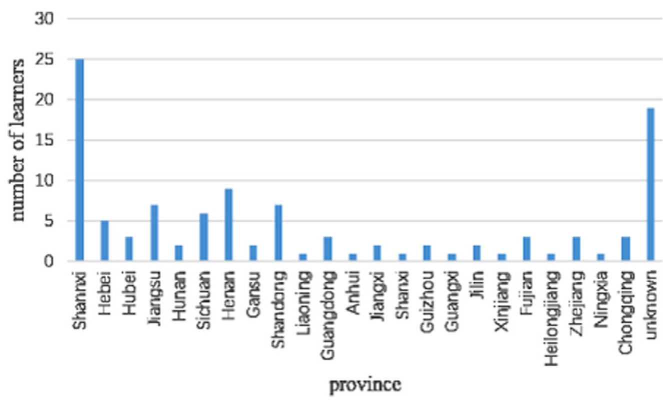


Fig. 8. Learners' batch information.

Table 6
Example of learners' path requirements in different learning scenarios.

	Initial learning	Usual review	Pre-exam learning	Pre-exam review
Learner A	P6	P5	P4	P4
Learner B	P4	P6	P2	P8
Learner C	P6	P4	P4	P7

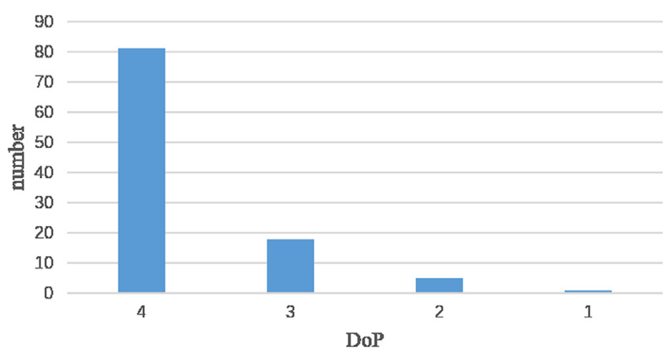


Fig. 9. Number of learners in different DoP.

We propose an indicator called diversity of paths (DoP) to represent the diversity of a learner's choices of learning paths in different learning scenarios. If four paths are completely different, the value of DoP is 4. If and only if two in the four paths are identical, the value of DoP is 3. If there exist only two different paths in the four paths, the value of DoP is 2. And if all the four paths are identical, the value of DoP is 1. For example, in Table 6, DoP of learner A and learner C is 3, DoP of learner B is 4. In Eq. (6), function *UNIQUE*() is used to find the distinct choices (learning paths) of one learner, and function *LENGTH*() is used to calculate the number of the distinct choices.

$$DoP = LENGTH(UNIQUE)([learner's\ choices\ in\ different\ learning\ scenarios]) \tag{6}$$

As shown in Fig. 9, we find that 77% learners have DoP value 4, and 17% learners have DoP value 3. Therefore, we conclude that most of the learners have different learning path requirements in different learning scenarios.

(2) Verification of Hypothesis II

For each kind of recommended learning path, we count the number of learners in the four learning scenarios. The result is shown in Fig. 10.

As shown in Fig. 10, statistical distributions of learners' requirement are not the same for the four learning scenarios. For example, in the initial learning scenario, 50.47% of the learners' path requirements are 'complete learning path', which means that they wanted to learn more KUs that they had not learned. While in the pre-exam learning scenario, 30.47% learners required 'the shortest learning path', 26.67% learners required 'critical learning path'.

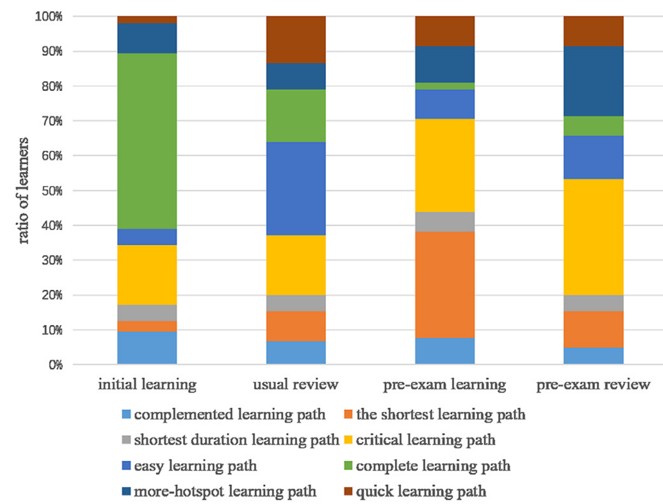


Fig. 10. Count of learning paths selected by learners in different learning scenarios.

Based on the statistical result of learners' requirement, our recommendation strategy is to recommend the Top-1 or Top-N learning paths to meet most of learners' requirement according to different learning scenarios. Namely, for the learning scenario in which there are more than 50% learners have the same learning path requirements, then we adopt the Top-1 learning path as the recommended learning path. In contrast, for the learning scenarios in which learners have not shown obvious preference of one learning path, we recommend more than one learning paths to those learners.

5.2.2. Evaluation of learners' preference to the recommended paths

To verify the effectiveness of the proposed recommendation strategy, we collect the learners' subjective scores for the recommended learning path using the questionnaire, and propose a satisfaction index that represents the learners' satisfaction for the recommended learning path in different learning scenarios.

$$sat(s, n) = \frac{\sum_{j=1}^{n} \sum_{i=1}^{N} score_i(s, p_j)}{N * 5 * n} \tag{7}$$

In Eq. (7), s represents learning scenarios. n represents the number of recommended learning paths for the learner, where $1 \le n \le 8$. N is the total number of participants $score_i.(s, p_j)$ represents the subjective score of a learner i to the recommended learning path p_j. For example, $sat(s, 1)$represents the learners' satisfaction to Top-1 recommended learning path. $sat(s, 2)$ represents the learners' satisfaction to Top-2 recommended learning paths.

We calculate all the values of $sat(s, n)$ based on 105 learners' subjective scores for the eight recommended learning paths in four learning scenarios. The results are shown in Table 7.

To determine the appropriate number of the recommended learning paths for each learning scenario, calculation is done as follows. We take x as an average of the differences between any pair of $sat(s, n)$ and $sat(s, n+1)$ where n ranges from 1 to 7. The simplified form of x is $(sat(s, 1) - sat(s, 8))/7$. We put upper limit of three on the number of the recommended paths, otherwise it may cause inconvenience to learners. So we set 2x as the threshold value. If $sat(s, 1) - sat(s, 3)$ is smaller than 2x, then three learning paths will be recommended. If $sat(s, 1) - sat(s, 3)$ is larger than 2x and $sat(s, 1) - sat(s, 2)$ is smaller than 2x, then two

Table 7
Values of $sat(s, n)$.

	$sat(s, 1)$	$sat(s, 2)$	$sat(s, 3)$	$sat(s, 4)$	$sat(s, 5)$	$sat(s, 6)$	$sat(s, 7)$	$sat(s, 8)$
Initial learning	0.705	0.638	0.615	0.605	0.589	0.567	0.551	0.488
Usual review	0.686	0.657	0.647	0.636	0.617	0.597	0.580	0.564
Pre-exam learning	0.678	0.676	0.641	0.617	0.5885	0.568	0.552	0.536
Pre-exam review	0.773	0.706	0.647	0.619	0.587	0.563	0.546	0.527

learning paths will be recommended. Otherwise, only one learning path will be recommended.

As shown in Table 7, we can conclude that the appropriate number of the recommended learning paths for initial learning scenario is one, for usual review scenario is two, for pre-exam learning scenario is three and for pre-exam review scenario is two. Comparing these results with the statistical result shown in Fig. 10, we can find that the number of these recommended paths are mostly consistent with learner's required learning path. This verifies that the recommended strategy must consider the learning scenarios we have proposed. It also verifies the existence of the similarity between learners' self-organized learning paths and the recommended learning paths in the four learning scenarios.

6. Conclusion

Facing with different learning scenarios and limited time, e-learners need various learning paths to follow, it is necessary to recommend an appropriate learning path to meet their needs and improve the learning efficiency of e-learners. This depends on perceiving the dynamic changes of the learning scenarios in time and making an accurate analysis of the user's fragmentation learning behavior.

The main contribution of the paper is that we proposed a novel approach to overcome the different learning path preferences of e-learners in different learning scenarios. Based on a knowledge map, the recommended learning path is generated by considering the combination of the domain knowledge structure and cognitive structure of the learners. Firstly, we present four different learning scenarios according to the e-learning process. Secondly, considering e-learner's different path requirements in different learning scenarios, eight kinds of constraint learning paths and their corresponding constraint factors are presented based on the characteristic analysis of the learners and resources. Thirdly, a multi-constraint learning path recommendation algorithm based on knowledge map for different learning scenarios is proposed. Finally, the experiments verify the similarity between the learners' self-organized learning paths and the recommended learning paths in the four learning scenarios. We can draw conclusion that the proposed algorithm is effective for e-learners.

Additionally, there are some limitations that require further improvement. Currently we have only considered four basic learning scenarios according to the learning process. In future work, scenarios could be divided into fine-grained scene according to user's demands. Furthermore, we will optimize the multi-constraint learning path recommendation model on basis of the learner's feedback, and then provide more personalized and more accurate learning paths for e-learners.

Acknowledgment

It is a project supported by the National Key Research and Development Program of China under Grant No. 2016YFB1000903, Innovative Research Group of the National Natural Science Foundation of China (61721002), Innovation Research Team of Ministry of Education (IRT_17R86), the National Science Foundation of China under Grant No. 61472315, 61402392, 61428206, 61532015 and 61532004, the project of China Knowledge Centre for Engineering Science and Technology, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant No. 2016JM6027, 2016JM6080, Online Education Research Foundation of MOE Research Center for Online Education under Grant No. 2016YB165, 2016YB169.

Appendix A

Table A.1
Relevant properties of knowledge unit of the course Assembly language.

No.	KU	Learning dependency	Duration (second)	Attention degree	Learning centrality
#1	Programming language	#2,#14	1825	2.0771	3.6651
#2	Assembly language	#3	924	1.8661	3.0211
#3	Arrangement of Assembly language	#4	1267	1.7172	2.9878
#4	Preliminary knowledge and binary	#5,#10,#11,#17,#19,#21,#26,#29	1677	0.6489	4.1352
#5	Binary operation	#6	603	1.0963	2.2457
#6	Presentation of number	#7	3690	0.4660	1.4369
#7	Presentation of character	#8	895	1.8010	3.1214
#8	Computer system composition	#9,#11,#12	997	1.7119	2.9754
#9	CPU	#10,#16	1298	1.1591	3.4665
#10	Register	#13,#14,#15,#16,#17,#26,#27,#29, #49,#51,#52	1674	0.9719	4.6880
#11	Memory	#13,#14,#15,#18,#19,#20,#21	1317	0.7323	1.4234
#12	External device	#13,#14,#23,#49	542	1.7928	2.0871
#13	The use of the Debug	#14	660	1.9926	1.0284
#14	Instruction sets and addressing modes	#15,#21	619	2.0063	1.5728
#15	Immediate addressing	#16	719	1.3100	1.6554
#16	Register addressing	#18,#29	262	1.9103	1.7865
#17	Effective address	#18,#21,#55,#56	915	1.4320	2.7512
#18	Memory operand addressing mode 1	#19	2582	0.8207	1.6547
#19	Memory operand addressing mode 2	#20	1040	1.7901	1.9008
#20	Instruction jump addressing	#29	2095	0.5937	1.7327

(continued on next page)

Table A.1 (*continued*)

No.	KU	Learning dependency	Duration (second)	Attention degree	Learning centrality
#21	Data transfer instructions	#22,#24	1802	0.6120	2.0283
#22	Stack and instructions	#23,#40	1251	0.7913	2.4341
#23	I/O instructions	#29,#50	1382	1.0757	2.6987
#24	Arithmetic instructions	#25	3380	0.4953	1.2748
#25	Logic instructions	#26	1657	0.4352	1.7667
#26	String processing instructions	#27	1944	0.5878	1.9750
#27	Control transfer instructions	#28,#34,#38	2698	0.8006	1.4728
#28	Processor control instructions	#29	415	1.1901	2.0867
#29	Pseudo operation 1	#30	2038	1.6835	2.4335
#30	Pseudo operation 2	#31	1593	0.8922	1.8976
#32	Assembly language processing	#33	692	1.6601	1.7657
#33	Debugging method of debug	#34,#38	844	1.4705	1.0972
#34	Structure of cycle program	#35	957	0.9037	2.5348
#35	Cycle programming method 1	#36	90	1.4771	1.4332
#36	Cycle programming method 2	#37	2789	0.5258	1.6754
#37	Multi-cycle programming	#40	1969	0.7383	1.1857
#38	Branch programming	#39	3326	0.3093	2.0896
#39	Method of skip list	#40	893	1.3009	2.3514
#40	Subprogram and stack	#41,#42,#45	1892	0.7162	1.6809
#41	Subprogram definition method	#42	2026	0.4988	1.8245
#42	Parameter passing	#43	2602	0.3912	2.5612
#43	Structure definition and enhanced process definition	#44	638	1.6736	1.1380
#44	Nested and recursive subprogram	#45,#47,#48	1774	0.7469	1.7347
#45	Macro Assembly 1	#46	1569	0.9191	2.0965
#46	Macro Assembly 2	#49	1720	0.7016	3.0992
#47	Conditional assembly	#49	3326	0.2453	2.8774
#48	Advanced language structure	#49	1033	1.0523	1.0651
#49	Data transfer of I/O devices	#50,#51	595	1.8034	1.6785
#50	I/O polling	#53	1297	0.9542	1.9227
#51	I/O interrupt	#52,#53,#54	1949	0.6664	2.0475
#52	Interrupt handlers	#53	1852	0.7927	3.1742
#53	Keyboard I/O	#54	2538	1.4742	2.7865
#54	Display I/O	#55	2370	0.6342	1.6799
#55	Assembler	#56	882	1.5307	1.0428
#56	Linker	#57	1097	1.0023	1.1327
#57	Assembly and C mixed programming		1698	0.9204	4.3257

Table A.2
Multi-constraint learning paths of No. 0035 learner.

Constrained conditions Φ	Path name	$P(\#5, \#30, G', \phi)$
{1, 0, 0, 0, 0, 0, 0}	Complemented learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, external device, the introduction of the Debug's usage, instruction system and addressing mode, data transfer instruction, arithmetical instruction, logic instruction, String processing instruction, control transfer instruction, processor control instruction, pseudo-operation1, pseudo-operation2}
{0, 1, 0, 0, 0, 0, 0}	Shortest learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, instruction system and addressing mode, directly addressing mode, pseudo-operation 1, pseudo-operation 2}
{0, 0, 1, 0, 0, 0, 0}	Shortest learning duration path	{binary operation, the presentation of number, the presentation of character, the computer system composition, instruction system and addressing mode, the addressing mode 1 of the operand in the RAM, the addressing mode 2 of the operand in the RAM, the addressing mode of the instruction jump, pseudo-operation1, pseudo-operation 2}
{0, 0, 0, 1, 0, 0, 0}	Critical learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, CPU, register, valid address, data transfer instruction, stack and its instruction, input and output instruction, pseudo-operation1, pseudo-operation 2}
{0, 0, 0, 0, 1, 0, 0}	Easy learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, CPU, register, instruction system and addressing mode, data transfer instruction, stack and its instruction, input and output instruction, pseudo-operation1, pseudo-operation 2}
{0, 0, 0, 0, 0, 1, 0}	Complete learning path	{binary operation the presentation of number, the presentation of character, the computer system composition, external device, the introduction of the Debug's usage, instruction system and addressing mode, data transfer instruction, stack and its instruction, input and output instruction, arithmetical instruction, logic instruction, String processing instruction, control transfer instruction, processor control instruction, pseudo-operation1, pseudo-operation 2}
{0, 0, 0, 0, 0, 0, 1}	hot pot learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, CPU, register, instruction system and addressing mode, direct addressing mode, register addressing mode, pseudo-operation1, pseudo-operation 2}
{0, 0.5, 0, 0, 0.5, 0, 0}	quick learning path	{binary operation, the presentation of number, the presentation of character, the computer system composition, instruction system and addressing mode, data transfer instruction, stack and its instruction, input and output instruction, pseudo-operation 1, pseudo-operation 2}

Appendix B

Algorithm B.1 The edit distance algorithm.

Input:

Learning path P_1, P_2, their path length len_1, len_2

Output:

Edit distance between the two paths *EditDistance*

$e[][]$=newint$[len_1 + 1][len_2 + 1]$; {build matrix of edit distance}
for i=0 to len_1 **do**
  $e[i][0] = i$; {initialize first column of the matrix}
end for
for j=0 to len_2 **do**
  $e[0][j] = j$; {initialize t first row of the matrix}
end for
for i=1 to len_1 **do**
  for j=1 to len_2 **do**
    $cost = P_1[i-1] == P_2[j-1]?$ 0:1; {judge whether $P_1[i-1]$ and $P_2[j-1]$ is the same}
    $deletion = e[i-1][j] + 1$; {calculate edit distance of deletion operation between $P_1[i]$ and $P_2[j]$}
    $insertion = e[i][j-1] + 1$; {calculate edit distance of insertion operation between $P_1[i]$ and $P_2[j]$}
    $substitution = e[i-1][j-1] + cost$; {calculate edit distance of substitution operation between $P_1[i]$ and $P_2[j]$}
    $d[i][j] = min(deletion, insertion, substitution)$; {select shortest edit distance between $P_1[i]$ and $P_2[j]$}
  end for
end for
$EditDistance = e[len_1][len_2]$; {get the edit distance between the two paths}
return *EditDistance*

References

[1] M. Al-Hassan, H. Lu, J. Lu, A semantic enhanced hybrid recommendation approach: a case study of e-government tourism service recommendation system, Decis. Support Syst. 72 (2015) 97–109.

[2] P. Basu, S. Bhattacharya, S. Roy, Online Recommendation of Learning Path for an E-Learner under Virtual University, Springer Berlin Heidelberg, 2013.

[3] M. Bendahmane, B.E. Falaki, M. Benattou, Individualized Learning Path Through a Services-Oriented Approach, Springer International Publishing, 2017.

[4] K.K. Chu, C.I. Lee, R.S. Tsai, Ontology technology to assist learners' navigation in the concept map learning system, Expert Syst. Appl. 38 (9) (2011) 11293–11299.

[5] P. Dwivedi, V. Kant, K.K. Bharadwaj, Learning path recommendation based on modified variable length genetic algorithm, Educ. Inf. Technol. 6 (2017) 1–18.

[6] D. Guillaume, B. Nabil, L. Franois, Graph theory based model for learning path recommendation, Inf. Sci. 251 (4) (2013) 10–21.

[7] Z. Gustavo, B. Nelson, Combining learning with patterns and geo-collaboration to support situated learning, in: IEEE International Conference on Systems, Man, and Cybernetics, 2012, pp. 925–930.

[8] J. Han, K. Micheline, J. Pei, Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann, 2012.

[9] R. Hanewald, Cultivating life-long learning skills in undergraduate students through the collaborative creation of digital knowledge maps, Proc. Soc. Behav. Sci. 69 (2012) 847–853.

[10] Y. Hirate, H. Yamana, Generalized sequential pattern mining with item intervals, J. Comput. Educ. 1 (3) (2006) 51–60.

[11] I. Kamsa, R. Elouahbi, F.E. Khoukhi, Optimizing collaborative learning path by ant's optimization technique in e-learning system, in: International Conference on Information Technology Based Higher Education and Training, 2016, pp. 1–5.

[12] E. Kurilovas, I. Zilinskiene, V. Dagiene, Recommending suitable learning scenarios according to learners' preferences: an improved swarm based approach, Comput. Hum. Behav. 30 (2014) 550–557.

[13] J. Lee, A. Segev, Knowledge maps for e-learning, Comput. Educ. 59 (2) (2012) 353–364.

[14] C.F. Lin, Y.C. Yeh, H.H. Yu, R.I. Chang, Data mining for providing a personalized learning path in creativity: an application of decision trees, Comput. Educ. 68 (2013) 199–210.

[15] J. Liu, L. Jiang, Z. Wu, Mining learning-dependency between knowledge units from text, VLDB J. 20 (3) (2011) 335–345.

[16] X. Liu, The Study of Goal-based Scenario(GBS) Experiential Learning Environment, Yangzhou University, 2012 Master's thesis.

[17] J. Lu, D. Wu, M. Mao, Recommender system application developments: a survey, Decis. Support Syst. 74 (C) (2015) 12–32.

[18] P.-S. Mar, M.-M. Pedro, A.-H. Carlos, S. Xavier, D.K. Carlos, Lessons learned from the design of situated learning environments to support collaborative knowledge construction, Comput. Educ. 87 (C) (2015) 70–82.

[19] W. Masek, M. Paterson, A faster algorithm computing string edit distances, J. Comput. Syst. Sci. 20 (1) (1980) 18–31.

[20] S. Ouf, M.A. Ellatif, S.E. Salama, A proposed paradigm for smart learning environment based on semantic web, Comput. Hum. Behav. 72 (2017) 796–818.

[21] M. Salehi, I.N. Kamalabadi, Hybrid recommendation approach for learning material based on sequential pattern of the accessed material and the learner's preference tree, Knowl. Based Syst. 48 (2013) 57–69.

[22] M. Salehi, I.N. Kamalabadi, Personalized recommendation of learning material using sequential pattern mining and attribute based collaborative filtering, Educ. Inf. Technol. 19 (4) (2014) 713–735.

[23] V. Shmelev, M. Karpova, A. Dukhanov, An approach of learning path sequencing based on revised bloom's taxonomy and domain ontologies with the use of genetic algorithms, Proc. Comput. Sci. 66 (2015) 711–719.

[24] C.H. Su, Designing and developing a novel hybrid adaptive learning path recommendation system (alprs) for gamification mathematics geometry course, Eurasia J. Math. Sci. Technol. Educ. 13 (2017).

[25] V. Tam, E.Y. Lam, S.T. Fung, A new framework of concept clustering and learning path optimization to develop the next-generation e-learning systems, J. Comput. Educ. 1 (4) (2014) 335–352.

[26] D. TME, J. DHE, Constructivism and the Technology of Instruction: A Conversation, Routledge, 2013.

[27] H.C. Tseng, C.F. Chiang, J.M. Su, Building an Online Adaptive Learning and Recommendation Platform, 2016, pp. 428–432.

[28] S. Wan, Z. Niu, A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm, Knowl. Based Syst. 103 (C) (2016) 28–40.

[29] X. Wang, A Personalized Recommendation Method of Learning Resource Based on Knowledge Map, Xi'an Jiaotong University, 2012 Master's thesis.

[30] F. Yang, Z. Dong, Learning Path Construction in E-learning, Lecture Notes in Educational Technology, 2017.

[31] C. Zhao, Z. Chen, Z. Huang, Recommendation algorithm and application of adaptive learning path, China Educ. Technol. (8) (2015) 85–91.

[32] Q. Zhao, Y. Zhang, J. Chen, An improved ant colony optimization algorithm for recommendation of micro-learning path, in: IEEE International Conference on Computer and Information Technology, 2017, pp. 190–196.