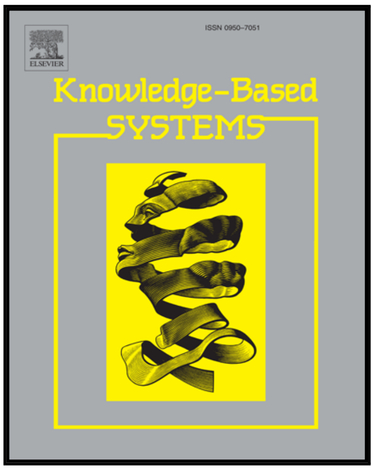
Accepted Manuscript



An E-learning Recommendation Approach Based on the Self-Organization of Learning Resource

Shanshan Wan, Zhendong Niu

PII: S0950-7051(18)30320-4
DOI: 10.1016/j.knosys.2018.06.014
Reference: KNOSYS 4386

To appear in: *Knowledge-Based Systems*

Received date: 31 October 2017
Revised date: 26 May 2018
Accepted date: 16 June 2018

Please cite this article as: Shanshan Wan, Zhendong Niu, An E-learning Recommendation Approach Based on the Self-Organization of Learning Resource, *Knowledge-Based Systems* (2018), doi: 10.1016/j.knosys.2018.06.014

An E-learning Recommendation Approach Based on the Self-Organization of Learning Resource

Shanshan Wan[a,b,c], Zhendong Niu[a,d,*]

[a]Beijing Institute of Technology, Beijing, China
[b]Beijing University of Civil Engineering and Architecture, Beijing, China
[c]Beijing Key Laboratory of Intelligent Processing for Building Big Data, Beijing 100044, China
[d]School of Computing & Information, University of Pittsburgh, PA, USA

Abstract

In e-learning, most content-based (CB) recommender systems provide recommendations depending on matching rules between learners and learning objects (LOs). Such learner-oriented approaches are limited when it comes to detecting learners' changes, furthermore, the recommendations show low adaptability and diversity. In this study, in order to improve the adaptability and diversity of recommendations, we incorporate an LO-oriented recommendation mechanism to learner-oriented recommender systems, and propose an LO self-organization based recommendation approach (Self). LO self-organization means LO interacts with each other in a spontaneous and autonomous way. Such self-organization behavior is conducive to generating a stable LO structure through information propagation. The proposed approach works as follows: firstly, LOs are simulated as intelligent entities using the self-organization theory. LOs can receive information, transmit information, as well as move. Secondly, an environment perception module is designed. This module can capture and perceive learner's preference drifts by analyzing LOs' self-organization behaviors. Finally, according to learners' explicit requirements and implicit preference drifts, recommendations are generated through LOs' self-organization behaviors. Based on application to real-life learning processes, the ample experimental results demonstrate the high adaptability, diversity, and personalization of the recommendations.

Keywords: personalized recommender system, e-learning, self-organization, diversity, adaptability

1. Introduction

E-learning recommender systems aim to recommend a sequence of items to learners, that is, to suggest the most efficient or effective paths through a plethora of learning resources to achieve a certain competence [1, 2]. However, over specification and excessive searching in e-learning recommender systems result in information overload. Learners do not have enough time to deal with these massive recommendations. In addition, the adaptability and diversity of recommendations are desirable in e-learning recommender systems, because learners' preferences and abilities keep changing, and also because the functionality of some learning resources for active learners keeps changing. The diverse and adaptive recommendations should be presented to motivate learning potential of learners and ensure a long-term learning experience. In this study, we aim to improve the adaptability and diversity of content-based (CB) recommendations.

*Corresponding author
Email address: zniu@bit.edu.cn (Zhendong Niu)

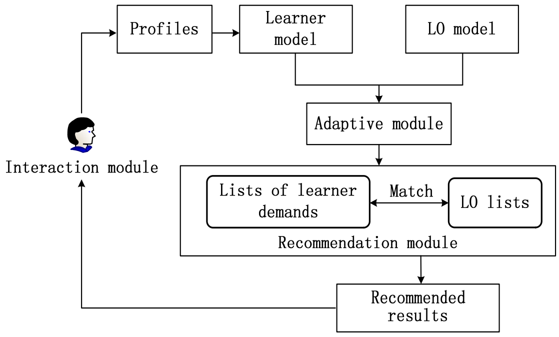


Figure 1: General architecture of adaptive CB recommendation systems

Generally, learning resources are filtered in three ways: CB, collaborative filtering (CF), and hybrid filtering (HF) [3, 4]. CF recommender systems compute the similarity computation between learners based on their rating and then make predictions[5, 6]. CF methods are known to suffer from the rating sparsity problem which occurs when the users or items have insufficient rating records [7]. In e-learning, the rating information is extremely sparse. The main reasons for this are a lack of motivation for learners to rate, a lack of scoring mechanisms, and the scheduled but limited learning time of learners [1]. The high data sparsity makes it difficult to apply CF techniques. CB recommender systems recommend relevant learning contents which are highly matched to learners' learning goals and preferences. In addition, CB recommender systems do not usually suffer from the first-rater problem, therefore, CB recommendation approaches are the only ones capable of recommending items not previously rated by any learner [8]. The learner's ability, goal, mood, and cognitive style are often used as criteria in CB recommendation systems [8, 9].

A few researches have been done to study learners' learning activities and give corresponding adaptive recommendations [10, 11, 12, 13, 14]. The architecture of adaptive CB recommender system is described in Fig. 1. In it, the learner model is extracted form learner profiles, and the adaptive module builds the adaptive concept selection rules or content selection rules through analyzing the relationships between the learner and LO models. The recommendation module executes the adaptive matching rules and provides the recommendations. The learning profiles of learners are updated by the feedback from the interaction module. Lu et al.[15] pointed out the matching rules are important for discovering associations between student requirements and the learning material tree. Nevertheless, CB recommender systems in e-learning still face the following problems [3]:

- The low adaptability of CB recommender systems makes it difficult for them to keeo up with the constant and rapid changes in e-learning environments. This problem refers to the fact that the recommendation approaches have a low ability to capture and perceive the changes in learners' preferences in an adaptive way [16]. The main reasons are: (1) the learner and LO models are often limited when it comes to comprehensively extracting the useful information of both learners and learning resources. As a result, the adaptive models are also limited; (2) the implicit or predictive demands are seldom considered in the existing adaptive models.

- The CB recommendation approach has the problem of low diversity due to the existence of excessive

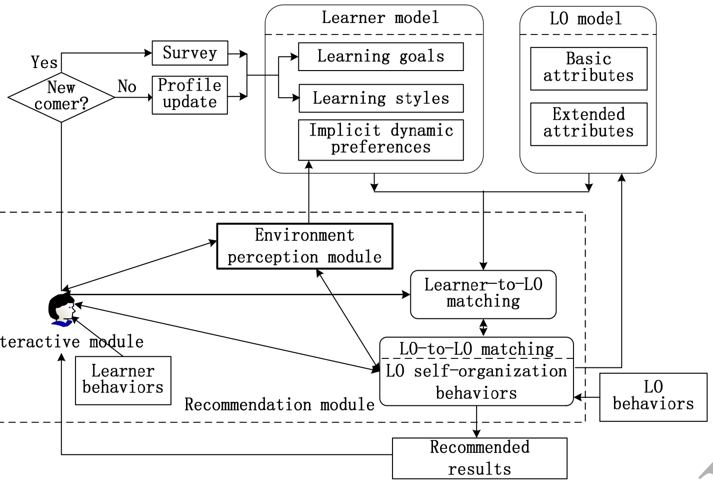


Figure 2: Architecture of LO self-organization based recommendation approach

similar recommendations. Many researches have mentioned that the diversity should be given higher priority than in previous recommender systems [17, 18]. The problem of low diversity has not been addressed enough in traditional CB recommendation systems. In e-learning, learners are bothered when very similar, almost identical, items appear in a recommendation list [19]. CB recommendation strategies are top-down and completely learner-oriented, as a result, learners only receive items that are similar to what they previously liked or showed to prefer. These approaches lack inherent mechanisms to improve the diversity of recommendations. Also, the highly matched resources are not likely to stimulate learners' interest or promote their learning potential.

In order to improve the adaptability and diversity of recommendations, we propose a recommendation approach based on the self-organization theory. The self-organization theory refers to the self-organizing phenomenon in which subsystems or individuals can form certain structures according to some rules without any external instruction. In our study, we model LOs as entities with the ability to receive information, transit information, and move towards target learners in a self-organizing way. The behaviors of LO individuals have the purpose to satisfy the requirements of active learners, which ensures the presentation of effective recommendations.

The architecture of the proposed recommendation approach is shown in Fig. 2. It includes the learner model, LO model, and the recommendation module and interactive module.

In learner model, learning styles are applied on describing a learner in e-learning environments. Learning styles are defined as how a learner behaves while learning, and what kind of preferences he/she has. We define learning styles using several aspects such as competency, preferences, attitude, learning experiences etc. In LO model, besides the basic attributes such as content, media or difficulty, some extended attributes are added such as LO states, as well as the statistics of LOs being operated and accessed.

In the recommendation module, LOs are modeled as intelligent entities which have autonomous and spontaneous behaviors, such as information transmitting and position changing. In the environment perception module, both the real-time operations of learners and the self-organization tendency of LOs are mined in order to perceive learners' preference drifts. When learners begin to study, recommendations are initialized according to learners' goals and learning styles. Next, learners' interactions with LOs become the stimuli for LOs' self-organization. According to learners' requirements and the preference drifts feedback from environment perception module, LO entities perform micro and bottom-up self-organization behaviors. The final recommendations will be presented when the self-organization process becomes stable. The LO self-organization behaviors will be continuously triggered by learners' learning activities. If learners achieve their goals, the recommendation stops.

According to Fig. 1, there are some characteristics which distinguish the proposed recommendation approach from previous approaches.

- The proposed recommendation strategy is based on both learner-LO and LO-LO relationships and not only based on learner-LO relationships. In most CB recommender systems, recommendations are provided to learners according to learner-LO matching rules. In the proposed approach, only few LOs are directly influenced by learners, and most of the LOs exhibit their behaviors according to LO-LO relationships. The appropriate reduction of learner-LO dependence will help address the problem of excessive recommendations. Consequently, the recommendations are expected to be more diversified.

- The self-organization based recommendation approach is bottom-up, spontaneous and distributed, rather than top-down, predefined and centralized. Our approach emphasizes the role of LOs' individual behavior. The fact is that LOs being operated by learners deliver effective information to other candidate LOs, and learners' operations carry the information of their real-time demands, abilities and learning experiences. The interactive information is generated, transmitted, received, and processed in a self-organizing way. The distributed and bottom-up recommendation strategies guarantee the diversity and adaptivity of recommendations.

- Besides learners' learning profiles, the dynamic changes of learners are also obtained through LOs' self-organization behaviors. The environment perception module is applied to capture the changes within the recommendation environment. Once an LO is studied by learners or be influenced by other LOs, its attributes, such as extended metadata, immediately are updated. Such attribute changes transmit the real-time preference drift of learners. Hence, we can update the environment parameters such as the changes in learners' preferences and abilities, the changes in LOs' position, the clustering tendency, LOs' quality analysis, system entropy, etc. By analyzing the environment parameters, we can further obtain the predictive environment parameters.

The remainder of the paper is organized as follows: Section 2 describes some previous research related to our study. Section 3 provides details on learner and LO modeling. Section 4 introduces the self-organization recommendation strategy. Section 5 presents the experimental setup of the proposed recommendation approach. Section 6 includes the experimental results and our discussions. Finally, section 7 summarizes the characteristics of the proposed approach and introduces future work.

2. Related Works

In this section, we review relevant researches on the learner model, the LO model and recommendation methods related to improving the diversity and adaptability of recommendations.

2.1. Learner and LO models

Martins et al. [20] mentioned that the user model should be conducive to making the educational process more adaptive and capable of preparing learners for future professions. Some modeling methods are used to generate user models in adaptive hypermedia systems such as the overlay model and the uncertainty-based user model [21, 22]. In the application of the overlay model for recommender systems, besides some common characteristics such as learners' learning goals, age, gender, and education level, learning styles are also often used to describe learners' personalized preferences [23]. Many previous studies have considered accurately matching one or two of the above learning styles [4, 24]. Felder and Silverman's learning style and David Kolb's learning style (KLS) are the most widely used. In the e-learning environments, since learners' preferences and dynamic demands are subjective and multidimensional, more comprehensive learning styles and dynamic preferences should be combined together in CB recommender systems.

Some methods are used to build user models with the consideration of the imprecise and complex natures of human behaviors, for example Bayesian belief network, matching learning, fuzzy logic-based techniques, etc [25, 26]. Collaborative filtering approaches are mainly applied on predicting the preferences and processing concept drift [27, 28]. Si et al. [29] considered diversity features in check-in data for constructing adaptive recommendation method, especially variability and consecutiveness features of temporal factor. Those studies also pointed that popular windowing and instance weighing approaches for handling concept drift or dynamic changes are not the best, simply because in collaborative filtering, the rating matrix is high dimensional and extremely sparse. Most importantly, the need for large amounts of historical data is an unavoidable limitation in such model building methods.

In e-learning recommender systems, the LO model is also one of the most important inputs. Besides ontology and concept map [30, 31], metadata is often applied to the description of LOs [32]. Some metadata is extended for special educational recommendation systems. The metadata includes competence [33], reputation [34], etc. Some researches studied how to accurately create and update the domain model. For example, Kahraman et al. [35] focused on developing a domain model of adaptive hypermedia applications by redefining the storage layer of existing adaptive hypermedia systems. In [36], the course-based learning activities are fused into a fuzzy tree-structured learner profile to meet learners' requirements. The similarity measure is most based on learning activities and pedagogical issues. Until now, there is no research that models LOs as intelligent entities and uses LOs' interactions to make recommendations.

2.2. Recommendation Strategies

It is acknowledged that adaptive presentation and adaptive navigation should be achieved according to the real-time needs of the learners [14]. Currently, recommender systems obtain the changes in learner preferences by analyzing learners' behaviors or their ability tests, then adjust recommendation strategies to make appropriate recommendations [37, 38]. For example, DiNoia et al. [19] focused on modeling user propensity to select diverse items, where diversity is computed by means of content-based item attributes. Vargas et al. [39] also tried to detect the diversity of users' preferences by analyzing users' history profiles. Project et al. [40] proposed an adaptive recommender system by incorporating the learner model to the resource model through the combination of hierarchical and network-based approaches to represent the resource domain. Chen et al. [41] proposed a method for automatically constructing concepts maps for adaptive learning systems based on data mining techniques. Colace et al. [42] studied the adaptive hypermedia system by using the definition of methodologies which are able to manage user model, learning contents, tracking strategies, and adaptation model. Kahraman et al. [35] focused on redefining the storage layer of existing adaptive hypermedia reference models in order to create and update the domain. From the above researches, it is clear that the adaptive model is confined to the predefined learner model, and that the CB recommendation strategies are directly based on learner-LO similarity. Compared with the fast-changing in learners' preferences, the learner profile-based evaluations and recommendations often lag behind, which eventually decreases the adaptability of recommendations. All these considered, there is a need for further research on detecting the ever-changing information of both learners and LOs and incorporating the information into the recommender systems.

Moreover, the interest and curiosity of learners is often reduced due to a lack of diversity. In order to decrease excessive recommendations and increase adaptability, recommendation systems should have the ability to generate unexpected but attractive recommendations. One possible solution is the introduction of probability-based recommendation mechanisms. For example, Sheth et al. [43] applied the probability-based genetic algorithms to the information filtering. Yueh et al. [44] studied Markov's chain model-based meta-rules to help learners achieve effective web-based learning paths. Additionally, Bayesian Knowledge Tracing (BKT) is a common way of determining student knowledge of skills in adaptive educational systems and cognitive tutors [45]. The basic BKT is a Hidden Markov Model (HMM) that models student knowledge based on five parameters: prior, learn rate, forget, guess, and slip. Some other factors are introduced to improve the basic BKT. In [46], learners' affective states are combined with learners' knowledge ststes to optimize the knowledge tracing algorithm. Pavlic et al. [47] predicted learners' performance by using logistic regression on the difficulty of LOs and the time series-based studying results.

Currently, few attempts have been made to improve the quality of recommendations using influence propagation among individuals. Janssen et al. [48] provided recommendations for active learners by feeding back information on successful learning tracks from other learners. Koper et al. [49] showed that indirect social interaction helps learners achieve their required competence effectively, they concentrated on studying the influence of LO quality, the disturbance of the external environment, and the matching errors. Those studies focused solely on the influence propagation of individuals' behaviors, the individuals specifically refer to the learners. Until now, there is no research that takes LOs as entities which can transmit information. Actually, LOs can respond to the input signal of stimulus and spontaneously reach a steady state, then the recommendations can be achieved. The LOs operated by learners carry valuable information on learners' implicit demands, so the predictive and adaptive recommendations can be obtained by analyzing the inherent correlation and interactions among LOs.

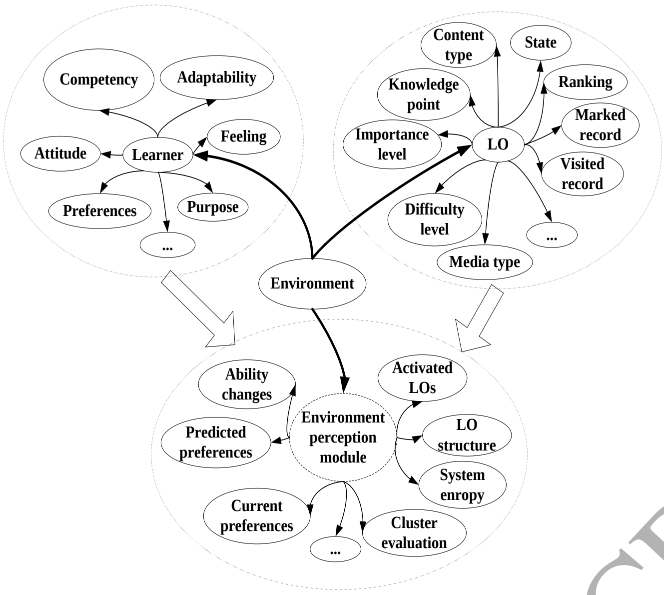


Figure 3: The ontology graph of LO, learner and environment

In this study, we incorporate self-organization theory into CB e-learning recommender systems. The LO-oriented recommendation mechanism is fused into the learner-oriented recommender system. This bottom-up and distributed approach is conducive to improving the diversity and adaptivity of recommendations.

3. Preliminaries

In this study, the e-learning environment is defined as the interactive place for learner and learning resource entities. Fig. 3 is a brief ontology graph of the LO, learner, and the environment. In this section, we introduce the LO model and learner model based on the ontology descriptions.

3.1. Learner Model

Each learner is represented as a two-tuple $<GU, TU, BU>$. A detailed description of each tuple is as follows.

(1) GU represents learning goals. Each learner's goals are composed of a concept set that the learner needs to learn.

(2) TU describes learning styles. Referring to other research [50], we design the elements of learning styles as: $TU = \{CL, MP, FP, PU, AT, FE, AC, DC, HP\}$. *Competency* ($CL$) includes a learner's current knowledge level and cognitive ability; *media preference* (MP) and *content preference* (FP) refer to learners' preferences for specific LOs; *purpose* (PU) describes learning expectations for learners' study, which refers to Bloom's taxonomy of educational objectives; *attitude* (AT) means learners' behaviors with regards to knowledge acquisition such as active, impulse and cautious types; *learning feeling* (LF) reflects learners' current learning experiences, such as being attracted, impatient or feeling difficult to follow. LF can be obtained from learners' marking behaviors and studying process; *adaptability* (AC) indicates the extent to which a learner is willing to accept the LOs with low low-matching value with

learners; *tolerance of repeated LOs* (DC) represents learners' tolerance for repeatedly recommended LOs; *preference priority* (HP) is a sequence of the above preferences for a certain learner. These metadata are represented as digits or digital sequences and can be updated according to learner profiles.

(3)BU represents learners' behaviors. $BU = \{bu_1, bu_2, \ldots, bu_n\}$. BU includes the possible operations of learners in e-learning platform. Although some sequential pattern mining algorithms are applied to extracting learners' learning activities [21], it is not wise to predefine or utilize constant activity patterns to model learners. Therefore, in this study, learners' behaviors are the basic and direct stimuli for the self-organization of LOs. For example, bu_1 means marking *Too hard*. If a learner marks an LO with *Too hard*, $bu_1 = 1$, otherwise, $bu_1 = 0$. bu_2 means marking *Too easy*; bu_3 means marking *Later* and bu_4 means marking *Ignore*. The other behaviors include learners' self-adjustment on preferences and the performances on tests.

3.2. LO Model

The number of n LOs are represented as a set $L = \{l_i | i \in [1, n]\}$. LO's attribute set is LA. For each l_i, its attribute set LA_{l_i} is represented as $LA_{l_i} = \{C_{l_i}, S_{l_i}, D_{l_i}, K_{l_i}, M_{l_i}, F_{l_i}, T_{l_i}, G_{l_i}, EC_{l_i}, EP_{l_i}, ES_{l_i}, EM_{l_i}, ET_{l_i}, EF_{l_i}\}$. The attributes are divided into BL and EL. The implications are described as follows:

(1) BL represents the basic attributes of LOs. Table 1 shows the implication of each element in BL. In it, S_{l_i} and D_{l_i} are initialized according to expert suggestions and previous learners' feedback. A larger value indicates higher importance or higher difficulty. These two attributes will be updated according to the LOs' visited time and visited frequency during the learning process. T_{l_i} is preset a suggested time based on teaching experiences and the average study time of learners.

(2) EL represents the extended metadata. EL includes LOs' attributes related to the learning process. The elements in EL are explained in Table 2.

Two weights, $W^s_{l_i}$ and $W^d_{l_i}$, are set to adjust $l_i^{'}s$ importance and difficulty.

$$W^s_{l_i} = ET_{l_i} / \sum_{i=1}^{k} \max_{0 \le j \le m} ET^j_{l_i} \tag{1}$$

To compute the weight $W^s_{l_i}$, we only consider the ***Same*** LOs. ***Same*** LOs refer to the LOs which belong to the same knowledge point, same media type, and same content type. If a learner visits an LO for m times, the longest ET_{l_i} is effective. In the above equation, k assumes the number of ***Same*** LOs, and $ET^j_{l_i}$ means the length of time when the l_i is visited by a leaner for the jth time.

EF_{l_i} is applied to determining a weight - $W^d_{l_i}$,

$$W^d_{l_i} = EF_{l_i} / \sum_{i=1}^{n} EF_{l_i} \tag{2}$$

Along with the learning process, the importance and difficulty of l_i are updated using $W^s_{l_i}$ and $W^d_{l_i}$. The equations for updating the importance and difficulty are added as follows:

$$C_{l_i} = C_{l_i} \times (1 + (W^s_{l_i} + 2 \times W^d_{l_i})/3) \tag{3}$$

$$D_{l_i} = D_{l_i} \times (1 + (2 \times W^s_{l_i} + W^d_{l_i})/3) \tag{4}$$

The importance and difficulty are normalized into [1, 5] again.

Table 1: Basic Metadata of l_i

Symbol	Implication	Description	Others
C_{l_i}	Knowledge point	$C_{l_i} \in [1, m]$	m knowledge points considered, C_{l_i} represents one or more knowledge points for specific knowledge module
S_{l_i}	Importance level	$S_{l_i} \in [1, 5]$	the importance level of l_i
D_{l_i}	Difficulty level	$D_{l_i} \in [1, 5]$	the difficulty level of l_i
K_{l_i}	Size	K_{l_i} is a digit	the unit is byte
M_{l_i}	Media attribute	$M_{l_i} \in [1, 6]$	1-Chart, 2-Animation, 3-Audio, 4-Video, 5-Office document, 6-Web file
F_{l_i}	Content attribute	$F_{l_i} \in [1, 5]$	1-Theory, 2-Explanatory, 3-Objective test, 4-Subjective test, 5-Example, 6-Module test
T_{l_i}	Suggested learning time	T_{l_i} is a digit	the unit is second
G_{l_i}	Prior constraint matrix	G_{l_i} is a matrix of $g_{l_i l_j}$, $i, j \in [1, n]$	$g_{l_i l_j}$ means the prior constraints between l_i and l_j, $g_{l_i l_j} = 1$, means l_i should be learned before l_j. 0 means no order constraints

4. LO Self-organization Based Recommendation Approach

In this section, the computation methods of similarity between entities are introduced first. Then the environment perception module is modeled. Based on the similarity computation and environment perception module, we further introduce the self-organization behaviors of LOs and the detailed recommendation strategy.

4.1. Similarity Computation

The similarity between LOs is critical for the information propagation in the LO self-organization process. The matching degree between learners and LOs is important to initialize the LO recommendations. The methods for computing similarity are listed in this section.

4.1.1. LO Similarity

Considering the differences between these attributes of LOs, we divide the LO attributes into two groups. In the first group $A1$, the numerical values of the attributes are of practical significance, and the differences for the attributes are of practical significance also. For l_i, $A1_{l_i} = \{S_{l_i}, D_{l_i}, K_{l_i}, T_{l_i}, EP_{l_i}, ES_{l_i}, ET_{l_i}, EF_{l_i}\}$. For example, if EP_{l_a}=5, EP_{l_b}=10, it means l_a has higher priority of being selected. The similarity of this group is calculated using the cosine similarity approach. $A1^j_{l_i}$ represents the jth attribute in $A1_{l_i}$,

Table 2: Extended Metadata of l_i

Symbol	Implication	Description	Others
EC_{l_i}	Current state	$EC_{l_i} \in \{1, 2, 3, 4\}$	1-Studied, 2-Un-studied, 3-Candidate, 4-Un-candidate
EP_{l_i}	Position ranking in current queue	$EP_{l_i} \in [1, n]$	n is the total number of LOs, the smallest number represents the top ranking
ES_{l_i}	Similarity ranking	$ES_{l_i} \in [1, n]$	The smallest number represents the highest similarity
EM_{l_i}	Marked label	$EM_{l_i} \in \{1, 2, 3, 4\}$	1 means Too hard, 2 means Too easy, 3 means Later, and 4 means Ignore
ET_{l_i}	Visited time	$ET_{l_i} = vt_{l_i}/K_{l_i}$	vt_{l_i} is the visited time of l_i
EF_{l_i}	Visited frequency	EF_{l_i} is a natural digit	EF_{l_i} records the visited times of l_i

the similarity between l_a and l_b $sim(l_a, l_b)^*$, is computed as follows.

$$sim(l_a, l_b)^* = \sum_{j=1}^{8} (A1^j_{l_a} * A1^j_{l_b}) / (\sqrt{\sum_{j=1}^{8} {A1^j_{l_a}}^2} * \sqrt{\sum_{j=1}^{8} {A1^j_{l_b}}^2}) \tag{5}$$

For the second group $A2$, the values and differences of these attributes have no practical significance. $A2_{l_i} = \{C_{l_i}, M_{l_i}, F_{l_i}, G_{l_i}, EC_{l_i}, EM_{l_i}\}$. To compute the similarity based on group $A2$, we only consider LOs that have the same attributes, so the Jaccard coefficient is applied for the similarity computation. Let $A2^j_{l_i}$ represent the jth attribute in $A2_{l_i}$, the similarity of l_a and l_b, $sim(l_a, l_b)^{**}$ is computed as follows.

$$sim(l_a, l_b)^{**} = \sum_{j=1}^{6} (A2^j_{l_a} \bigcap A2^j_{l_b}) / (A2^j_{l_a} \bigcup A2^j_{l_b}) \tag{6}$$

$sim(l_a, l_b)^*$ and $sim(l_a, l_b)^{**}$ are combined together to determine the similarity of two LOs.

$$sim(l_a, l_b) = \omega^* * norm(sim(l_a, l_b)^*) + \omega^{**} * norm(sim(l_a, l_b)^{**}) \tag{7}$$

ω^* and ω^{**} are two weights. In the previous definition of HP, it has been mentioned that each preference is assigned with a digital order in HP. The smaller the value, the higher the priority of the preference. The sum of LO preference orders in these two groups are computed to decide the weights ω^* and ω^{**}. $LA^j_{l_i}$ represents the jth attribute in LA_{l_i}. $Order_{HP}(LA^j_{l_i})$ is the order of $LA^j_{l_i}$ in HP. $\omega^* = 1/\sum_{LA^j_{l_i} \in A1} Order_{HP}(LA^j_{l_i})$, $\omega^{**} = 1/\sum_{LA^j_{l_i} \in A2} Order_{HP}(LA^j_{l_i})$. The weights are finally normalized.

4.1.2. Similarity Between LO and Learner

Only one learner is considered in this study, the attribute set of a learner U is described as: $UP = \{GU, CL, MP, LF, PU, AT, EP, AC, DC, HP\}$.

UP_i refers to the ith attribute in UP. The similarity between learner U and l_i consists of the following parts:

- $Hmatch^1$ measures the matching degree between learner's learning goals and $l_i^{'}s$ knowledge point.

$$Hmatch^1 = \begin{cases} 1 & \text{if} \quad C_{l_i} \in GU \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

- $Hmatch^2$ is the matching degree between learner U and the CL_{l_i}, MP_{l_i}, FP_{l_i}, PU_{l_i}, AT_{l_i} of l_i. The similarity computation refers to learning style surveys and association rules [51]. The related study is presented in our previous paper [50].

$$\begin{aligned} Hmatch^2 = 1/(norm(|PU - D_{l_i}| + |AT - D_{l_i}| + |CL - D_{l_i}| + \\ |CL - C_{l_i}|) + norm(|Order_{FP}(l_i) - Order^*_{FP}(l_i)|) + \\ norm(|Order_{MP}(l_i) - Order^*_{MP}(l_i)|)) \end{aligned} \tag{9}$$

The first part is obtained from association rules. The second part indicates the order of difference in the learner's attribute preference. For example, $Order_{FP}(l_i)$ is the actual position of l_i in the LO sequence FP among the same knowledge, and $Order^*_{FP}(l_i)$ is the expected sequence of l_i for FP.

- $Hmatch^3$ evaluates whether LOs can provide a smooth learning experience. There are three factors that are included. Being attracted refers to the condition when that the number of marked labels is small, learners perform well in module test and keep on moving to next concepts. Being impatient means the learner's marked labels focus on $EM_{l_i} \in \{2, 4\}$, and they frequently switch to other concepts. Feeling difficult means learners marked a lot on $EM_{l_i} \in \{1, 3\}$, and learners are not actively switching to new concepts. By quantitative assessments, $Hmatch^3$ is finally normalized to a digit in [1, 5].

- $Hmatch^4$ is evaluated by analyzing the low-matching degree between learners and LOs. AC is a percentage, which refers to the extent of to which a learner accepts mismatching LOs. The greater the value, the higher the acceptance. QM_{l_i} refers to the times that l_i is marked, QA_{l_i} refers to the times that l_i is accessed.

$$Hmatch^4 = \begin{cases} 0 & \text{if} \quad QM_{l_i}/QA_{l_i} \geq AC \\ 1 & \text{Otherwise} \end{cases}$$

- $Hmatch^5$ is evaluated by analyzing the number of repeated resources. PR_{l_i} is the times that l_i is repeatedly recommended. Similar to AC, DC is a percentage.

$$Hmatch^5 = \begin{cases} 0 & \text{if} \quad PR_{l_i} \geq \lfloor (1/DC) \rfloor \\ 1 & \text{Otherwise} \end{cases}$$

- $Hmatch^6$ aims to compute the differences between the expected order (HP) of attributes and their actual order in commendations.

  $Hmatch^6 = 1/\sum_{LA_{l_i}^j \in LA_{l_i}} |Order_{HP}(LA_{l_i}^j) - Order_{HP}^*(LA_{l_i}^j)|$,

The similarity between a leaner U and an LO-l_i is computed as follows.

$$sim(l_i, U) = \sum_{j=1}^{6} norm(Hmatch^j) \tag{10}$$

It needs to be noted that although the similarity of learner and LO is critical, the recommendations will also be updated based on the subsequent self-organization behaviors of LOs.

4.2. environment perception module

In this study, the recommender system is set as a three-tuple $<L, U, E>$. Here, L is the LO set, U is a learner, and E refers to the recommendation environment. It is assumed that all the entities are positioned in a two-dimensional environment using automatic cell machine theory. The entities around one active entity are considered as its neighbors. Each parameter in E reflects the specific state of LOs and learners, and the parameters are evaluated and updated within a certain time window TW. With n LOs considered, the main parameters of E are as follows:

- $\boldsymbol{EC}$ is a set, which consists of the LO's current state. $\boldsymbol{EC} = \{EC_{l_1}, EC_{l_2}, \ldots, EC_{l_n}\}$.

- $\boldsymbol{EP}$ is a set which reflects the current ranking of LOs in recommendations. $\boldsymbol{EP} = \{EP_{l_1}, EP_{l_2}, \ldots, EP_{l_n}\}$.

- $\boldsymbol{ES}$ is a set which reflects LOs' ranking based on the similarity between the learner and LOs. $\boldsymbol{ES} = \{ES_{l_1}, ES_{l_2}, \ldots, ES_{l_n}\}$.

- $\boldsymbol{EPV}$ records the variation of the position of the LOs in EP. $[t-1, t]$ is the time interval. $\boldsymbol{EPV} = (EP_{l_1}^t - EP_{l_1}^{t-1}), \ldots, (EP_{l_n}^t - EP_{l_n}^{t-1})$.

- $\boldsymbol{ESV}$ records the variations in the order of LOs in ES. $\boldsymbol{ESV} = (ES_{l_1}^t - ES_{l_1}^{t-1}), \ldots, (ES_{l_n}^t - ES_{l_n}^{t-1})$.

- $\boldsymbol{EPA}$ is an LO set which records the LOs' average ranking in TW. $\boldsymbol{EPA} = \{EPA_{l_1}, EPA_{l_2}, \ldots, EPA_{l_n}\}$. EPA_{l_i} refers to the average position of l_i.

- $\boldsymbol{CL}$ shows the change in ability of learners according to their visited LOs and test results. The ability evaluation mechanism is designed according to the pedagogical methods.

- $\boldsymbol{UPA}$ is the set of learners' preferences obtained by analyzing the frequency of accessing LOs with a certain attribute. $\boldsymbol{UPA} = \{LA_{AF}^1, \ldots, LA_{AF}^j, \ldots, LA_{AF}^m\}$, m is the number of LOs' attributes. $LA^j \in LA$. LA_{AF}^j represents the frequency of occurrence of the jth attribute in LA. $LA_{AF}^j = Count\_AF_{i\in[1,n]}(LA_{l_i}^j)$. If the LOs with a certain attribute have a highest access frequency, it means that learner tends to choose the LOs with such an attribute.

- $\boldsymbol{UPM}$ shows learner's preference set obtained by the number of times LOs are marked. $\boldsymbol{UPM} = \{LA^1_{MF}, \ldots, LA^j_{MF}, \ldots, LA^m_{MF}\}$, LA^j_{MF} represents the marked frequency of the jth attribute in LA. $LA^j_{MF} = Count\_MF_{i\in[1,n]}(LA^j_{l_i})$. If LOs are marked frequently by learners, it means learners have low willingness to study these LOs. This is because the marking behavior represent learners' attitude towards the quality of the target LOs, especially in the age of information overloaded [52].

- $\boldsymbol{SE}$ is the system entropy which reflects the stability of the environment, low entropy means the recommender system provides a stable LO structure. The equation for computing $\boldsymbol{SE}$ is given in section 6.3.1.

- $\boldsymbol{SC}$ is the stable cluster sets of LOs. With information propagation, LOs evolve from a relatively disorderly state to a relatively stable state. When the similarity of inter clusters is greater than a preset threshold, and system entropy decreases below a threshold, LO clusters are recorded. For example, $\boldsymbol{SC} = \{l_1, l_5\}, \{l_3, l_7\}, \ldots, \{l_2, l_4, l_8, l_{17}\}$.

- $\boldsymbol{CC}$ records the constant clusters of LOs. For each cluster in CC, LOs have fixed neighbor relationships during time TW. For example, $\boldsymbol{CC} = \{l_1, l_3, l_7\}, \ldots, \{l_2, l_5, l_8\}$, this means that l_1, l_3 and l_7 are neighbors during TW.

  $\boldsymbol{SC}$ and $\boldsymbol{CC}$ reflect the quality of the LO regions. When the learner shows preference drift, the LO sets with similar preferences in $\boldsymbol{SC}$ and the LOs that have a higher stability in $\boldsymbol{CC}$ will be immediately recommended.

E includes learners' explicit and predictive preferences changes, LOs' cluster state and LOs' quality. By analyzing the environment detection parameters - E, more information about the learner and LO is further mined and defined as the predictive environment (PE). PE has the following parameters:

- LPC. It is the current preference set obtained from learners' direct operations on LOs. $LPC = \{LPC_{LA^j} | j \in [1, m]\}$. Each attribute UP_i is ranked in LPC according to an evaluation of its preferences in both $\boldsymbol{UPA}$ and $\boldsymbol{UPM}$. The preference evaluation of LA^j is computed as $LPC_{LA^j} = (Order_{\boldsymbol{UPA}}(LA^j) + 2 \times Order_{\boldsymbol{UPM}}(LA^j))/3$. Learners' marking behaviors are given higher weights.

- LPF. It is the predictive preference set for learners. $LPF = \{LPF_{LA^j} | j \in [1, m]\}$, it is obtained by analyzing the LOs which are not directly operated by learners. The LOs which have the same attribute are extracted to compute the tendency. Three factors influence this perception parameter, the position changes in EPV, ESV and $EC3$. $EC3$ is an LO candidate cluster with the attribute $EC = 3$. $LPF_{LA^j} = \sum_{\omega\in\{EPV,ESV,EC3\}} Order_\omega(LA^j)$. $Order_\omega(LA^j)$ refers to the sum of the order of LA^j in ω set.

- LOP. It is an ordered LO set which records LO's potential to be selected. $LOP = \{LOP_{l_i} | i \in [1, n]\}$, $LOP_{l_i} = \sum_{\omega\in\{EP,ES,EPA\}} Order_\omega(l_i)$.

- $Environmentperceptionmodule$. It is the optimal LO clusters, $LC = \{lc_1, \ldots, lc_i, \ldots, lc_k\}$. The LOs in lc_j is a sub-set in LC. The LOs belong to the intersection of elements in both both $\boldsymbol{SC}$ and $\boldsymbol{CC}$. For example, $\{l_3, l_7\}$ belongs to $\boldsymbol{SC}$ and $\boldsymbol{CC}$, therefore, $\{l_3, l_7\}$ is an element in LOR.

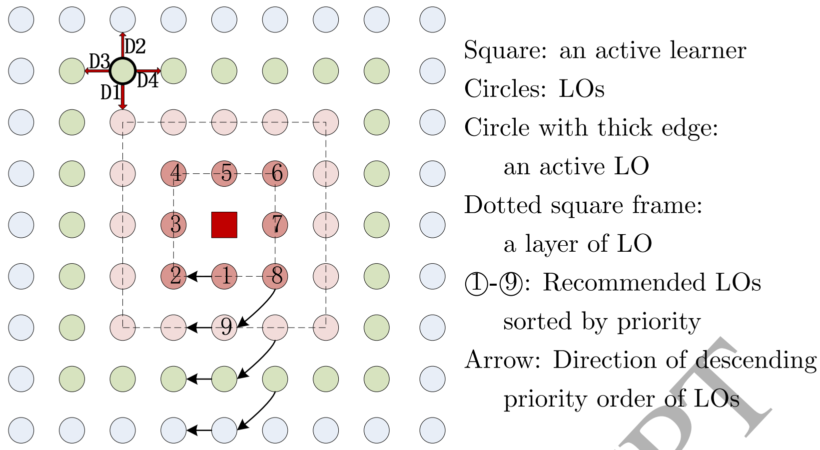


Figure 4: The diagram of the recommendation environment, and the demo explaining LOs' self-organization behaviors

4.3. LO Self-organization Based Recommendation Strategy

Fig. 4 is the diagram describing the recommendation environment, and it is also a demo explaining the self-organization behaviors of LOs (seeing 4.3.1). In such a two-dimensional environment, the central square indicates a learner, also, its position is fixed. LOs are represented as colorful circles. LOs' positions are initialized using cell automatic machine theory. The learner is surrounded by layers of LOs. The LOs which belong to the layer nearest to the learner have the highest priority when being selected. In the same layer, the priority is set from high to low clockwise.

In the study, recommendations are initialized by providing learners with high matched LOs. Then learner's learning behavior causes the changes of learning environment, thus the self-organization of resources is triggered. The LO that learner is studying is initialized as an active LO. At the beginning, the neighbors in same LOR with the active LO are influenced first. Their similarity to the active LO and their latent quality in LOP are the criteria to decide their behaviors. Under given conditions, these neighbors may move according to certain rules. The moving LOs transmit the information to their new neighbors in LOR. The recommendations are given when a stale LO state appears. Fig. 5 gives the overall framework of the Self. The three algorithms in this figure are given in the following part.

4.3.1. LO Self-organization Algorithm

LO behaviors can be abstracted as the movements towards some directions (forward, backward, left, and right) in a certain amplitude and probability. As for an active LO, its attributes, such as EC and EM, are updated immediately. Such changes in attribute transmit the real-time preference drift of learners. The environment perception module captures this preference drift and broadcasts this information to other LOs.

To describe the behavior rules of LOs, we simplify the dynamic preferences as two types, active and passive ones. Active type means that the LO is successfully studied by learners without any marking; passive type means the LO is marked with given labels. The active preferences will attract LOs move towards the learner (forward movement), on the contrary, passive preferences will cause LOs to move far away from the learner (backward movement). Algorithm 1 explains the self-organization behavior of

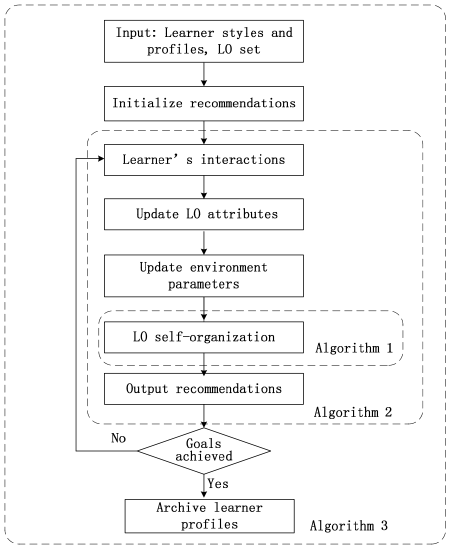


Figure 5: Overall framework of the Self

LOs. In this algorithm, the active dynamic preferences are taken as examples. Some related symbols are explained as follows:

- ST is set as the threshold of similarity between entities. For example, if l_i is active, and the similarity between l_i and l_j is larger than the threshold, l_j will be activated and move.
- QT is the threshold of LOP. It is used to determine whether the resource is in a good position in the current module.
- $p1$ and $p2$ are the moving probabilities, they refer to the probability of LOs moving forward or backward under the guidance of matching degree.
- Top_{LPC} refers to the subset of LPC, which includes the top 30% preferences in LPC set.
- Top_{LPF} refers to the subset of LPF, which includes the top 30% preferences in LPF set.

Generally, ST, QT, $p1$ and $p2$ are first set as empirical values. When implementing the experiments, we decide parameter combinations according to the simulation experiments. In which, the problem scale, entropy and clustering validation are the main criteria that determine these parameters [53].

In the Algorithm 1, the self-organization behaviors of LOs are mainly determined by their similarity to the active LO, as well as their ranking orders in candidate LOs. The current and predictive preferences of learners also play important roles in LOs' self-organization.

4.3.2. Recommendation Generating Algorithm

When a learner begins to study, the similarities between the learner and LOs are computed. The LOs with high matching degree with the learner are recommended first. Then, when the recommended LO, l_i, is operated by the learner, l_i exerts implicit behaviors on its neighbors. The algorithm of LOs' behaviors after being operated by learners is described as Algorithm 2.

Algorithm 1 LO self-organization algorithm

Input: An active LO

Output: LO list

1: **for all** $l_i \in LOR$, **do**

2: Compute the similarity between l_i and its neighbor l_j - $sim(l_i, l_j)$

3: If $sim(l_i, l_j) > ST$, $LOP_{l_j} \succ QT$, and $\exists BL_{l_j} \in Top_{LPC}$, l_j moves towards the inner layer with the probability of $p1$. Such movement means l_j has high matching similarity with l_i, l_j has high potential, and its basic attributes are consistent with learner's current preferences. In Fig. 4, this kind of movement is labeled as $D1$

4: If $sim(l_i, l_j) < 0.3 * ST$, $LOP_{l_j} \prec QT$, $\forall BL_{l_j} \notin (Top_{LPC} \bigcup Top_{LPF})$, l_j moves to the outer layer with the probability of $p2$. In Fig. 4, this kind of movement is labeled as $D2$

5: If $0.3 * ST \leq sim(l_i, l_j) \leq 0.6 * ST$ and $\exists BL_{l_j} \in Top_{LPC}$, or if $sim(l_i, l_j) \leq 0.3 * ST$ and $\exists BL_{l_j} \in Top_{LPF}$, l_j moves left along its layer anti-clockwise with the probability of $p1$. In Fig. 4, this kind of movement is labeled as $D3$

6: If $0.3 * ST \leq sim(l_i, l_j) \leq 0.6 * ST$ and $\forall BL_{l_j} \notin (Top_{LPC} \bigcap Top_{LPF})$, l_j moves right along its layer clockwise with the probability of $p2$. In Fig. 4, this kind of movement is labeled as $D4$

7: **end for**

8: If the moving frequency decreases below the threshold of the system entropy, jump to line 11. Otherwise, update E, PE, LA, and jump to line 9

9: Take one neighbor (towards the outward direction) as a new active LO (l_i), update its neighbors

10: Jump to line 1

11: Output the LO list according to their orders in EP

Algorithm 2 Interactive algorithm of learner and n LOs

Input: Learner's behaviors

Output: New recommendations

1: Update BU, E and PE

2: **for all** each l_i, $i \in [1, n]$ **do**

3: Update the extended attributes EL

4: Choose one neighbor of the activated (being operated) LO randomly as a new active LO, and execute its information propagation following LO self-organization algorithm (seeing Algorithm 1).

5: Update LO's neighbors

6: **end for**

7: Output new recommendations

Algorithm 3 LO self-organization based recommendation strategy

Input: Learners' goals and learning styles, LO set

Output: Archived learner profiles

1: Find out the candidate LOs according to $sim(l_i, U)$, $i \in [1, n]$.

2: Output the initialized LO list according to Top-N strategy.

3: Learner's behaviors trigger the self-organization behaviors of LOs (seeing Algorithm 2).

4: If learner's learning goals are achieved or learner interrupt the learning process intentionally, jump to Step 5, otherwise, jump to Step 3.

5: Output archived learner profiles.

4.3.3. LO self-organization based recommendation algorithm

The whole recommendation process of the proposed approach is shown as Algorithm 3. The whole recommendation process is bottom-up and distributed, every LO entity has the possibility to be recommended and the LO clusters are always dynamic and well-structured to predict the recommendations.

5. Experiment Setup

In this study, the experiments focus on formal settings in e-learning, that is, the environment offered by educational institutions (e.g., universities and schools) within a curriculum or syllabus framework [54]. The proposed method was applied to actual e-learning teaching practices. Given open educational resources, a recommender system is designed to recommend LOs for university students to achieve their learning goals.

5.1. Selection of Comparison Strategies

Some heuristic algorithms have been applied to e-learning recommender systems and have achieved good performances in adapting to rapid changes existing in learning environments [43]. The Genetic Algorithm (GA) is based on evolutionary principles such as natural selection and survival of the fittest, and it performs well in personalized e-learning recommendation. GA is selected as the comparison algorithm in this study. Considering the Markov chain approach has the random walking mechanism, we applied the Markov chain (MC) as one of the comparison strategies [35, 55].

In the absence of any e-learning system, learners organize the resources by themselves. The students are only provided with instructors' one-for-all general suggestions. The learning behaviors of learners depend mainly on their subjective inclinations and emotions. This approach is named as the traditional teaching method (Tra).

5.2. Experimental Data

Since there is no suitable public data, our experimental data is taken from two courses which were taught by the authors. One course is Visual Basic (VB) and the other one is C Programming (C). Both of them are compulsory courses for the participants. Digital resources are in the form of video, audio, PPT, Word documents, HTML pages, etc. The content of the digital resources includes pretest, theory, explanation, example, quiz, analysis, summary, module test, etc.

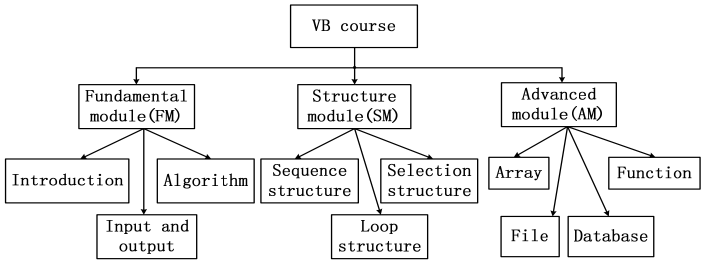


Figure 6: Course knowledge structure

Table 3: Information of participants and approaches for VB experiment

Class	Average age		Gender		Approach	
	Male	Female	Male	Female	FM	SM
WS151	19.8	20.3	25	7	Self	Self
WS152	20.5	19.8	23	8	GA	GA
WS153	19.6	18.7	24	8	MC	MC
WS154	19.6	18.9	24	7	Tra	Tra

5.2.1. VB Course and Participants

The content of VB course includes 10 chapters and we divide them into three modules, *Fundamental Module* (*FM*), *Structure Module* (*SM*), and *Advanced Module* (*AM*) (seeing Fig. 6). We apply *FM* and *SM* as two subcases in this study, that is, the periodic experimental data of *FM* and *SM* are analyzed. The quantities of the fine-granular LOs of these two modules are 175 and 482 respectively. LOs are annotated according expert experiences and learners' feedback.

The VB course was taught in 2016. Four classes of freshmen participated in the VB course experiment. The participants majored in Water Supply. The four classes are named WS151, WS152, WS153, and WS154. This classification of classes is according to the rankings of the entrance examination scores of participants, as well as their course scores in last semester, which ensures that these four groups have the approximate overall ability of academic performance. At the beginning of the course, all of the students took a survey, thus their learning goals and learning styles can be initialized.

These four different recommendation strategies (Self, GA, MC and Tra) are assigned to four classes respectively. In order to ensure the consistency and validity of the results, learners cannot repeat the learning process once the course has been finished. The detailed information of participants and approaches is shown in Table 3.

5.2.2. C Course and Participants

C is also a required course for freshmen. C includes 4 main modules: the *Fundamental module* (*Fund*) which includes *Operator*, *Expression*, *Input*, and *Output*; the *Structure module* (*Stru*), which includes *Sequence structure*, *Selection structure*, and *Loop structure*; the *Advanced module* (*Adva*), which includes *Array*, *Function*, and *Pointer*; and *Hard module* (*Hard*) which includes *Structure*,

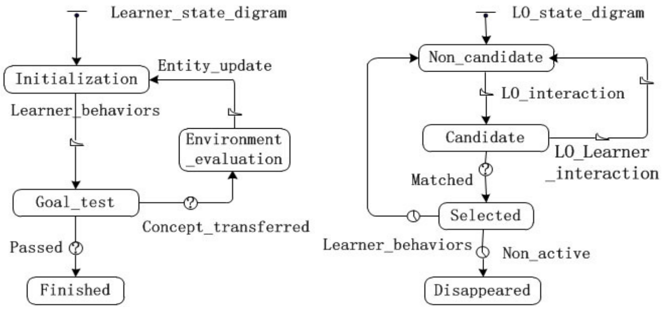


Figure 7: State transition diagrams of learner and LO entities

Union, *Bit operation*, and *Files*. The number of LOs with the smallest granularity is 2386. The C course was taught in 2017. Freshmen from three institutions participated in the experiment. These three institutions are Civil Engineering (CE) institution, Water Supply (WS) institution and Mechanical and Vehicle Engineering (ME) institution. Each institution consists of different classes, furthermore, each institution is taken as a group. The participant groups from these three institutions are marked with CE16, WS16 and MV16. The total number of students is 623.

The experiment setup of C course is shown in Table 4. All these three groups studied the $Fund$ and $Stru$ modules using the Tra method. $Fund$ and $Stru$ modules are labeled as $Part1$ which has 928 LOs. The other two modules, $Adva$ and $Hard$, are labeled as $Part2$ which has 1458 LOs. In these three groups, $Part2$ is assigned with three different recommendation strategies (Self, GA, and MC) respectively.

Table 4: Information of participants and approaches for C experiment

Groups	Average age		Gender		Approach			
	Male	Female	Male	Female	Part1		Part2	
CV16	18.2	18.5	218	78	Tra	Tra	Self	Self
WS16	18.7	19.4	134	46	Tra	Tra	GA	GA
MV16	19.1	18.3	115	32	Tra	Tra	MC	MC

5.3. Interaction Interface

Anylogic is applied to in order to implement the self-organization based recommendation system. Anylogic modeling enables us to describe and observe the behavior of entities [56]. Concurrent and independent behaviors of LOs will emerge, which promotes a global evolution of the recommender system. The main steps to simulating the LOs' self-organization are as follows: (1) construct an entity-based environment and place entities into the environment. (2) define the behavior and behavior rule of entities, and thus establish the behavior relationship between the LO and learner. (3) design state diagrams to describe the state change of entities. Fig. 7 shows the state transition diagrams of a learner and an LO respectively. (4) evaluate the stable state after LO self-organization process.

Most of the parameters are preset according to surveys and individual's demands, such as learning preferences and learning goals. Learners are permitted to adjust media and content preferences through

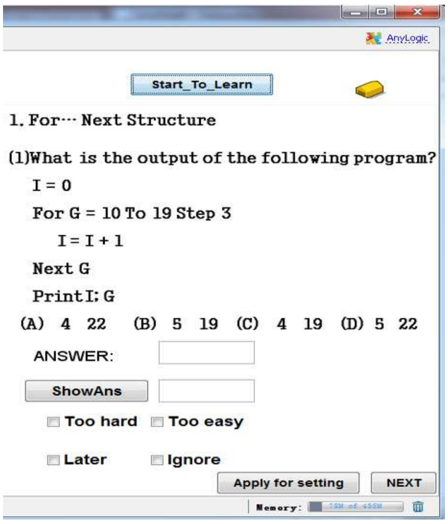


Figure 8: Interaction interface of learner activities

the interactive interface shown in Fig. 8. During learning process, a learner is encouraged to mark an LO using the check boxes, such as *Too hard*. If learners feel poor learning experience due to the current recommended resources, they can submit a request for fine-tuning some preferences by pressing the *Apply for setting* button. The request will be approved if it is consistent with the LPC and LPF parameters.

6. Results and Discussions

The evaluation protocols in this study mainly include three parts: effectiveness and efficiency, personalization and diversity, and evolution process. The evaluation strategies and the experiment results are discussed in this section.

6.1. Effectiveness and Efficiency

The effectiveness and efficiency of the recommendation strategies are evaluated by analyzing some basic data, such as resource utilization, learners' scores, and learning time [4, 57].

6.1.1. Score Evaluations

The average score ($Mean$) reflects the overall knowledge level of the group. The standard deviation ($Std.D$) reflects the difference of the group.

Table 5 lists the results of the VB experiment. The $Mean$ of the Self group is slightly higher than in the other groups (the Self group refers to the learners who use Self-based recommender system). The $Std.D$ of the Self group is correspondingly lower than the other groups. It indicates that the Self group can effectively improve the knowledge level of all the students. Table 6 lists the results of the C experiment. It can be seen that the $Mean$ in $Part2$ do not achieve better scores than $Part1$. This is mainly because the test in $Part2$ is more difficult than $Part1$. The CV16 group which applied the Self approach in $Part2$ obtains slightly higher $Mean$ results, also, its $Std.D$ is obviously lower than in $Part1$. It shows that compared with GA and MC, Self performs better in helping learners decreasing standard deviation and ensuring good effectiveness.

Table 5: Comparisons of Scores in VB Experiment

Module	Self		GA		MC		Tra	
	Mean	Std.D	Mean	Std.D	Mean	Std.D	Mean	Std.D
FM	76.3	10.3	70.9	14.9	74.0	13.1	72.3	16.4
SM	74.4	6.8	69.3	15.8	73.3	12.7	71.9	18.7

Table 6: Comparisons of Scores in C Experiment

Group	Part1		Part2	
	Mean	Std.D	Mean	Std.D
CV16	70.1	14.7	71.8	7.9
WS16	70.6	13.6	69.3	12.2
MV16	69.2	14.1	67.9	14.5

6.1.2. Learning Time

The average learning time is utilized to evaluate the efficiency of recommendation strategies. Generally, efficient recommender systems are expected to help learners accomplish their learning goals in a relatively short time. Table 7 lists the results of the VB experiment. In it, the TRL refers to the total length of suggested time of recommended LOs, and the $TimeP$ refers to the ratio between the length of time that learners actually spent and the TRL. As for the Tra, TRL means the total length suggested time of the LOs that learners selected first.

Through the comparisons of the TRL results, it is found that the TRL of the Self group is significantly lower than that of the other two algorithm groups. MC performs better than GA. In addition, from the analysis of $TimeP$, it is clear that the actual length of the spent time by the Self group is only slightly higher than the length of the suggested time of recommended LOs. Specifically, the $TimeP$ of SM in the Self group is 108%. This indicates the actual learning process of the Self group is highly consistent with learners' learning plans. As for the Tra group, the TRL shows that the students selected the LOs with the shortest time first, but the $Timep$ indicates that they spent longer learning time than their expectations.

Table 8 lists the results of the C experiment. Just as in Table 7, the values of $Timep$ in $Part1$ remain high, otherwise, the values of $Timep$ in $Part2$ are lower than the $Timep$ in $Part1$. It indicates

Table 7: Time of Recommended LOs and Learners' Actual Spend in VB Experiment

Module	Self		GA		MC		Tra	
	TRL	Timep	TRL	Timep	TRL	Timep	TRL	Timep
FM	103	126%	147	168%	132	155%	114	192%
SM	347	108%	367	137%	387	149%	322	161%

Table 8: Time of Recommended LOs and Learners' Actual Spend in C Experiment

Group	Part1		Part2	
	TRL	Timep	TRL	Timep
CV16	34	154%	219	92%
WS16	39	166%	193	126%
MV16	42	172%	204	148%

that, compared with the Tra, the other three approaches (Self, GA and MC) show better effectiveness in decreasing the learning time of learners. Further, by analyzing the results of $Timep$ of $Part2$, we are surprised to find that CV16 has a value of 92%, not only is it lower than 100%, it is also lower than 108% in VB experiment. Considering the fact that the number of LOs in $Part2$ is larger than in SM, it is can be concluded that Self can achieve higher performance in providing accurate and suitable recommendations when faced with more massive amount of LOs.

6.1.3. LO Utilization

LO utilization refers to the percentage of the recommended LOs out of the total candidate LOs. In recommender systems, the higher LO utilization means more LOs are recommended. Therefore, higher LO utilization is often combined with information overload. Fig. 9 displays the results of LO utilization in both the VB and C experiments. The LO utilization of the Tra group in C is computed by analyzing the three groups' learners' profiles of $Part1$.

The results of FM and SM show that the LO utilization in the Self group are slightly lower than the MA and GA groups. Especially, the LOs' quantity in the Self group becomes lower in SM. It can be seen that the Self can satisfy learners' goals with fewer LOs, and it has the ability to reduce the similar resources in recommendations. It can also be observed that the Tra group obtains the lowest LO utilizations in SM and C. The main reason is that learners often insist on selecting fewer resources when faced with a large number of candidate LOs because they want to complete the course as quickly as possible.

From Fig. 9 it can be seen that the results of Self, GA, MC groups in C are lower than those in VB. The Self has the lowest LO utilization in C, it shows that the recommendation approaches are better applied when there are more LOs.

6.2. Personalization and Diversity

In order to evaluate whether the proposed approach can facilitate the personalization and increase the diversity, we analyze the quality of the recommendations and learners' subjective feelings.

6.2.1. Objective Evaluations on Personalization

The evaluation of personalization includes fitness function, LOs' hit rate and learners' marking information.

(1) Fitness evaluation

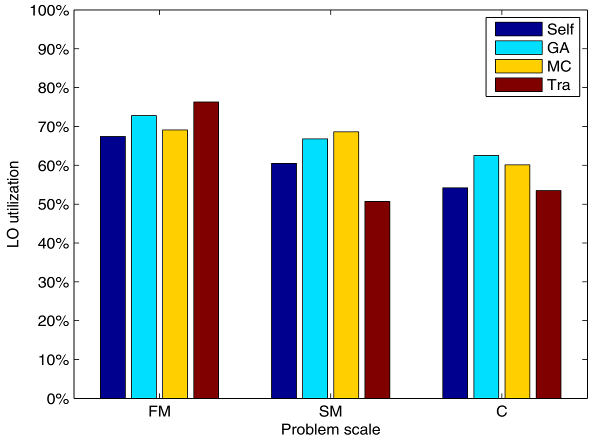


Figure 9: Comparisons of LO utilization

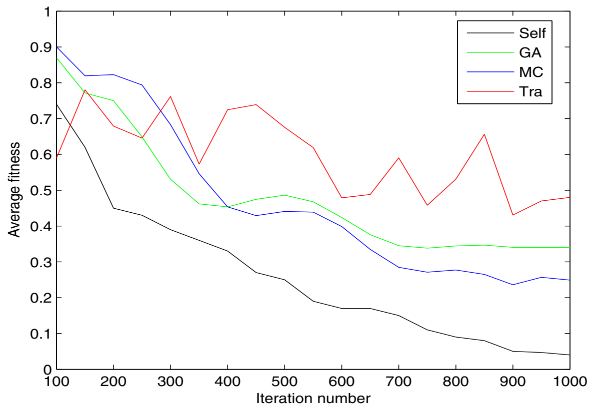


Figure 10: Fitness comparisons of penalty function

E-learning recommendation problem is multi-objective and multi-constraint, it can be simplified to a constraint satisfaction problem (CSP). In order to evaluate whether the recommended LOs conform to learners' demands, the fitness of the LO set the learner studied during the learning process is analyzed.

Fitness is a penalty function, it is computed by tracking the formation process of the recommendations. The fitness function is defined as:

$$F = norm(n/\sum_{i=1}^{n} sim(l_i, U)) \tag{11}$$

This evaluation is based on the recommendations for *Loop structure* in the VB experiment. Fig. 10 gives the fitness comparisons of different groups. In it, the x-axis is the iterative number. GA and MC use 1,000 as termination conditions. For the Self group, the transmission of information among LOs and learners is continuous. As such, the penalty function remains in a dynamic state. To make better comparison, the total learning time is mapped to a range of 0-1000 in the x-axis.

Based on Fig. 10, it is can be seen that the values of the Self and MC groups are far lower than that of GA and Tra groups. MC performs a distinct downward trend at the back part of their curves. However,

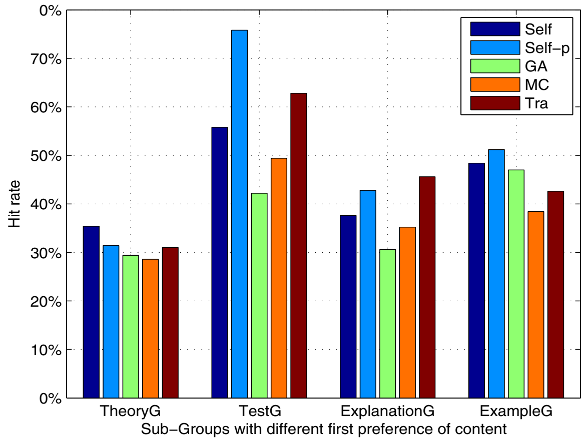


Figure 11: Hit rate comparisons for content preferences

the Self reaches a stable state earlier than the other three approaches. In addition, the curve of the Self group approach decreases sharply, and it does not fall into local optimum. The results show that the Self can provide suitable LOs to learners with a small margin of matching error. As for the Tra, the curve is fluctuating and it does not show a significant convergence trend. This indicates that the students in the Tra group have a lower ability than other groups in adapting to their dynamic demands.

(2) Hit rate evaluation

In order to test whether the recommended LOs can appropriately suit learners' learning styles well, the hit rate of recommendations is analyzed. Hit rate refers to the proportion of recommended results which are consistent with learners' top preferences in LPC. We take content preference as an example. All the participants in both VB and C experiments are divided into four sub-groups, that is, $TheoryG$, $TestG$, $ExplanationG$ and $ExampleG$. For example, students in the $TheoryG$ sub-group have the same first preference for the theory attribute. First, learners' current content preferences in LPC are taken as the references. The hit rates for different content attributes in these four sub-groups are recorded and shown in Fig. 11. Second, the hit rates for predictive preferences in the Self group are denoted as Self-p in Fig. 11. As for Self-p, the predictive preferences in LPF are used as the statistical data.

From the statistical data, it is found that, compared with the recommendations from the GA and MC groups, the recommendations from the Self show higher consistency with the preferences in LPC. Self-p has even higher hit rate than that in the Self. The Self approach is able to tailor to learners' demands and provide high accurate and predictive recommendations.

(3) Marking evaluation

Fig. 12 and Fig. 13 list the proportion of learners who marked LOs with $Ignore$ in these two experiments. LOs which are marked with $Ignore$ will not appear in the candidate queue again. It is interesting that in these two figures, the marking proportions of the Tra group are the highest. This means that learners are still willing to achieve their learning goals with smaller LOs. The marking proportions in the Self group are 3.4% and 4.1% in VB and C respectively. The values are lower than other groups. Combining the results in Fig. 9, we conclude that Self approach is able to recommend less but more

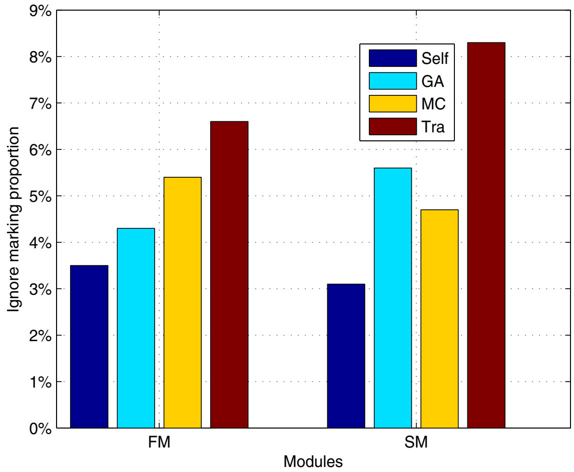


Figure 12: LO proportions marked with *Ignore* in VB experiment

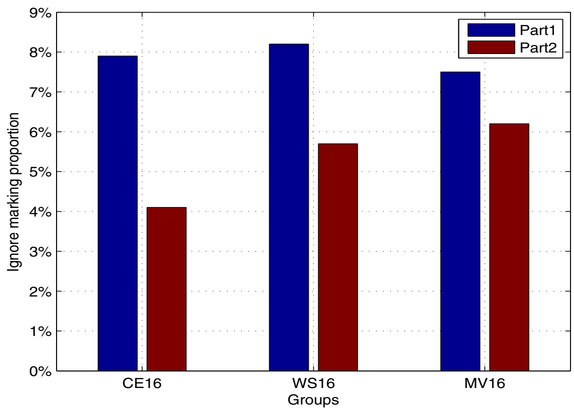


Figure 13: LO proportions marked with *Ignore* in C experiment

accurate LOs than other approaches.

Fig. 14 and Fig. 15 list the proportion of LOs which are marked with *Later* in the VB and C experiments. When learners feel that the recommended LO cannot match their preferences, they mark it by *Later* in order to temporarily avoid this LO. However, the LO still has the chance of being selected by the learner. Comparing Fig. 12 and Fig. 13, we can see that learners' *Later* marking proportions are higher than the *Ignore* proportions. This shows that learners are more cautious when making *Ignore*. Alternatively, when learners encounter an unsmooth learning experience, they do not easily give up the LOs that may be useful. It is also clear the marking proportions of Self, GA, and MC groups decrease distinctly. Recommender systems are effective to ensure learners' learning experiences. Learners give lower marking proportions when using the Self approach due to their high recognition for the recommendations.

6.2.2. *Objective Evaluation on Diversity*

(1) Diversity evaluation

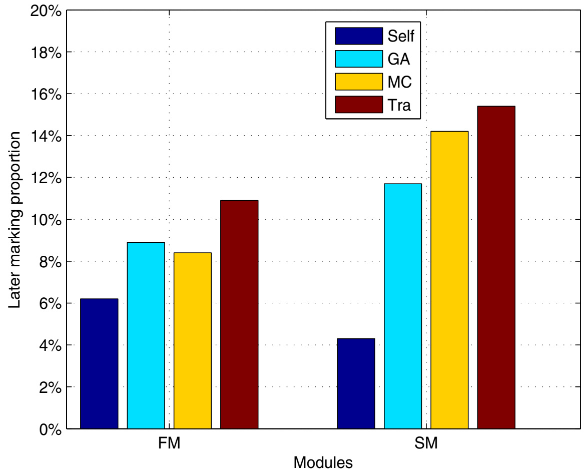


Figure 14: LO proportions marked with *Later* in VB experiment

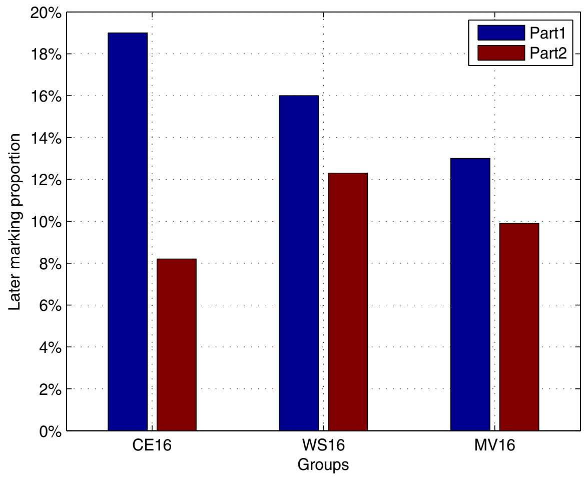


Figure 15: LO proportions marked with *Later* in C experiment

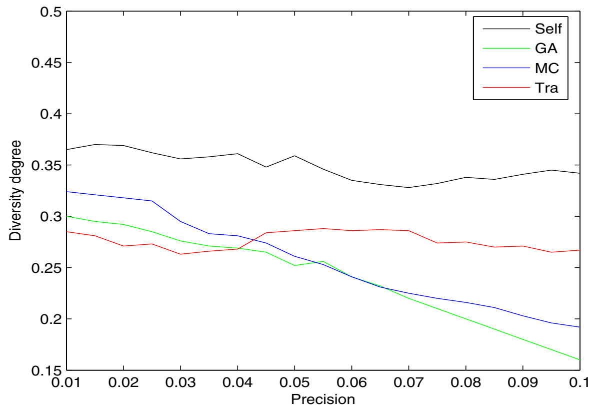


Figure 16: Comparisons of accuracy and diversity

The method for measuring the diversity in Top-N recommendation is applied here [58]. The diversity of recommendations is calculated according to whether the LO distribution in recommendations is balanced. Since the item characteristic-based diversity function can be regarded as a complementary for similarity measurement [59], the non-centralized distribution of attributes can be treated an expression of diversity.

$$DI(R) = (1/(|R| * (|R| - 1))) * \sum_{i \in R} \sum_{j \in R, j \neq i} div(l_i, l_j) \tag{12}$$

R is the LO set of recommendations for a specific learner and its size is $|R|$. We compute $div(l_i, l_j)$ as the complement of $sim(l_i, l_j)$, where the similarity between l_i and l_j is computed with the consideration of LOs' basic attributes. These attributes include the content attribute, media attribute, difficulty, and the importance level. $sim(l_i, l_j)$ is given referring to equation (7).

Generally, improving the diversity is often accompanied with a decrease in accuracy [60]. In this study, the analysis of diversity is also based on accuracy. The fitness values shown in Fig. 10 are first converted to a reciprocal ($1/F$) and then they are mapped as the precision of the recommendations. Then the relationship between accuracy and diversity is recorded in Fig. 16. It is noticed that the diversity of recommendations from the Self group does not change a lot. In other words, it still maintains a higher diversity value compared to the MC and GA. The high adaptability and the bottom-up recommendation strategy ensure that the diversity of recommendations is maintained with non-significant decreases in precision.

(2) Novelty evaluation

Inspired by the popularity-based novelty method which measures the unexpectedness of an object relative to its global popularity [59], the novelty of the recommendations is evaluated with the percentage of long-tail items among the recommendations across all users [61, 62]. In this study, taking the similarity of learners and LOs as the basis for matching, we consider 40% of low matching items as long-tail items. The statistics are about the percentage of LOs being recommended from the long-tail LO sets. The long-tail LO sets are grouped based on different attributes, that is, importance, difficulty, media, and content.

The statistical results of novelty proportions in VB experiment are shown in Fig. 17. We focus on

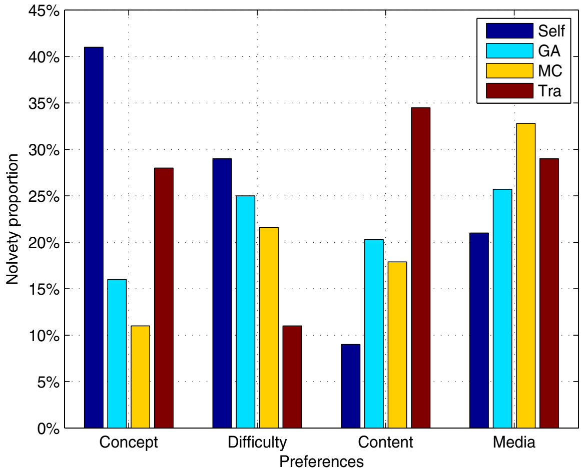


Figure 17: Novelty degree comparison in VB experiment

the results of *Concept* first. Both the Self and Tra groups have higher proportions than the other two groups, which indicates that learners are positive about considering the other concepts. In addition, the Self provides possibility of triggering such expectation. As for *Difficulty*, learners in the Tra group like to stick to their difficult preferences. However, the Self recommends more LOs with different levels of difficulty. This owes to the environment perception module with it learners' ability changes can be quickly captured and predicted, hence, the recommendations are more adaptive. It is unexpected that learners in the Tra group like to learn LOs with different attributes in *Content* and *Media*. Maybe it is because learners feel interesting in trying other attributes in *Content* and *Media*. However, the Self shows a lower proportion of these two attributes. The results of the C experiment are shown in Fig. 18 which show conclusions similar to the VB experiment.

The marking evaluations being considered, the Self group marked little for the recommendations. It can be seen that although learners in the Tra group show a preference for high novelty in *Content* and *Media*, they are more likely to be attracted by the novelty of *Content* and *Difficulty*.

6.2.3. Subjective Evaluation

When the learners finished the learning process, they were invited to complete a questionnaire designed to evaluate the learners' learning experience. The constituents and the results of the questionnaire are listed in Table 9 and Table 10. The ratings of each question range from 1 to 5 (1-Very unsatisfied, 2-Unsatisfied, 3-Fair, 4-Satisfied, 5-Very satisfied). For example, if the learner rates 5 on *Difficulty*, it means he is very satisfied with the difficulty of the recommendations.

The questionnaire results in the VB experiment are shown in Table 9. The first part of the questionnaire is *Personal satisfaction evaluation*, which aims to obtain learners' evaluations on the difficulty, media type, content sequence, and learning time. The Tra group rated high scores in the first three terms, which reflects that learners are satisfied with their selections. As for *Time*, the Self performs better than the GA and MC, while the Tra group received the lowest ratings. The main reason is that the first three approaches are capable of balancing learners' diverse demands with the massive LOs. The accuracy of

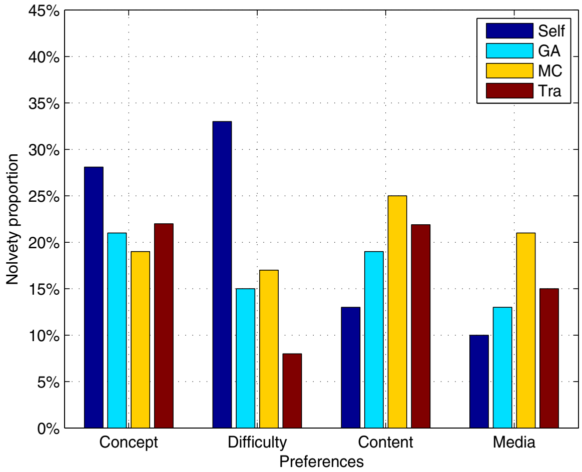


Figure 18: Novelty degree comparison in C experiment

Table 9: Evaluations of learners' experiences in VB experiment

Content	Items	Self	GA	MC	Tra
Personal realization evaluation	Difficulty	3.9	3.4	3.5	4.1
	Media	3.1	3.3	3.3	4.5
	Content	4.3	3.7	3.7	4.2
	Time	4.5	4.2	4.2	3.8
Flow experience evaluation	Control	3.8	3.7	3.5	3.1
	Attention focus	4.5	4.2	4.1	3.3
	Curiosity	4.2	3.6	3.9	2.9
	Intrinsic interest	3.5	3.2	3.5	2.8
Enhancing learning evaluation	Interactive flexibility	4.6	3.4	3.9	4.0
	Weakness strengthen ability	4.1	3.8	3.8	3.7
Overall satisfaction		4.5	4.3	4.2	3.9

the proposed approach is proven by the results.

The second part evaluates learner's learning feelings. We apply the four dimensions test of Trevino and Webster which includes control, attraction focus, curiosity, and intrinsic interest [63]. The Self group receives noticeably higher scores in flow experience, which indicates that the learners were immersed and engrossed in the learning process. The Self group also significantly outperforms the other groups in *Attention focus* and *Curiosity*, which is attributed to the randomness and probability mechanisms included in the self-organization approach. The diversity of the proposed approach has been proven by this part of the questionnaires.

The third part evaluates the dynamic interactive ability of the recommendation systems by analyzing

Table 10: Evaluations of learners' experiences in C experiment

Content	Items	Self	GA	MC	Tra
Personal realization evaluation	Difficulty	4.3	4.2	3.9	4.3
	Media	3.0	3.2	3.6	4.1
	Content	3.2	3.3	3.7	4.0
	Time	4.8	4.3	4.4	3.5
Flow experience evaluation	Control	4.3	3.7	3.8	3.4
	Attention focus	4.0	3.9	4.0	3.0
	Curiosity	3.0	2.7	2.2	2.5
	Intrinsic interest	3.9	3.0	3.1	2.7
Enhancing learning evaluation	Interactive flexibility	4.4	3.0	3.3	3.2
	Weakness strengthen ability	4.3	3.8	3.3	3.2
Overall satisfaction		4.2	3.6	3.8	3.3

whether the recommended LOs are relevant to the learners' knowledge level, especially when it comes to strengthening the weak abilities. As a whole, the Self received higher evaluations. The Self performs well in interactive flexibility, and its recommendations are consistent with learners' domain weaknesses. Learners' satisfaction also denotes their acceptance of the predictive and adaptive recommendations. Of course, although the interactive information is generated in a self-organized way, the environment perception module plays an important role in capturing and predicting the real time demands of learners in the Self group. The adaptability of the proposed approach has been well proved.

Finally, the learners rated the recommendation systems with $Overall\ satisfaction$. The ratings show that the Self is greatly approved by the learners in VB experiment.

Table 10 shows the results of the C experiment. We first focus on $Overall\ satisfaction$. Although learners highly rated the Self approach, it is found that learners in the C experiment tend to give low ratings. Under this situation, learners in the Self group also rated higher scores on the $Difficulty$, $Time$, $Control$, $Intrinsic\ interest$ and $Enhancing\ learning\ flexibility$. This proves learners' high recognition for the Self.

6.3. Evolution Performance Evaluation

To evaluate the evolution performance of the proposed approach, the entropy and self-organization process are evaluated through the analysis of entities.

6.3.1. Entropy Evaluation

In recommender systems, lower entropy means the frequency and magnitude of information exchange decreases, and the recommended LO's sequence tends to be stable. The equation of entropy for the Self

Table 11: Computation performance comparisons

	Self		GA		MC	
	FM	SM	FM	SM	FM	SM
Ave time(s)	0.0717	0.073	2.15	4.69	2.27	3.98
Ave entropy	0.34	0.31	0.57	0.75	0.64	0.69

- SE is listed as follows:

$$SE = -\sum_{i=1}^{n}(\sum_{j=1}^{m} log\Delta P_{l_i,l_j}) \tag{13}$$

Where, $\Delta P_{l_i,l_j} = |P^t_{l_i,l_j} - P^{t-1}_{l_i,l_j}|$, $P^t_{l_i,l_j}$ is the similarity between l_i and l_j of time t. n is the number of LOs, and m is the number of l_i's neighbors which belong to SC.

For the GA, entropy can be computed by each gene's selection probability. For the MC, entropy is related to the transfer probability of states.

When the recommended LOs reach a stable state for the first time, the average computation time (*Ave time*) and average entropy (*Ave entropy*) are recorded. The results of the VB experiment are shown in Table 11. In terms of *Ave time*, it is noticed that the value of the Self is faster than that of the other algorithms. Moreover, the *Ave time* of GA and MC groups increases when faced with large scale problems; however, the Self does not show large differences. This demonstrates that the Self approach has obvious advantages in large scale problems. The lower average entropy of the Self approach shows that it can reach a stable state more quickly.

In order to conduct further analysis of the Self, the system entropy of the Self approach is calculated at equal time interval. Four concepts from FM, SM, $Part1$, and $Part2$ are selected to make the entropy evaluations. They are *Input and output* in FM, *Loop structure* in SM, *Sequence structure* in $Part1$ and *Array* in $Part2$. The quantities of these four numbers are 82, 185, 274 and 332, respectively. We aim to evaluate the entropies of different problem scales. The results are listed in Fig. 19. The entropies of all these concepts show a clear downward trend. This is because that the information transmission is very frequent at the beginning of the learning process. When self-organization behaviors continue, the frequency of information transmission decreases. This implies that the positions of LOs are relatively fixed. In addition, with larger number of LOs, the entropy shows a distinct downward trend. This also indicates that the Self approach has the ability to adapt to larger scale problems. The adaptability and quick performances of the self-organization based strategy are highly proven by the entropy evaluations.

6.3.2. Self-organization evolution evaluation

The self-organization process is evaluated through variations in LOs' ranking. When implementing the self-organization recommendation systems, the learner is represented as a blue square, and is located in the central part of the environment. The candidate LOs are designated by colorful dots, randomly placed in the environment. The higher the matching degree, the more similar the LO's color becomes to the learner's color.

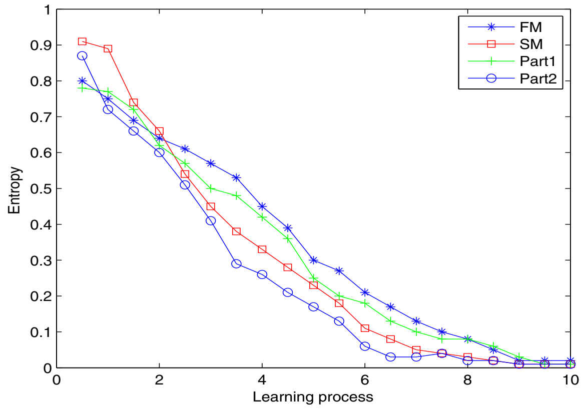
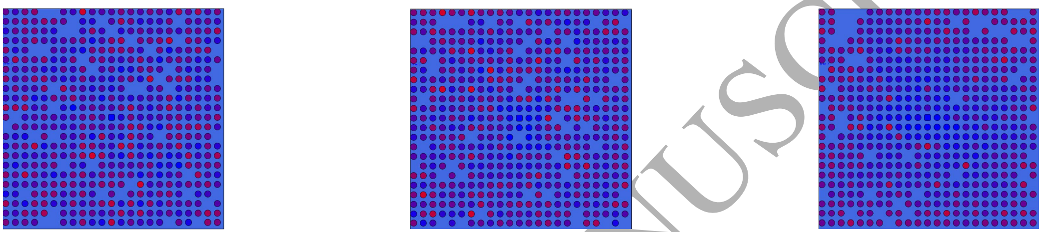


Figure 19: Changing tendency of the entropy of the Self



(a) Initial state (b) Intermediate state (c) Stable state

Figure 20: Self-organization process of the SM module

Fig. 20 shows the position variations in SM. In this figure, (a) is the initial state of the LOs. LOs are randomly scattered around the learner. (b) is the intermediate state of the self-organization based recommendation process. In it, the learner is surrounded by an increasing number of LOs with high similarity. (c) is the periodically stable state of the recommendation. In it, learners are surrounded by a high density of LOs with similar colors. The color changing tendency in Fig. 20(c) indicates that most suitable LOs are recommended according to the learners' demands, and that highly personalized recommendations have been achieved. However, it should be noted that some LOs with different colors surround the learner in Fig. 20(c), which indicates the diversity of the recommendations.

7. Conclusions and Future Work

The importance of improving the adaptability and diversity of recommender systems has strongly emerged. The fast-changing characteristics of the e-learning environment show a higher demand for adaptability and diversity than in other fields. In this paper, we propose an LO's self-organization based recommendation approach in e-learning. The LO-oriented recommender mechanism is combined with the learner-oriented CB recommender system. LOs are modeled as intelligent entities, and related metadata are extended to describe the LO's state. To ensure the intelligent recommendations, we propose a bottom-up and distributed self-organization recommendation strategy. With the stimuli of the learner's behaviors, LOs interacts with other LOs in an autonomous way. The positive information carried by LO-LO relationships is a critical criterion for LOs' behaviors. Moreover, the environment perception module is designed to make adaptive and predictive recommendations by analyzing both learners' learning activities and LOs' self-organization behavior. All of the experimental results indicate that the proposed approach can increase the possibility of diversity and adaptability in terms of little cost of precision. The definitive characteristics of the self-organization recommendation approach can be summarized as follows: excellent personalized performance, degradation of excessive recommendations, improvement of diversity, and good adaptability performance.

In future work, more case studies will be used to verify the effectiveness of the proposed approach, especially in massive open online courses (MOOCs). Furthermore, we will study the recommender systems based on learners' self-organization behaviors, Finally, we will devote to finding a way to combine LO self-organization and learner self-organization to provide personalized recommendations is also a key issue.

Acknowledgements

The authors wish to acknowledge the support of the National Natural Science Foundation of China (Project No. 61370137), the National 973 Project of China (No. 2012CB720702), Beijing emergency project(No, Z171100004417031), the Ministry of EducationChina Mobile Research Foundation Project (2016/2-7), and Beijing University of Civil Engineering and Architecture Science Research Foundation (No.00331614045).

References

References

[1] H. Drachsler, K. Verbert, O. C. Santos, N. Manouselis, Panorama of recommender systems to support learning, in: Recommender systems handbook, Springer, 2015, pp. 421–451.

[2] J. Lu, D. Wu, M. Mao, W. Wei, Z. Guangquan, Recommender system application developments: a survey, Decision Support Systems 74 (2015) 12–32.

[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, Knowledge and Data Engineering, IEEE Transactions on 17 (6) (2005) 734–749.

[4] O. Bourkoukou, E. El Bachari, E-learning personalization based on collaborative filtering and learner's preference, Journal of Engineering Science and Technology 11 (11) (2016) 1565–1581.

[5] K. Niu, Z. Niu, X. Zhao, C. Wang, K. Kang, M. Ye, A Coupled User Clustering Algorithm for Web-based Learning Systems, Proceedings of the 9th International Conference on Educational Data Mining (2016) 175–182.

[6] J. K. Tarus, Z. Niu, A. Yousif, A hybrid recommender system for e-learning based on context awareness and sequential pattern mining, Soft Computing 72 (2017) 37–48.

[7] M. Mao, J. Lu, G. Zhang, J. Zhang, Multirelational Social Recommendations via Multigraph Ranking, IEEE Transactions on Cybernetics 47 (12) (2017) 4049–4061. doi:10.1109/TCYB.2016.2595620.

[8] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, in: Recommender systems handbook, Springer, 2011, pp. 73–105.

[9] L. De-Marcos, R. Barchino, J.-J. Martínez, J.-A. Gutiérrez, J.-R. Hilera, Competency-based intelligent curriculum sequencing: comparing two evolutionary approaches, in: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 03, IEEE Computer Society, 2008, pp. 339–342.

[10] M. J. Pazzani, D. Billsus, Content-based recommendation systems, in: The adaptive web, Springer, 2007, pp. 325–341.

[11] T. Lee, H. P. In, An adaptive algorithm to recommend favorable digital music, International Journal of Multimedia and Ubiquitous Engineering 8 (6) (2013) 87–96. doi:10.14257/ijmue.2013.8.6.09.

[12] T. Quiroz, O. M. Salazar, D. A. Ovalle, Adaptable and adaptive human-computer interface to recommend learning objects from repositories, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9753, 2016, pp. 539–549. doi:10.1007/978-3-319-39483-1_49.

[13] P. Brusilovsky, Adaptive navigation support in educational hypermedia, in: Human-Computer Interaction: INTERACT 97, Springer, 2013, pp. 278–285.

[14] C. Wilson, B. Scott, Adaptive systems in education: a review and conceptual unification, International Journal of Information and Learning Technology 34 (1) (2017) 2–19. doi:10.1108/IJILT-09-2016-0040.

[15] J. Lu, Personalized e-learning material recommender system, in: Proceedings of the 2nd International Conference on Information Technology for Application (ICITA 2004), Citeseer, 2004, pp. 374–379.

[16] P. A. Schein A I, U. L. H, Methods and metrics for cold-start recommendations, in: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002, pp. 253–260.

[17] S. Vargas, P. Castells, Improving sales diversity by recommending users to items, Proceedings of the 8th ACM Conference on Recommender Systems (2014) 145–152doi:10.1145/2645710.2645744.

[18] P. Adamopoulos, A. Tuzhilin, On Unexpectedness in Recommender Systems : Or How to Better Expect the Unexpected, ACM Transactions on Intelligent Systems Technology 5 (4) (2015) 54:1—-54:32. doi:10.1145/2559952.

[19] T. Di Noia, J. Rosati, P. Tomeo, E. D. Sciascio, Adaptive multi-attribute diversity for recommender systems, Information Sciences 382-383 (2017) 234–253. doi:10.1016/j.ins.2016.11.015.

[20] C. Martins, L. Faria, C. V. De Carvalho, E. Carrapatoso, User modeling in adaptive hypermedia educational systems, Educational Technology & Society 11 (1) (2008) 194–207.

[21] P. Brusilovsky, E. Millán, User models for adaptive hypermedia and adaptive educational systems, in: The adaptive web, Springer-Verlag, 2007, pp. 3–53.

[22] E. Frias-Martinez, G. Magoulas, S. Chen, R. Macredie, Recent soft computing approaches to user modeling in adaptive hypermedia, Adaptive Hypermedia and Adapotive Web-Based Systems, Proceedings 3137 (2004) 104–114.

[23] R. I. Chang, H. H. Yu, F. L. Chun, Survey of learning experiences and influence of learning style preferences on user intentions regarding moocs, British Journal of Educational Technology 46 (3) (2015) 528–541.

[24] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, E-learning personalization based on hybrid recommendation strategy and learning style identification, Computers & Education 56 (3) (2011) 885–899.

[25] R. Stathacopoulou, G. D. Magoulas, M. Grigoriadou, M. Samarakou, Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis, Information Sciences 170 (2-4) (2005) 273–307. doi:10.1016/j.ins.2004.02.026.

[26] P. Pu, L. Chen, R. Hu, Evaluating recommender systems from the user's perspective: Survey of the state of the art, User Modeling and User-Adapted Interaction 22 (4-5) (2012) 317–355. doi:10.1007/s11257-011-9115-7.

[27] Y. Koren, Collaborative filtering with temporal dynamics, Knowledge discovery and data mining {KDD} (2009) 447–456doi:10.1145/1557019.1557072.

[28] J. Gama, I. Žliobait, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Computing Surveys 46 (4) (2014) 1–37. arXiv:1010.4784, doi:10.1145/2523813.

[29] Y. Si, F. Zhang, W. Liu, CTF-ARA: An adaptive method for POI recommendation based on check-in and temporal features, Knowledge-Based Systems 128 (2017) 59–70. doi:10.1016/j.knosys.2017.04.013.

[30] T. von der Heidt, Concept maps for assessing change in learning: a study of undergraduate business students in first-year marketing in china, Assessment & Evaluation in Higher Education 40 (2) (2015) 286–308.

[31] H.-S. Chung, J.-M. Kim, Ontology design for creating adaptive learning path in e-learning environment, in: Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. 1, 2012, pp. 585–588.

[32] J. F. Sowa, Ontology, metadata, and semiotics, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 1867, 2000, pp. 55–81. doi:10.1007/10722280_5.

[33] D. G. Sampson, Competence-related metadata for educational resources that support lifelong competence development programmes., Educational Technology & Society 12 (4) (2009) 149–159.

[34] T. Kerkiri, A. Manitsaris, A. Mavridou, Reputation metadata for recommending personalized e-learning resources, in: Semantic Media Adaptation and Personalization, Second International Workshop on, IEEE, 2007, pp. 110–115.

[35] H. T. Kahraman, S. Sagiroglu, I. Colak, A novel model for web-based adaptive educational hypermedia systems: SAHM (supervised adaptive hypermedia model), Computer Applications in Engineering Education 21 (1) (2013) 60–74. doi:10.1002/cae.20451.

[36] D. Wu, J. Lu, G. Zhang, A Fuzzy Tree Matching-Based Personalized E-Learning Recommender System, IEEE Transactions on Fuzzy Systems 23 (6) (2015) 2412–2426. doi:10.1109/TFUZZ.2015.2426201.

[37] P. Karampiperis, D. Sampson, Adaptive learning resources sequencing in educational hypermedia systems, Educational Technology & Society 8 (4) (2005) 128–147.

[38] B. Vesin, A. Klašnja-Milićević, M. Ivanović, Z. Budimac, Applying recommender systems and adaptive hypermedia for e-learning personalization, Computing and Informatics 32 (3) (2013) 629–659.

[39] S. Vargas, P. Castells, Exploiting the Diversity of User Preferences for Recommendation, Proceedings of the 10th Conference on Open Research Areas in Information Retrieval (2013) 129–136.

[40] M. T. Project, S. I. Report, Adaptive recommendation system for MOOC, Indian Institute of Technology (2013) 1–40.

[41] S.-M. Chen, P.-J. Sue, Constructing concept maps for adaptive learning systems based on data mining techniques, Expert Systems with Applications 40 (7) (2013) 2746–2755.

[42] F. Colace, M. De Santo, L. Greco, E-learning and personalized learning path: A proposal based on the adaptive educational hypermedia system, International Journal of Emerging Technologies in Learning 9 (2) (2014) 9–16.

[43] B. Sheth, P. Maes, Evolving agents for personalized information filtering, in: Artificial Intelligence for Applications, 1993. Proceedings., Ninth Conference on, IEEE, 1993, pp. 345–352.

[44] H. Yueh-Min, H. Tien-Chi, K.-T. Wang, W.-Y. Hwang, A markov-based recommendation model for exploring the transfer of learning on the web, Journal of Educational Technology & Society 12 (2) (2009) 144–162.

[45] M. Falakmasir, Z. Pardos, G. J. Gordon, P. Brusilovsky, A Spectral Learning Approach to Knowledge Tracing, in: Proceedings of the 6th International Conference on Educational Data Mining, 2013, pp. 28–34.

[46] S. E. Schultz, I. Arroyo, Tracing Knowledge and Engagement in Parallel in an Intelligent Tutoring System, Proceedings of the 7th International Conference on Educational Data Mining (EDM) 47 (3) (2014) 312–315.

[47] P. Pavlik, H. Cen, K. Koedinger, Performance Factors Analysis–A New Alternative to Knowledge Tracing, Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling (2009) 531–538doi:10.3233/978-1-60750-028-5-531.

[48] J. Janssen, C. Tattersall, W. Waterink, B. Van den Berg, R. Van Es, C. Bolman, R. Koper, Self-organising navigational support in lifelong learning: how predecessors can lead the way, Computers & Education 49 (3) (2007) 781–793.

[49] R. Koper, Increasing learner retention in a simulated learning network using indirect social interaction, Journal of Artificial Societies and Social Simulation 8 (2).

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

[51] S. B. Aher, L. Lobo, Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data, Knowledge-Based Systems 51 (2013) 1–14.

[52] X. Zhao, Z. Niu, W. Chen, Interest before liking: Two-step recommendation approaches, Knowledge-Based Systems 48 (2013) 46–56. doi:10.1016/j.knosys.2013.04.009.

[53] Y. Liu, Z. Li, H. Xiong, X. Gao, J. Wu, Understanding of intenal clustering validation measures, IEEE Internatinal Conference on Data mining (2010) 911–916doi:10.1109/ICDM.2010.35.

[54] M. Nilashi, O. bin Ibrahim, N. Ithnin, Hybrid recommendation approaches for multi-criteria collaborative filtering, Expert Systems with Applications 41 (8) (2014) 3879–3900.

[55] S. Gong, A collaborative filtering recommendation algorithm based on user clustering and item clustering, Journal of Software 5 (7) (2010) 745–752. doi:10.4304/jsw.5.7.745-752.

[56] A. Borshchev, The Big Book of Simulation Modeling. Multimethod Modeling with Anylogic 6, Anylogic North America, 2013.

[57] M. Erdt, A. Fernández, C. Rensing, Evaluating recommender systems for technology enhanced learning: A quantitative survey, IEEE Transactions on Learning Technologies 8 (4) (2015) 326–344.

[58] N. Hurley, M. Zhang, Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation, ACM Transactions on Internet Technology 10 (4) (2011) 1–30. doi:10.1145/1944339.1944341.

[59] S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, Proceedings of the 5th ACM conference on Recommender systems - RecSys '11 (2011) 109doi:10.1145/2043932.2043955.

[60] T. Di Noia, V. C. Ostuni, J. Rosati, P. Tomeo, E. Di Sciascio, An analysis of users' propensity toward diversity in recommendations, in: Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14, 2014, pp. 285–288. doi:10.1145/2645710.2645774.

[61] G. Adomavicius, Y. O. Kwon, Improving aggregate recommendation diversity using ranking-based techniques, IEEE Transactions on Knowledge and Data Engineering 24 (5) (2012) 896–911. doi:10.1109/TKDE.2011.15.

[62] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, Y.-C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proceedings of the National Academy of Sciences 107 (10) (2010) 4511–4515. arXiv:0808.2670, doi:10.1073/pnas.1000488107.

[63] N. Katuk, H. Ryu, Does a longer usage mean flow experience? an evaluation of learning experience with curriculum sequencing systems (css), in: Electronic Design, Test and Application (DELTA), 2011 Sixth IEEE International Symposium on, IEEE, 2011, pp. 13–18.