

PLORS: a personalized learning object recommender system

Hazra Imran¹  · Mohammad Belghis-Zadeh¹ · Ting-Wen Chang² · Kinshuk¹ · Sabine Graf¹

Received: 13 February 2015 / Accepted: 30 July 2015 / Published online: 27 August 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract Learning management systems (LMS) are typically used by large educational institutions and focus on supporting instructors in managing and administrating online courses. However, such LMS typically use a “one size fits all” approach without considering individual learner’s profile. A learner’s profile can, for example, consists of his/her learning styles, goals, prior knowledge, abilities, and interests. Generally, LMSs do not cater individual learners’ needs based on their profile. However, considering learners’ profiles can help in enhancing the learning experiences and performance of learners within the course. To support personalization in LMS, recommender systems can be used to recommend appropriate learning objects to learners to increase their learning. In this paper, we introduce the personalized learning object recommender system. The proposed system supports learners by providing them recommendations about which learning objects within the course are more useful for them, considering the learning object they are visiting as well as the learning objects visited by other learners with similar profiles. This kind of personalization can help in improving the overall quality of learning by providing recommenda-

tions of learning objects that are useful but were overlooked or intentionally skipped by learners. Such recommendations can increase learners’ performance and satisfaction during the course.

Keywords Personalization · E-learning · Learning management systems · Recommender system · Association rule mining · Learning objects

1 Introduction

The innovation of information and communication technologies plays an important role in the popularity of e-learning. E-learning can be supported through different forms like web-based learning, computer-based learning, or virtual classrooms and content delivery via e-networks, audio or video tape, satellite TV, video conferencing, CD-ROM, i-pods, e-mails, wireless and mobile technology [1]. E-learning offers various benefits like increased accessibility to information, better content delivery, personalized instruction, content standardization, accountability, on-demand, availability, self-pacing, interactivity, confidence, and increased convenience [2]. Due to the benefits mentioned above, many educational institutions are focusing on e-learning. To organize the learning content in e-learning, learning management systems (LMSs) are typically used. Learning management system can be defined as an “infrastructure that delivers and manages instructional content, identifies and assesses individual and organizational learning or training goals, tracks the progress towards meeting those goals, and collects and presents data for supervising the learning process of an organization as a whole” [3]. Typically, the courses in LMS consists of learning objects (LOs). The IEEE Learning Technology Standards Committee (LTSC), in Learning Object

✉ Hazra Imran
hazraimran@athabasca.ca; hazrabano@gmail.com

Mohammad Belghis-Zadeh
mobelghis@yahoo.ca

Ting-Wen Chang
tingwenchang@bnu.edu.cn

Kinshuk
kinshuk@athabasca.ca

Sabine Graf
sabineg@athabasca.ca

¹ Athabasca University, Edmonton, Canada

² Beijing Normal University, Beijing, China

Metadata Standard, defines a LO as “any object, digital or non-digital, that may be used for learning, education or training” [4]. Generally, LMSs deliver the same kind of course structure and LOs to each learner [5–7] as teachers develop courses based on their preferable teaching methods without considering learners’ characteristics. This is termed as “one size fits all” approach. But, each learner has different characteristics, and therefore, a “one size fits all” approach does not support most learners particularly well.

Personalization in LMS refers to the functionality which enables the system to uniquely address a learner’s needs and characteristics such as levels of expertise, prior knowledge, cognitive abilities, skills, interests, preferences and learning styles [8] so as to improve a learner’s satisfaction and performance within the course. The personalization can ensure that learners’ get different attention, according to their needs. To make LMSs personalized, recommender systems can be integrated into LMSs to recommend learning material based on what worked well or was perceived as useful for similar learners. Personalization in the form of recommendations for resources and learning materials is an area that has gained significant interest from researchers recently. Recommendations exhibit prominent social behavior in day-to-day life [9]. For example, websites like Google [10], Yahoo! [11], Amazon [12], ebay [13] and Netflix [14] provide a personalized mechanism to support users by presenting information that is assumed to be more interesting to them. Often, users need some kind of support in their decisions [15] as they get overwhelmed by the huge amount of available information. Many times, users find particular information interesting if someone has provided a recommendation of it. Such successful integration of recommender systems in e-commerce has prompted researchers to explore similar benefits in the e-learning domain [16,17] since the integration of recommender systems in e-learning has high potential for achieving advanced personalization. For example, in an academic environment, the course contents are generally static. But each learner is different and has different learning needs. Therefore, recommender systems can benefit learners by providing personalized recommendations on which learning objects to focus on next. Furthermore, because of the complexity of some courses in terms of their structure, learners might overlook some of the useful learning objects. Again, recommender systems can help in such situations by suggesting to look at such learning objects, if those learning objects were found to be essential to look at by other similar learners. This is particularly helpful since in the context of e-learning, learners learn at their own pace, and have no feedback from other peers on the usefulness of learning objects. Given the high potential of recommender systems for e-learning, this paper focuses on the research question “How to integrate a recommendation approach in LMSs to find and recommend useful learning objects within a course, considering the learning

object they are visiting as well as the learning objects visited by other learners with similar profiles to facilitate learner’s learning?”. To answer the research question, we introduce the personalized learning object recommender system (PLORS) which can be integrated into LMS to provide personalization to learners. The main key features of the system are:

- It considers learners’ profiles consisting of learners’ learning styles, expertise level, prior knowledge and performance to provide advanced personalization.
- It forms a neighborhood of learners based on their profile and discovers associations among learning objects (through association rule mining) that led to identifying the useful learning objects visited by other similar learners. The approach to find the neighborhood of a learner is advanced as it is considering different characteristics of the learners such as their learning styles, prior knowledge, expertise level and performance within the course. Accordingly, we get more similar learners in a neighborhood, which enables our approach to generate more suitable recommendations that fit to the learners’ need more accurately.
- It creates a personalized list of recommendations of learning objects to be presented to an individual learner based on the navigation history of members of his/her neighborhood.
- Information on whether or not a certain learning object was helpful for a particular learner is retrieved through association rule mining among the learning objects visited by him/her and other similar learners instead of asking him/her to provide a rating on the visited learning objects.

The organization of this paper is as follows. The following section reviews the relevant literature on providing personalization through recommender systems. Section 3 discusses the architecture of the personalized learning object recommender system. Section 4 concludes the paper by summarizing the main contributions of our work and presenting future directions.

2 Related work

Recommender systems support individual user in making decisions from vast available choices by recommending the appropriate choice(s) based on behavior or opinions of a group with the similar characteristics/behavior. Recently, some recommender systems have been applied in the e-learning domain. This section describes such works in two directions: First, we discuss research works focusing on providing recommendations based on the activities done by

learners in the course. These works mainly use association rule mining to find the associations among the activities done by learners and then provide recommendations to the individual learner. In these works, recommendations are based on learners' activities in a course rather than learner characteristics, needs and/or profiles. Second, we describe research works in which recommendations were provided to learners based on other similar learners having similar characteristics/attributes. These works either used clustering techniques which uses learners' characteristics to form groups or calculate the similarity between the learners based on the ratings provided by them. Subsequently, they looked at what worked well for other similar learners and provided respective recommendations. Research work falling under the first group used association rule mining to generate rules based on which recommendations were provided to learners. For example, Zaiane [18] built a recommender agent that provides recommendations of learning activities within a course based on learner access histories. Khribi et al. [19] developed a recommender system based on the learners' recent navigation histories, and similarities and dissimilarities among the contents of the learning materials. Markellou et al. [20] proposed a framework for personalized e-learning. The personalization is done based on the domain ontology and the association among the usage profiles. The system uses Apriori algorithm for finding association rules. The recommended learning materials were determined based on the domain ontology. Furthermore, Liu and Shih [21] used association rule mining as basis for their material recommendation system. The system analyzes the logs to look into learner's learning behavior and to identify the associations between the learning course content. The system used the behavior of previous learners to recommend the learning content.

The first group of research works mainly considers the web usage data of the learners' in a course and associations between the activities of learners in a course. These works focus on grouping similar learners based on their activities in a course. Our work is different from the above-cited works as we are grouping similar learners based on their characteristics (e.g., learning styles, prior knowledge, expertise levels and performance) as opposed to the activities, which has potential to allow for a more accurate grouping since we are considering the basic explanation behind learners' behavior (e.g., not much background knowledge, a certain learning style) rather than just the actions themselves.

The second group of research works finds similar learners based on the characteristics and then recommendations are provided based on the information from the similar learners. For example, Tang and McCalla [22] proposed an evolving web-based learning system that finds the relevant content from the web. They used a clustering technique to cluster the learners (based on their learning interests) to calculate learners' similarities for content recommendation. Tai

et al. [23] proposed a course recommender system based on self-organizing maps and data mining techniques. Self-organizing maps were used to categorize learners based on similar interests into groups. Then a data mining technique was used to draw the rules of the best learning path for each group of learners. Kerkiri et al. [24] proposed a framework that uses reputation metadata in a recommender system. The reputation metadata was the ratings of learning resources provided by the learners. The system used learning object metadata and the learners profile based on PAPI [25]. The registered learners were requested to provide information for their profile including qualifications, skills, licenses, etc. The similarity between the learners was calculated using the Pearson's r correlation coefficient. Having all the information about learners and learning resources (metadata and reputation metadata), collaborative filtering was applied to recommend personalized learning resources. The recommendations of learning material were based on the learners' ratings. Their experiment shows that such recommendations help in increasing learner's satisfaction level. Yang et al. [26] proposed a personalized recommendation algorithm for curriculum resources based on semantic web technology using domain ontology. The algorithm first gathers curriculum resources of interest based on user evaluation and user browsing behavior. Yang et al. [26] assume that "different users evaluate different core concepts, according to domain knowledge, as there is a certain similarity between core concepts, so there are similarities between the user's interests". Therefore, similarity among users can be calculated from the similarity between core concepts. The users were asked to provide ratings to the learning resources. The similarity among learners was computed based on their ratings. Then the interest degree of users is calculated for each interest category of the nearest neighbors, and finally recommendations were provided based on the interest of the nearest neighborhood. Ghauth and Abdullah [27] developed an e-learning content based recommendation system. The system was based on vector space model and learners' rating. The system considered rating of only those learners who had studied learning material and got marks higher than 80 %. Mojtaba and Isa [28] proposed a recommender system to suggest e-learning material. They model the learning material in multidimensional space of material's attributes like authors' name, subject, price and educational level. The learners were modeled in a way so that the learning material attributes can be considered. To consider learning material attributes in learner profiles, learners' ratings were used. The recommendations were generated based on content-based, collaborative-based and hybrid-based methods. The research works mentioned in the second group provide recommendations based on other similar learners. However, these works mainly used the learner interest and learners' ratings as the parameter for generating groups. In the e-learning area, we

generally do not have ratings for the content. If a learner is asked to provide ratings for each learning object in a course, it puts a great deal of exertions on the learner. In our work, we aim at providing automatic recommendations without requiring any additional effort from learners. Instead of using ratings to identify the usefulness of learning objects, we use associations among the learning objects visited by him/her and other similar learners.

Furthermore, our work is different as it considers learners' characteristics, including their learning styles, expertise level, skills and prior knowledge, along with their performance in the course. By recognizing similar learners based on the several characteristics, we believe to place a learner together with learners who learn in a very much alike fashion, leading to more accurate recommendations.

3 PLORS: personalized learning object recommender system

In this section, we introduce the Personalized Learning Object Recommender System (PLORS) which provides recommendations of learning objects within a course. The aim of the system is to enable LMSs to provide recommendations to learners, considering the learning object they are accessing as well as the learning objects visited by other learners with similar profiles. Currently, PLORS can provide the recommendations for the following types of learning objects (LO) but can easily be extended if needed.

1. Commentary—through commentary, learners get a brief idea on what the unit is about.
2. Content objects—these are the learning materials of the course. They are rich in content. They provide the description and the explanation of the concepts covered in the unit of the course.
3. Reflection quizzes—these contain open-ended questions about the concepts covered in the unit. These questions aim at encouraging learners to reflect about their knowledge of the learned material.
4. Self-assessment tests—these tests include several closed-ended questions about the concepts of a unit. By attempting these tests, learners can check their understanding of the concepts of the unit since they receive immediate feedback on whether their answers were correct or not.
5. Discussion forum—through discussion forums, learners can ask questions or can join/initiate a discussion topic with their peers and instructor.
6. Additional reading materials—these materials provide additional sources for reading about the topics in the unit. For example, they can include more detailed explanations of the topic or advanced concepts related to the topic.

7. Animations—animations explain the concepts of the unit in an animated multimedia format.
8. Exercises—through exercises, learners are asked to do some practical work. This helps the *learner* to practice what they have learned in the course.
9. Examples—Examples are used to illustrate the theoretical concepts in a more concrete way to make them more understandable to learners.
10. Real-life applications—these learning objects demonstrate how learners can apply the learned material in real-life situations.
11. Conclusions—summarize the topics learned in a unit.

A course can be divided into different units and unit can (but not necessarily) have several sections. In each section, there can be different number and types of LOs. Figure 1 shows an example of such a course. The course consists of seven units. Unit 0 contains only two types of LOs namely Commentary and Digital Reading Room Activity while Unit 1 consists of nine types of LOs (Commentary, Learning objectives, Additional reading materials, Examples, Real-life applications, Self-assessment quiz, Reflection quiz, Forum activity and Conclusion). The PLORS uses information about learners' behavior through accessing log data tracked by the LMS, which includes what learning objects have been visited by each learner and how much time he/she spent on each learning object. This is information that every LMS typically tracks.

In order to provide recommendations, PLORS finds the neighbors of a learner who have similar characteristics. We are making the assumption that since learners within a neighborhood are similar to each other, learning objects visited by one learner can be beneficial to other similar learners. The overall aim of PLORS is to provide recommendations of learning objects to the learner in a situation where the learner is visiting different learning objects than other similar learners. For example, a learner may be advised to consult some unread material that other similar learners have read before attempting a particular reflection quiz. PLORS has been designed to be integrated in any LMS and consists of four modules that gather information about learners (Learner Modelling Module), create neighborhoods (Neighborhood generation module), generate recommendations (recommendation generation module), and display recommendations to learners (Recommendation display module). Figure 2 depicts the architecture of PLORS. In the next subsections, the four main modules of PLORS are discussed in further detail.

3.1 Learner modelling module

The Learner Modelling Module aims at generating and updating the Learner Model. The Learner Model contains

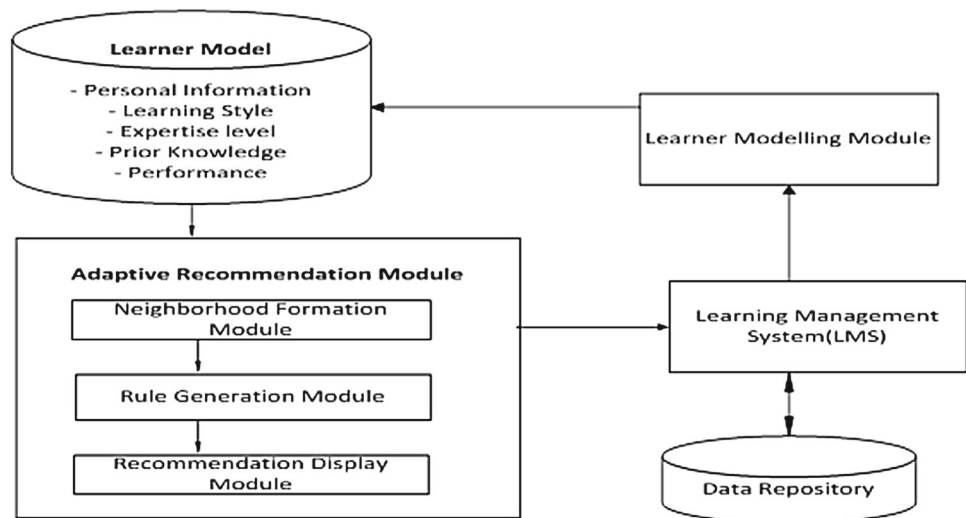
Fig. 1 Structure of a course containing different learning objects

information gathered from the learner, i.e., personal information (first name and last name), previous knowledge (related to the course), expertise level (i.e., Beginner, Intermediate or Expert), learning styles and performance. When learners register in the LMS through a registration form, the Learner Model is initialized. During the registration, learners provide personal information like first name and last name, which is stored in the Learner Model. Furthermore, once they enroll in a certain course, they are asked about their prior knowledge

and expertise level specific to that course. Figure 3 shows the interface that is used to collect this information for a course on Computing and Information Systems.

In addition, the Learning Module aims at gathering information about the learning styles of learners. Every learner learns in a different and unique way, as each one has their own preferences, need and approaches toward learning. These individual differences are coined as learning styles. According to Dunn et al. [29] learningstyles can be defined as

Fig. 2 Architecture of PLORS



Questions related to your Expert Level and Prior Knowledge

Expertise level (Please make appropriate selection indicating your level related to Interactive Technologies)*

Beginner Intermediate Experienced

Prior Knowledge (Please make appropriate selection indicating your prior knowledge in the areas below)

Human and Computer Interaction Input and Output of Interactive Technologies Application interface User interface

Fig. 3 Interface for gathering expertise level and prior knowledge information

“unique manners in which learners begin to concentrate on, process, absorb, and retain new and difficult information”. Learning styles are important in understanding how a learner learns and participates in learning activities. More specifically, a learner’s learning style is the way to gather and understand the information. There are many models about learning styles in literature like Kolb [30], Honey and Mumford [31] and Felder and Silverman [32]. To identify the learning styles, the Learner Modelling Module uses a well-investigated and commonly used questionnaire, called Index of Learning Styles (ILS) [33] developed by Felder and Solomon, which identifies the preferences of learning in four dimensions based on the Felder–Silverman Learning Style Model [32]. These four dimensions are: active/reflective, sensing/intuition, visual/verbal and sequential/global. At the time of registration, a learner is asked to fill out the ILS questionnaire, consisting of 44 questions. Based on a learner’s responses, the result is calculated as four values between +11 and –11 indicating the preference on each of the four learning style dimensions. These four values are stored in the Learner Model and are used as the identified learning styles of learners.

Furthermore, performance data are stored in the Learner Model. Performance data describe a learner’s performance in the course. The performance data are gathered from the

learner’s performance on assignments and quizzes that count towards a learner’s final grade.

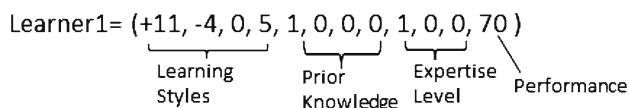
3.2 Adaptivity recommendation module (ARM)

This module is responsible for creating and displaying recommendations based on other similar learners’ profile. ARM has three main steps: neighborhood formation, rule generation, and recommendation display. Each step is discussed in the next subsections in more detail.

Neighborhood formation module In PLORS, we assume that if a learner visits particular LOs then those LOs might be helpful to other similar learners who have not yet visited those LOs. These other similar learners build the neighborhood of a learner and are learners with similar characteristics (i.e., learning styles, prior knowledge, expertise level and performance). The purpose of the neighborhood formation module is to find such other similar learners. There were two main requirements for our algorithm to build a neighborhood: (1) the number of learners in the neighborhood of a particular learner should not be predefined but flexible and (2) the neighborhood should include the data points (learners) that are close to another. Based on the above stated requirements, our approach for finding similar learners is different

from traditional clustering algorithms. Our approach does not demand the number of neighborhoods or neighbors as input a priori and can use a distance measure to place a learner only together with learners who have very similar characteristics.

To find the neighborhood, we use an algorithm that describes each learner, L_i ($i = 1, \dots, m$) as a vector whose components are the values of learner’s learning styles, prior knowledge, expertise level and performance. Each learner vector consist of learner’s four learning style values (between +11 and –11), prior knowledge in the area of Human and Computer Interaction, Input and Output of Interactive Technologies, Application Interface and User Interface (1 mean that learner have the prior knowledge in that the area and 0 mean learner does not have any prior knowledge in that area), expertise level as Beginner, Intermediate or Advanced (0 means that learner does not possess that particular expertise level and 1 mean that learner have the particular expertise level) and the learner’s performance value within the course. For example learner 1 can be represented as following:



The number of registered learners in a specific course is denoted by m . The algorithm computes similarities between learners based on the commonly used distance measure, Euclidean distance. Euclidean distance is used because we assume that all attributes of learners are of same importance. As mentioned before, we are using different characteristics of learners including learning styles, expertise level, prior knowledge and performance. Each characteristic has a different scale of values. To ensure the equal impact of each characteristic, we normalize the data between 0 and 1. Once the characteristics values are normalized, Euclidean distance is used to compute the similarity between learners based on their characteristics. Euclidean distance (L_i, L_j) is the distance between the vectors representing two learners. The formula to calculate the Euclidean distance between two learners is shown in Formula (1).

$$\text{Euclidean_distance}(L_i, L_j) = \sqrt{\sum_{k=1}^n (L_{ik} - L_{jk})^2}, \quad (1)$$

where L_{ik} denotes the characteristic k of learner i and n is the number of characteristics of the learner. In order to calculate the neighborhood of a learner, a threshold t is used as radius for the neighborhood. Accordingly, for a learner L_i , we consider every other learner L_j ($j = 1 \dots m$ and $j \neq i$) as a member of the neighborhood if $\text{Euclidean_distance}(L_i, L_j) \leq t$. To determine a suitable value for a threshold t , we assume that two learners can be considered as similar if the difference between each characteristic is on average

Table 1 Example of transactions

Learner ID	Unit number	Learning objects visited by the learner
1	1	{Commentary, learning objectives, reflection quiz}
2	1	{Commentary, learning objectives, forum}
3	1	{Learning objectives, reflection quiz}
4	1	{Commentary, reflection quiz}
5	1	{Learning objectives, reflection quiz, forum}

equal or lower than 0.25 (on a scale from 0 to 1). Accordingly, the Euclidean distance between two such learners would be 0.66. Therefore, we consider 0.66 as threshold to calculate the neighborhood.

Recommendation generation module To generate recommendations, association rule mining [34] is used. Association rule mining is a technique of finding associations and relationships among the itemsets. Given an itemset I (in our case a set of learning objects visited by the learner), and a transaction $T \subset I$ is defined as set of learning objects visited by similar learner in a unit. An association rule between two sets A and B , such that $A, B \subset I$ and where A and B are two disjoint itemsets (i.e., sets with no common items). This means that the learning objects visited in the set A in the transaction T indicates a strong probability that learning objects visited in the set B are also present in T . Such an association rule is symbolized as $A \Rightarrow B$. Table 1 shows an example of such transactions belonging to one particular unit (e.g., unit 1). For example, such rule could be $\{\text{Commentary}\} \geq \{\text{Learning Objectives}\}$. The rule suggests that learners who have visited the Commentary learning object in unit 1 have also visited the Learning Objectives of unit 1.

The association rules are evaluated based on two measures namely, support and confidence. Support is defined as the percentage of the transactions containing both A and B among all transactions [34]. The formula to calculate support is shown in Formula (2).

$$\begin{aligned} \text{Support}(A \Rightarrow B, T) &= \frac{\text{Number of transactions containing } A \cup B}{\text{Total number of transactions}} \end{aligned} \quad (2)$$

Confidence is defined as the percentage of transactions that contain B among transactions that contain A [34]. The formula to calculate confidence is shown in Formula (3).

$$\begin{aligned} \text{Confidence}(A \Rightarrow B, T) &= \frac{\text{Number of transactions containing } A \cup B}{\text{Number of transactions containing } A} \end{aligned} \quad (3)$$

For example, let us consider an association rule with support 0.5 and confidence 0.7.

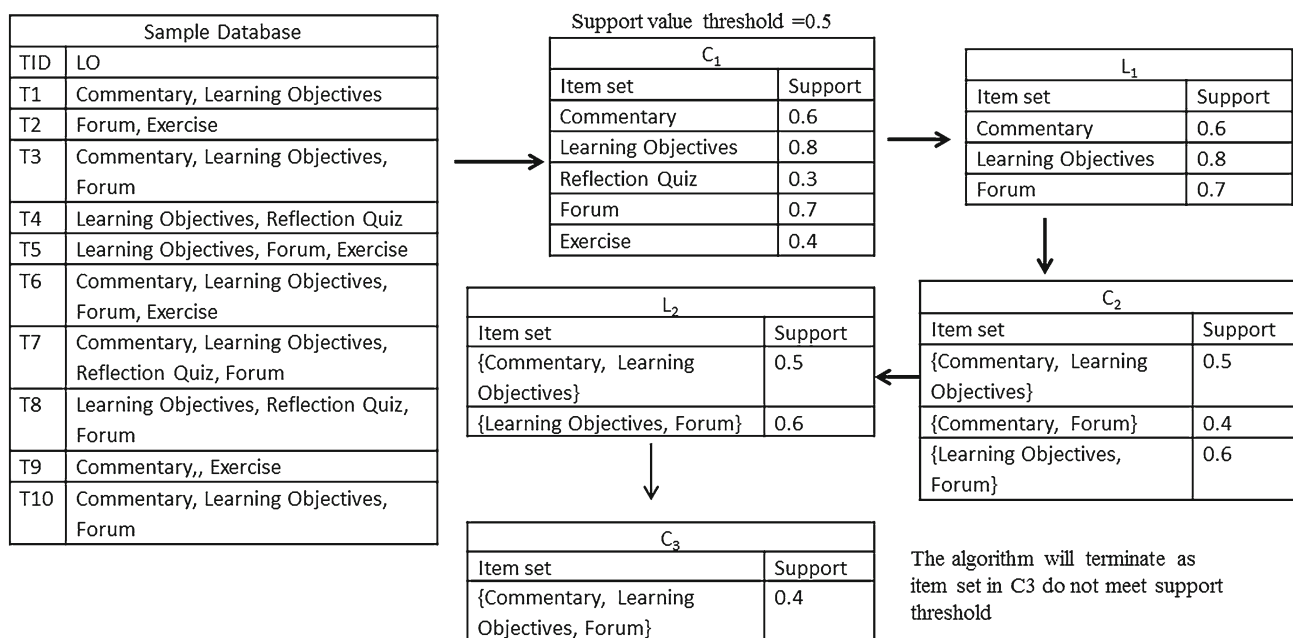


Fig. 4 Process of Apriori algorithm to find large itemset

{Commentary, Additional Reading Material}
=> {Self – Assessment Test}

This means that there is likelihood that learners who visited LOs namely, Commentary and Additional Reading Material also visited Self-Assessment Test (with confidence 70 %). The support value represents the fact that the itemset {Commentary, Additional Reading Material, and Self-Assessment Test} was present in 50 % of similar learners’ transactions. Association rule discovery methods initially find sets of items occurring frequently together in transactions, satisfying a minimum support (*minsup*) criteria. Such itemsets are referred as *frequent itemsets*. The goal of association rule mining technique is to find all the rules from transactions, *T* having support \geq *minsup* threshold and confidence \geq *minconf* threshold [34]. We progressed to the association rule discovery by applying the Apriori algorithm [35]. The Apriori algorithm is used to control the growth of candidate itemsets based on Apriori principle, which states “Any subset of frequent itemset must be frequent” [34, 35]. The candidate itemsets (denoted by *C*) are the sets of itemsets that require validation that they confirm the requirement (support \geq *minsup* and confidence \geq *minconf*). The frequent itemsets (denoted by *L*) are the largest possible sets, containing common items (in our case are the LOs visited by similar learners).

Apriori algorithm discovers the frequent itemsets in repeated iterations. For every similar learner, the transaction ID and each LO visited in a unit is maintained. At first step, all visited LOs are placed as candidate 1-itemset, and their support count is recorded. From Fig. 4, the large itemsets (*L*₁) are

generated by removing LOs having minimum support (*minsup*). The large items are also called frequent 1-itemset. In the subsequent step, candidate itemset (*C*_{*k*}) are generated by joining *L*_{*k*-1} X *L*_{*k*-1}. The candidate itemsets whose subsets are infrequent are discarded. And the remaining *C*_{*k*} itemsets are used for finding frequent *k* + 1 itemsets (large itemsets). An example is illustrated in Fig. 4 to show the detailed process of Apriori algorithm. This example includes a sample database, which contains ten transactions where each transaction denotes the learning objects visited by similar learners in unit 1. The minimum support value is set to be 0.5. This example shows how Apriori algorithm finds large itemsets.

After obtaining large itemsets, the association rules are generated based on minimum support and minimum confidence value. Figure 5 shows an example of finding rules. Here, minimum confidence value is 0.7.

Based on above figure, the itemset {Learning Objectives Commentary} would be discarded as the confidence value is 0.62 which is less than the minimum confidence value (0.7). Accordingly, following rules would be generated:

1. Commentary \Rightarrow Learning Objectives Support = 0.5; Confidence = 0.83
2. Learning Objectives \Rightarrow Forum Support = 0.6; Confidence = 0.80
3. Forum \Rightarrow Learning Objectives Support = 0.6; Confidence = 0.85.

To provide recommendations to a learner, PLORS consults the association rules to check for mismatches between the learning objects visited by the current learner and the learning

Fig. 5 Generation of association rules

L_2			
Item sets	Support (A,B)	Support(A)	Confidence
{Commentary, Learning Objectives}	0.5	0.6	0.83
{Learning Objectives, Forum}	0.6	0.8	0.75
{Learning Objectives, Commentary}	0.5	0.8	0.62
{Forum, Learning Objectives}	0.6	0.7	0.85

→ Less than 0.7

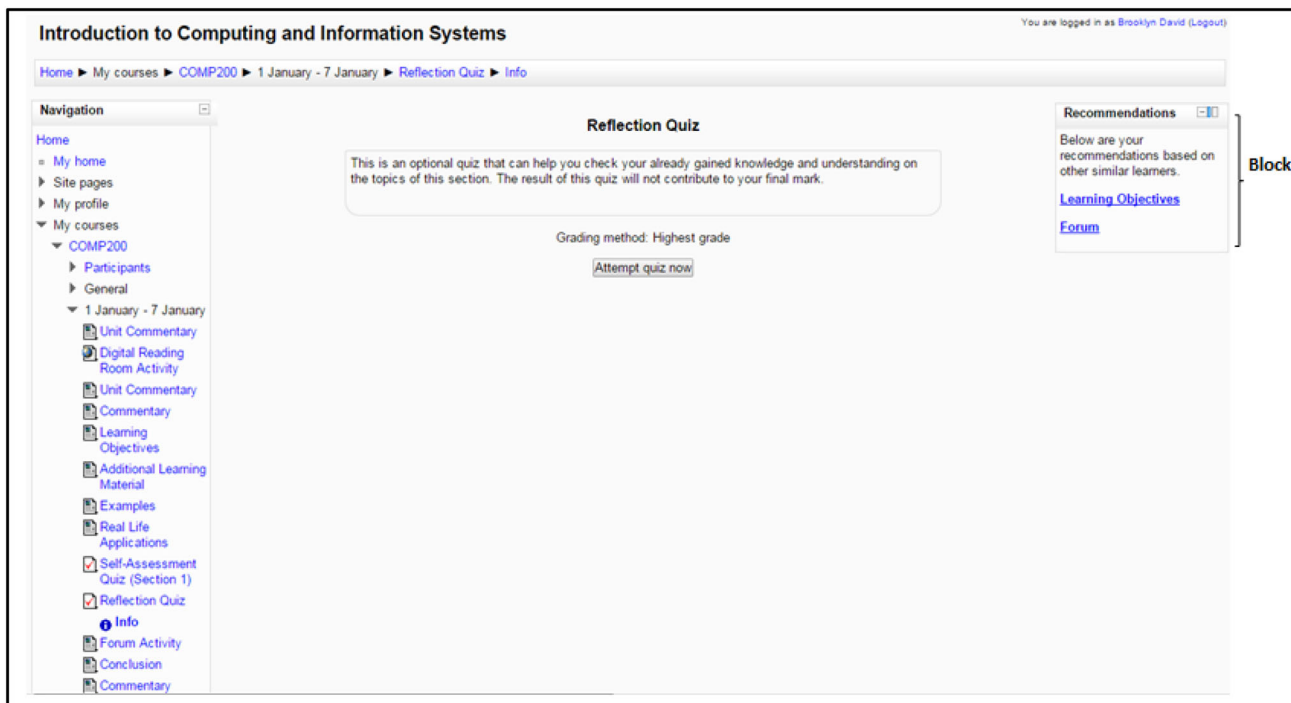


Fig. 6 Example of personalized recommendations

objects visited by the learners within the neighborhood. For example, suppose the current learner has not visited Learning Objectives yet and he/she is trying to visit Forum, but other similar learners in his/her neighborhood have visited Learning Objectives before visiting Forum. In such case, the recommendation to be provided (to the current learner) is to visit Learning Objective before Forum. Such recommendations are then passed to the recommendation display for being presented to the learner. *Recommendation display module* This module is responsible to display the personalized recommendations to the learner in an informative, precise and simple way. Recommendations are provided as a block on course page. Blocks are items that are added to the page of LMS (in our case we use Moodle LMS). The recommendation block is made sticky so that recommendations are shown on every page of the course. Recommendation block includes links to the recommended learning objects so that

the learners can go to these learning objects easily. When the learner clicks on any recommended learning object link then the learning object get opened. In case, where there are no Los for recommendation then a message is shown to learner (i.e., there is no recommendation available for you. Please check back at later time!). Figure 6 shows an example of a recommendation for a learner.

In the example, when the learner tries to attempt learning object in a course (i.e. Reflection Quiz), the recommender system recommends two learning objects namely, Learning Objectives and Forum. The learner may choose to visit Learning Objective first by clicking on the respective link. In this case, Learning Objective gets open. Or, if learner wants, he/she can open all the recommendations as separate tab. When the learner tries to attempt Reflection Quiz again, then the new recommendation is generated based on other similar learners' learning experience.

4 Conclusions and future work

This paper introduces the PLORS which integrates a recommender system approach into learning management systems. PLORS provide recommendations on which learning objects within a course are more useful for learners, considering the learning object he/she is visiting as well as the learning objects visited by other learners with similar profiles. The recommendation mechanism uses association rule mining to find the association between LOs and a neighborhood algorithm. The main contributions of the work are: first, to find similar learners, our system does not consider ratings given by learners as done in most of the traditional recommender systems. Instead, it uses different characteristics/attributes of learners like learning styles, previous knowledge, expertise level, and performance to identify highly similar learners. Secondly, the usefulness of a learning object is not based on learners' rating as done in most recommender systems. Instead of that, PLORS looks into other learners' navigation history and find the set of LOs which were commonly visited by other similar learners in the course. Third, in most of the previous works similar learners are found by using a clustering approach. In our work, we consciously decided against a clustering algorithm. Clustering algorithms typically aim at assigning each learner to a group/cluster. This leads to several relevant drawbacks such as the risk of creating clusters that include data points (or learners) that are actually not too close, the risk of getting different clusters when running the same clustering algorithm again, meaning that the clustering algorithm does not always group the nearest data points (or learners), or the need for a predefined number of clusters. Since our aim is to find learners who are close to a particular learner, a neighborhood approach is more accurate and free of the above mentioned drawbacks. By using such neighborhood approach, we expect to place a learner only together with learners who learn in a very similar way, and use the experience of similar learners to provide accurate recommendations. Fourth, while most other works focus on using recommender system in particular e-learning systems, the aim of our work is to integrate a recommender system into any LMS. LMSs are commonly used by educational institutions and by enhancing LMSs with personalized functionality to provide individual recommendations, teachers can continue using the systems that they are already using for online learning and learners are receiving additionally some personalized support. The provided recommendations can help learners to better navigate the course, and therefore can improve their learning performance and satisfaction. Currently, we are providing recommendations of learning objects within a course. As a future work, we will extend the system to additionally provide recommendations of learning objects from the web based on other similar learners' profiles.

Acknowledgments The authors are grateful to MITACS for their partial financial support through the ELEVATE program. The authors acknowledge the support of Athabasca University, NSERC, iCORE, Xerox, and the research related gift funding by Mr. A. Markin.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Eke, H.N.: Modeling LIS students' intention to adopt e-learning: a case from University of Nigeria, Nsukka. *Libr. Philos. Pract.* **1**, 113 (2011)
2. Bhuasiri, W., Xaymoungkhoun, O., Zo, H., Rho, J.J., Ciganek, A.P.: Critical success factors for e-learning in developing countries: a comparative analysis between ICT experts and faculty. *Comput. Educ.* **58**(2), 843–855 (2012)
3. Szabo, M.: CMI theory and practice: historical roots of learning management systems. *World Conf. E-Learn. Corp. Gov. Healthc. High. Educ.* **1**, 929–936 (2012)
4. IEEE Learning Technology Standardization Committee: Draft Standard for Learning Object Metadata (IEEE 1484.12.1-2002), New York (2002). <http://ltsc.ieee.org/wg12/index.html>. Accessed January 2015
5. Brusilovsky, P., Miller, P.: Course delivery systems for the virtual university. *Knowl. New Inf. Technol. Emerg. Virtual Univ.* 167–206 (2001)
6. Shishehchi, S., Banihashem, S.Y., Zin, N.A.M., Noah, S.A.M.: Review of personalized recommendation techniques for learners in e-learning systems. In: *Proceeding of International Conference on Semantic Technology and Information Retrieval (STAIR)*, pp. 277–281. IEEE, New York (2011)
7. Graf, S., Kinshuk, M.C., Ives, C.: A flexible mechanism for providing adaptivity based on learning styles in learning management systems. In: *Proceedings of the International Conference on Advanced Learning Technologies (ICALT 2010)*, Sousse, pp. 30–34. IEEE, New York (2010)
8. Huang, M.-J., Huang, H.-S., Chen, M.-Y.: Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach. *Expert Syst. Appl.* **33**(3), 551–564 (2007)
9. Tseng, C.: Cluster-Based Collaborative Filtering Recommendation Approach. Masters Thesis, Information Management Department, National Sun Yat-sen University. <http://etd.lib.nsysu.edu.tw/ETD-db/ETD-search/getfile> (2003)
10. Google: <https://www.google.ca/>. Accessed January 2015
11. Yahoo!: <https://ca.yahoo.com/>. Accessed January 2015
12. Amazon: <http://www.amazon.com/>. Accessed January 2015
13. ebay: <http://www.ebay.ca/>. Accessed January 2015
14. Netflix: <https://www.netflix.com/>. Accessed January 2015
15. Linden, G., Smith, B., York, J.: Amazon.com recommendations: item-to-item collaborative filtering. *Internet Comput. IEEE* **7**(1), 76–80 (2003)
16. Capuano, N., Iannone, R., Gaeta, M., Miranda, S., Ritrovato, P., Salerno, S.: A recommender system for learning goals. In: *Information Systems, E-learning, and Knowledge Management Research*, pp. 515–521. Springer (2013)
17. Manouselis, N., Drachsler, H., Verbert, K., Duval, E.: *Recommender Systems for Learning*. Springer (2013)

18. Zafane, O.R.: Building a recommender agent for e-learning systems. In: Proceedings of the International Conference in Education, Auckland, pp. 55–59. IEEE (2002)
19. Khribi, M.K., Jemni, M., Nasraoui, O.: Automatic recommendations for e-learning personalization based on web usage mining techniques and information retrieval. In: Proceeding of the International Conference on Advanced Learning Technologies, pp. 241–245. IEEE (2008)
20. Markellou, P., Mousourouli, I., Spiros, S., Tsakalidis, A.: Using semantic web mining technologies for personalized e-learning experiences. In: Proceedings of the Web-based Education, pp. 461–826 (2005)
21. Liu, F.-J., Shih, B.-J.: Learning activity-based e-learning material recommendation system. In: Ninth IEEE International Symposium on Multimedia Workshops, ISMW '07, pp. 343–348. IEEE, New York (2007)
22. Tang, T., McCalla, G.: Smart recommendation for an evolving e-learning system: architecture and experiment. *Int. J. E-learn.* **4**(1), 105–129 (2005)
23. Tai, D.W.-S., Wu, H.-J., Li, P.-H.: Effective e-learning recommendation system based on self-organizing maps and association mining. *Electron. Libr.* **26**(3), 329–344 (2008)
24. Kerkiri, T., Manitsaris, A., Mavridou, A.: Reputation metadata for recommending personalized e-learning resources. In: Proceedings of the Second International Workshop on Semantic Media Adaptation and Personalization, pp. 110–115. IEEE (2007)
25. PAPI: http://metadata-standards.org/Document-library/Meeting-reports/SC32WG2/2002-05-Seoul/WG2-SEL-042_SC36N0175_papi_learner_core_features.pdf. Accessed January 2015
26. Yang, Q., Sun, J., Wang, J., Jin, Z.: Semantic web-based personalized recommendation system of courses knowledge research. In: Proceedings of the International Conference on Intelligent Computing and Cognitive Informatics, pp. 214–217. IEEE (2010)
27. Ghauth, K.I., Abdullah, N.A.: Measuring learner's performance in e-learning recommender systems. *Australas. J. Educ. Technol.* **26**(6), 764–774 (2010)
28. Salehi, M., Kmalabadi, I.N.: A hybrid attribute-based recommender system for e-learning material recommendation. *IERI Proc.* **2**, 565–570 (2012)
29. Dunn, R., Dunn, K., Freeley, M.: Practical applications of the research: responding to students' learning styles-step one. III. *State Res. Dev. J.* **21**(1), 1–21 (1984)
30. Kolb, A.Y.: The Kolb learning style inventory-version 3.1 2005 technical specifications. Hay Resource Direct 200, Boston (2005)
31. Honey, P., Mumford, A.: The Learning Styles Helper's Guide. Peter Honey Maidenhead, Berkshire (2000)
32. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. *Eng. Educ.* **78**(7), 674–681 (1988)
33. Felder, R., Soloman, B.: Index of Learning Styles Questionnaire. North Carolina State University (2001). <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>. Accessed January 2015
34. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record 1993, vol. 2, pp. 207–216. ACM (1993)
35. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Santiago, Chile, pp. 487–499 (1994)