Toward Recommending Learning Tasks in a Learner-Centered Approach*

Hazra Imran[a], Mohammad Belghis-Zadeh[a], Ting-Wen Chang[b], Kinshuk[a], Sabine Graf[a]

[a] Athabasca University, Edmonton, Canada
{hazraimran,kinshuk,sabineg}@athabascau.ca;mobelghis@yahoo.ca
[b] Beijing Normal University, China
tingwenchang@bnu.edu.cn

Abstract. Learner-centered education becomes more and more popular. One way of offering learner-centered education is to have assignments where learners can select from a pool of learning tasks with different difficulty levels (e.g., many easy tasks, few challenging tasks, etc.). However, a problem that learners can face in such assignments is to select the tasks that are most appropriate for them. In this paper, we introduce a rule-based recommender system that supports learners in selecting learning tasks. Such recommendations aim at helping learners to select the tasks from which they can benefit most in terms of maximizing their learning.

Keywords: Recommender system, personalization, learning management system

1 Introduction

The demand of online learning is growing quickly as it provides learners with many advantages such as learning wherever and whenever they want, at their own pace. There are two main pedagogies for learning: teacher-centered and learner-centered. In teacher-centered learning, the teacher directs the learning process and decides what a learner has to do during a course. While this pedagogy has been very common, nowadays, a shift can be seen towards learner-centered learning [1]. Learner-centered learning [2] is an approach where a learner chooses what and how to learn. The benefits of such paradigm are that it motivates learners, promotes active learning, and can enhance their performance [3, 4]. When it comes to selecting what to learn, learners can base their decision on their knowledge and interests. However, in some situations, selecting the right topics, materials, and/or tasks can be difficult. In such situations, recommender systems can help making appropriate selections. Several recommender systems have been implemented in e-learning [e.g., 5, 6]. However, most of these recommender systems focus on recommending learning objects or learning material.

In this paper, we introduce a rule-based recommender system that aims at recommending learning tasks. Following a learner-centered approach, more and more

* The authors acknowledge the support of Mitacs, NSERC, iCORE, Xerox, Athabasca University and the research related gift funding by Mr. A. Markin.

G. Chen et al. (eds.), *Emerging Issues in Smart Learning*,
Lecture Notes in Educational Technology, DOI 10.1007/978-3-662-44188-6_47

courses have assignments that allow learners to select from a pool of learning tasks with different difficulty levels (e.g., a learner can choose to complete many easy tasks, a few challenging tasks, etc.). The proposed rule-based recommender system supports learners by providing them with recommendations on which learning tasks are most suitable for them, considering their performance, the performance of other similar learners, and which tasks the learners selected initially.

The remainder of the paper is structured as follows: Section 2 presents related work. Section 3 describes the rule-based recommender system. Section 4 presents the validation of the approach. Finally, Section 5 concludes the paper by summarizing the main contributions of our approach and presenting future directions.

2 Related Work

Recommender systems help users in making decisions from available choices. Recommender systems for e-learning platforms use techniques such as clustering and/or learner ratings to find other similar learners and provide recommendations based on what worked well for those other similar learners. For example, Tang & McCalla [6] developed an e-learning system to recommend technical articles like conference papers and book chapters to learners in a course. They used clustering approach to find similar learners based on learners' interests, background knowledge and recommendation goals. The recommendations were based on the usage and ratings of the papers.

Some recommender systems consider the difficulty level, for example, of a learning object in their recommendations. For instance, Chen et al. [7] proposed a personalized system to recommend course material to learners. Their system uses item response theory to estimate the abilities of learners based on their responses to questions related to the difficulty and content of course material. Based on a learner's response, the system re-evaluates the learner's ability, tunes the difficulty parameters and then recommends appropriate course materials. Another example is given by Kumaran & Sankar [5], proposing a recommender system to recommend a "topic of study" to learners. The system uses a semantic network to represent learner profile and domain knowledge. The recommendations are based on rating and performance of the learner.

Our work is different from existing research works in several ways. First, while most recommender systems focus on recommending learning objects or materials, our system aims at recommending learning tasks. Second, while many recommender systems depend on learners' ratings, our system is based on the actual performance of similar learners, which is automatically gathered by the system. Third, we consider the difficulty level of learning tasks in a detailed manner, where our approach first identifies priorities for difficulty levels and then selects tasks within the difficulty levels that are most suitable. Fourth, many recommender systems build groups of similar learners using clustering. Instead of clustering, our recommender system uses a neighborhood approach to find similar learners, which provides the advantage that the number of clusters and the number of learners within a cluster do not have to be predefined and therefore, ensures that only very similar learners are grouped together.

3 Architecture of a Rule-based Recommender System

In this section, we introduce a rule-based recommender system that provides recommendations of learning tasks within a course. The system has been designed to be integrated in any learning management system. Fig 1 depicts the architecture of the system. In the next subsections, the four main modules of the rule-based recommender system are discussed in further detail.

3.1 Learner Modelling Module

The Learner Modelling Module aims at gathering information about the learner. At this point, this module considers four types of information: learning styles, prior knowledge, expertise level and performance. Learning styles indicate a learner's preferences and approaches towards learning. To identify these learning style preferences, a well-investigated and commonly used questionnaire called Index of Learning Styles (ILS) [8] is provided to learners during the registration process. Furthermore, prior knowledge about the topics in the course and expertise level related to the course is gathered during the registration process. In addition, the module gathers learners' performance data throughout the course, whenever learners receive marks on tasks. All gathered data are stored in the Learner Model.

3.2 Neighborhood Generation Module

The Neighborhood Generation Module aims to find the neighbors of a target learner (a learner for whom a recommendation should be calculated). To find such neighbors, an algorithm is used that describes each learner, L_i (i =1,…,m) as a vector consisting of a learners' characteristics (i.e., learning styles, expertise level, prior knowledge and performance). The similarities between learners are computed based on the commonly used distance measure, Euclidean distance. In order to ensure the equal effect of each characteristic, the data are normalized to values between 0 and 1. Euclidean distance (L_i, L_j) is the distance between the vectors representing the two learners L_i and L_j. The formula to calculate the Euclidean distance between two learners is shown in Formula (1).

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

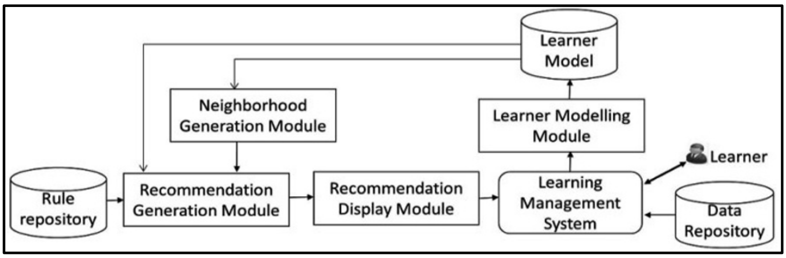

Fig. 1. Architecture of rule-based recommender system

where L_{ik} denotes the characteristic k of learner i and n denotes the number of characteristics considered. In order to calculate the neighbors, a threshold t is used as radius. Accordingly, for a target learner L_i, we consider every other learner L_j (j=1 … m and j != i) as a neighbor if *Euclidean_distance* $(L_i, L_j) <= t$. 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 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.

3.3 Recommendation Generation Module

The Recommendation Generation Module aims to generate suitable recommendations for a target learner. Each learning task can have one of three difficulty levels: Easy (E), Moderate (M), and Challenging (C). The recommendations are based on (1) the target learner's previous performance on tasks of each difficulty level within the whole course, (2) the average performance of the neighbor learners on tasks of each difficulty level in the unit where a recommendation has to be provided and (3) the selection of learning tasks proposed by the target learner to complete within the unit where a recommendation has to be provided. Since information about the target learner's performance is essential to provide a proper recommendation, the system does not provide a recommendation for the target learner's first unit. The recommendation generation process is based on following two steps:

Ranking of difficulty levels. This step aims to determine how well each difficulty level suits a target learner and accordingly determine a ranking of difficulty levels. Each difficulty level is associated with one of three priority levels: highest priority level (HPL), medium priority level (MPL) and low priority level (LPL). HPL indicates that the respective difficulty level (can be easy, moderate or challenging) is most appropriate for a target learner and will be recommended with highest priority. Each difficulty level is assigned to one priority level, where each priority level can be assigned only once. Table 1 shows the rules to identify HPL. If there is no information available about the target or neighbor learners' performance on tasks with certain difficulty level, then this difficulty level is ignored in the part concerning the target and neighbor learners' performance.

Table 1. Rules to identify HPL

	Target Learner' Performance (whole course)		Neighbors' Performance (respect. unit)		HPL
If	E > M & C	AND	E > M & C	Then	Easy
If	E > M & C	AND	M > E & C	Then	Moderate
If	E > M & C	AND	C > E & M	Then	Easy
If	M > E & C	AND	E > M & C	Then	Easy
If	M > E & C	AND	M > E & C	Then	Moderate
If	M > E & C	AND	C > E & M	Then	Challenging
If	C > E & M	AND	E > M & C	Then	Challenging
If	C > E & M	AND	M > E & C	Then	Moderate
If	C > E & M	AND	C > E & M	Then	Challenging

Table 2. Example to show how MPL and LPL are assigned

	Moderate	Challenging
Average target learner's performance (on all previously conducted task within the course)	70%	N/A
Average neighbor learners' performance (on the tasks in the unit where recommendation is sought)	80%	50%
Combined average performance of target learner and neighbor learners	75%	50%

Once HPL is identified, the module next determines which difficulty levels should be assigned to MPL and LPL. In order to do that, again, the average performance of the target learner on previously conducted tasks and the average performance of neighbor learners on tasks in the particular unit are considered. A combined average is built for both performances (from the target learner and the neighbor learners) and the difficulty level with the higher result is assigned to MPL and the difficulty level with the lower result is assigned to LPL. Table 2 shows an example where the difficulty level *easy* is already assigned to HPL and no information about the performance on challenging tasks is available for the target learner. Based on the results of the combined average performance of the target learner and neighbor learners (shown in Table 2), the difficulty level *moderate* is assigned to MPL and the difficulty level *challenging* is assigned to LPL. As a result, the overall ranking in this example shows that, easy tasks are more suitable than moderate tasks and moderate tasks are more suitable than challenging tasks.

Selection of learning tasks based on the ranks of difficulty levels. The aim of this step is to select the learning tasks that work best for the target learner based on the priority levels identified in the previous step, the average performance of the neighbor learners on the respective tasks (*Avg_N_Perform*) and what tasks the target learner chose initially. Twelve rules are used (shown in Table 3) and these rules are applied in the sequence shown in Table 3. The rules are applied until enough tasks are selected (e.g., until a maximum mark or number of tasks is reached). If *Avg_N_Perform* of a task is below a certain threshold *T*, this task is not selected since neighbor learners performed poorly on this task.

Table 3. Rules for the selection of learning tasks

1	Select HPL tasks where *Avg_N_Perform* > *T* and is SELECTED by target learner
2	Select HPL tasks where *Avg_N_Perform* > *T* and is NOT_SELECTED by target learner
3	Select HPL tasks where *Avg_N_Perform* is UNKNOWN and is SELECTED by target learner
4	Select HPL tasks where *Avg_N_Perform* is UNKNOWN and is NOT_SELECTED by target learner
5	Select MPL tasks where *Avg_N_Perform* > *T* and is SELECTED by target learner
6	Select MPL tasks where *Avg_N_Perform* > *T* and is NOT_SELECTED by target learner
7	Select MPL tasks where *Avg_N_Perform* is UNKNOWN and is SELECTED by target learner
8	Select MPL tasks where *Avg_N_Perform* is UNKNOWN and is NOT_SELECTED by target learner
9	Select LPL tasks where *Avg_N_Perform* > *T* and is SELECTED by target learner
10	Select LPL tasks where *Avg_N_Perform* > *T* and is NOT_SELECTED by target learner
11	Select LPL tasks where *Avg_N_Perform* is UNKNOWN and is SELECTED by target learner
12	Select LPL tasks where *Avg_N_Perform* is UNKNOWN and is NOT_SELECTED by target learner

This threshold is set at 60% by default but can be changed by users (e.g., teacher or administrators) if needed. Furthermore, a task is considered as *SELECTED* if a target learner has chosen this task initially in his/her plan and considered as *NOT_SELECTED* otherwise. As a result of this step, the most appropriate learning tasks are selected for recommendation and then passed to the recommendation display module.

3.4 Recommendation Display Module

This module displays the recommendations to the learner. When a learner visits a unit the first time, the recommendations are shown in a pop up window. The target learner can accept them or ignore them. In any way, the recommendations are saved and can be accessed at any time through a button "Your recommendation" at the top of every page within the unit.

4 Validation

To evaluate our approach we have implemented it into the learning management system Moodle and applied it to a university-level course on Interactive Technologies. The course consists of four units that can be completed in any order. Each unit has one assignment and each assignment consists of several tasks. In order to achieve full marks on an assignment, learners need to get 22 points. However, learners can select different sets of tasks in order to reach these 22 points (e.g., completing many easy tasks, completing few challenging tasks, etc.). Before learners start to learn in the course, they are asked to read through all tasks of all units and select which tasks they plan to do. Then learners have to submit their plan of selected tasks. However, learners are free to alter the plans later.

In the following paragraphs, we present a case study to illustrate how our approach for providing learners with recommendations for personalized plans works. Let us consider a learner named Mary. When Mary registers for the course, besides providing regular information in the registration process, she is asked to fill out a learning style questionnaire as well as provide information about her expertise level and prior knowledge related to Interactive Technologies (as shown in Fig. 2).

Before starting to learn in the course, Mary submits a plan showing what tasks she plans to complete in each unit. Then, Mary completes Unit 1 and performs better in moderate tasks than easy ones, and has not tried any challenging tasks.

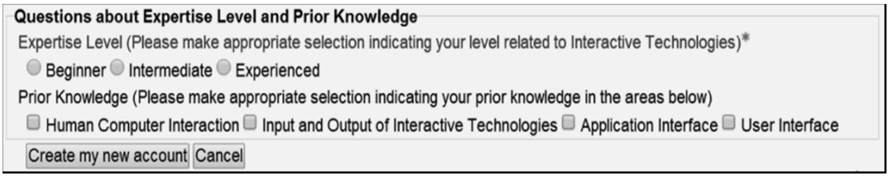

Fig. 2. Interface for gathering additional information about a learner

Table 4a. Average performance of Mary and neighbor learners

	Easy	Challenging
Average performance of Mary in Unit 1	86%	N/A
Average performance of neighbor learners in Unit 2	78%	67%
Combined average performance of Mary and neighbor learners	82%	67%

Table 4b. Mary's initial plan

Task	Difficulty level
2.2	M
2.4	C
2.5	C
2.6	C

Table 4c. Average performance of neighbor learners in Unit 2

Task	Difficulty Level	Average Perform.
2.1	E	78%
2.2	M	81%
2.3	M	87%
2.4	C	70%
2.5	C	74%
2.6	C	59%
2.7	M	76%

Once Mary starts the next unit, the system provides her with a personalized plan. In order to do so, the system looks for neighbors of Mary based on her characteristics and then the system calculates the average performances of neighbors for tasks in each difficulty level. The system finds that, on average, Mary's neighbors performed better on moderate than easy and challenging tasks in that particular unit .According to the rules in Table 1, moderate tasks are the most appropriate difficulty level for Mary in this unit. In the next step, the ranking of the other two difficulty levels is determined. Table 4a shows the average performance of Mary on previously conducted easy and challenging tasks, the neighbor learners' average performance on easy and challenging tasks in Unit 2, and the combined averages of these two performances for easy and challenging tasks. According to Table 4a, for Mary easy tasks fit better than challenging tasks. Overall, this means that moderate tasks fit best for Mary, then easy and then challenging tasks. After that, the system checks Mary's initial plan (shown in Table 4b), the ranking of difficulty levels and the average performance of Mary's neighbors on tasks of each difficulty level in Unit 2 (shown in Table 4c), and applies the rules for selecting individual tasks (shown in Table 3). Accordingly, the first rule in Table 3 is triggered and leads to the selection of task 2.2. Subsequently, the second rule is triggered and leads to the selection of tasks 2.3 and 2.7. After that, the sixth rule is triggered and task 2.1 is selected. With these four selected tasks, Mary can score 22 points. Therefore, the stop condition is reached and no further tasks are selected. The final recommendation displayed to Mary is shown in Fig. 3.

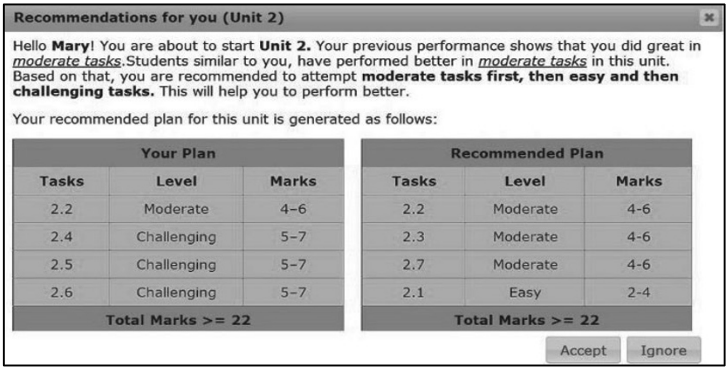

Fig. 3. An example of a displayed recommendation

5 Conclusions and Future Work

This paper presents a rule-based recommender system to provide recommendations of learning tasks to learners in a learning management system. A recommendation is based on the previous performance of the learner, the performance of neighbor learners, who have similar characteristics, and the selected learning tasks by the learner, considering the difficulty level of learning tasks for each of these criteria. The proposed recommender system supports learner-centered learning and helps learners to select tasks that are most suitable for them, with the focus on maximizing their learning. Furthermore, the system requires very little effort from learners. While many traditional recommender systems require learner ratings (e.g., of learning objects, etc.), the proposed system uses actual performance of other similar learners to identify which tasks worked well for those similar learners. In addition, the proposed recommender system uses an advance neighborhood approach to find similar learners. The neighborhood approach considers different characteristics of learners such as their learning styles, prior knowledge, expertise level and performance within the course and finds neighbors based on a distance measure. This enables our system to generate more suitable recommendations that support learners more effectively, leading to a better selection of learning tasks from which learners can benefit most. Currently, our recommender system requires teachers to label tasks based on their difficulty level. As future work, we will extend our approach to automatically identifying the difficulty levels of tasks and labelling tasks respectively. This would, on one hand, reduce the work required from teachers when using our system and, on the other hand, provide teachers with valuable feedback about how difficult their tasks are for their learners.

References

1. Rich, J. D., Colon, A.N., Mines, D., Council, C.: Learner-centered assessment strategies for greater student retention. Universal Journal of Education and General Studies, 2(6), 196-199 (2013)
2. Overby, K.: Student-Centered Learning: ESSAI: The College of DuPage Anthology of Academic Writing across the Curriculum, 9(32), 109-112 (2011)
3. O'Neill, G., McMahon, T.: Student-centered learning: What does it mean for students and lecturers? In O'Neill, G., Moore, S., McMullin, B. (eds.),Emerging Issues in the Practice of University Learning and Teaching, 1, 27-36, Dublin, Ireland (2005)
4. Armbruster, P., Patel, M., Johnson, E., Weiss, M.: Active learning and student-centered pedagogy improve student attitudes and performance in introductory biology. In: CBE-Life Sciences Education, 8(3), 203-213 (2009)
5. Kumaran, V. S., Sankar, A.: Recommendation system for adaptive e-learning using Semantic Net. International Journal of Computer Applications, 63 (7), 19-23 (2013)
6. Tang, T., McCalla, G.: Smart recommendation for an evolving e-learning system- Architecture and experiment. International Journal on E-learning, 4(1), 105-129 (2005)
7. Chen, C. M., Lee, H. M., Chen, Y. H.: Personalized e-learning system using Item Response Theory. Computers & Education, 44(3), 237-255 (2005)
8. Felder, R.M., Soloman, B.A.: Index of Learning Styles Questionnaire. North Carolina State University (1996). Available at http://www.engr.ncsu.edulearningstyles/ilsweb.html.