Liu, L., Liang, Y., & Li, W. (2017). Dynamic assessment and prediction in online learning: Exploring the methods of collaborative filtering in a task recommender system.
International Journal of Technology in Teaching and Learning, 13(2), 103-117.

Dynamic Assessment and Prediction in Online learning: Exploring the Methods of Collaborative Filtering in a Task Recommender System

Leping Liu
University of Nevada, Reno, USA
Ying Liang
Southwest University, China
Wenzhen Li
University of Nevada, Reno, USA

Dynamic learning is a featured learning style in the second decade of the 21st century, emphasizing on the processes of individual or collaborative learning. Conducting dynamic assessment becomes critical to achieve the goals of learning. As learning becomes more individualized, online learning platforms have embedded a task recommender system to identify and predict individual needs, and to recommend different exercises or tasks for each learner so they can gain knowledge more efficiently. This article introduces the logistics of a task recommender system that can be used to perform dynamic assessment, and the methods of the learner-based collaborative filtering in a task recommender system, followed by an example that demonstrates the methods and procedures to assess the recommender effect.

Keywords: dynamic learning, dynamic assessment, dynamic prediction, recommender system, collaborative filter, online learning

INTRODUCTION

Dynamic learning has become a featured learning style of the 21st century's learners (Quellmalz, et al., 2012; Saavedra, & Opfer, 2012), where learning occurs non-linearly, from multiple dimensions, as process-focused rather than state-focused, and with open-ended solutions or directions (Liu & Maddux, 2005; Rotherham & Willingham, 2010). As dynamic learning emphasizes more on the processes of learning, conducting dynamic assessment becomes more critical, and selecting efficient assessment tool has

---

Leping Liu is a Professor, Information Technology and Statistics, at the College of Education, University of Nevada, Reno. Ying Liang is a Lecturer at the School of Computer and Information Science, Southwest University, China. Wenzhen Li is an Instructional Design Specialist, at the University of Nevada, Reno. Leping Liu can be reached at liu@unr.edu

been a challenge to our educators (Liu, 2017; Silva, 2009). Now more and more network or online teaching platforms have embedded a task recommender system to identify individual needs, to suggest and provide differentiated exercises or tasks for each learner so they can gain knowledge more efficiently (Koren & Bell, 2011; Liang, & Chen, 2014). Technically, the functions of a recommender system do enable educators to conduct dynamic assessment for online learning (Liu, 2017, Liu, Gibson & Ifenthaler, 2018). The purpose of this article is to introduce the methods of using a task recommender system to perform dynamic assessment in an online learning environment.

In the following sections we will present: (a) dynamic learning and the procedures of dynamic assessment, (b) the logistics that a task recommender system can be used to perform dynamic assessment in online learning, (c) two methods of collaborative filtering in a task recommender system, and (d) an example that demonstrates the methods and procedures to assess the recommender effect.

DYNAMIC LEARNING AND DYNAMIC ASSESSMENT

DYNAMIC LEARNING

The four features of dynamic learning, (a) nonlinear, (b) multiple dimensional, (c) process-focused, and (d) open-ended, are based on a dynamic design model proposed by Liu and Maddux (2005). When learning activities are divided into the very basic units or operational tasks and procedures, they can be framed within the dynamic design model (Liu, 2017):

Nonlinear learning simply means that learning tasks are not organized in a linear manner. All required learning tasks can be arranged into a net map where students can work on several tasks simultaneously. The benefit of this nonlinear approach is that the learning of the knowledge structure and basic skills or details can occur at the same time.

Multiple Dimensions in learning is about learning contents, instruction delivery methods, or level of communications. Students can learn from different information resources (e.g., course materials, web sources, from online learning community, or social networking groups), which come in different formats (e.g., text, audio, or video). Instructions and communications can be conducted face-to-face, online, in individual or group. Course work can be written assignments, online discussions, or team projects. More important, with the nonlinear approach, the learning tasks or activities on different dimensions can be matched or paralleled, and then completed at each dimension as well.

Process-Focused learning indicates the details and logic flow of the knowledge to be learned, and the progresses the learner has made. This is where the researchers or instructors obtain the dynamic data of learning, and conduct the dynamic assessment. Examples can be an on-going portfolio, or a series of projects that reflect each stage of a theory or certain model.

Open-Ended learning outcomes or products vary with different features. In the field of instructional technology, a typical product could be the development of a personalized model. For instance, a technology integrated learning model, an assessment model, or an online learning system design model. Such self-developed models would open the directions for students' further research and practice. Another example of an open-ended learning product could be a publication or an initial research agenda based on which students can develop more studies.

DYNAMIC ASSESSMENT

Dynamic assessment originally was a product from Lev Vygotsky's research. Highly interactive and process-oriented, it emphasizes the learning process and seeks to identify the skills that an individual child possesses as well as their learning potential (Datnow & Hubbard, 2014; Haywood & Lidz, 2007; Sternberg, & Grigorenko, 2002). A classic application is to use dynamic assessment to study the zone of proximal development in Vygotsky's analysis of learning and instruction (Bozhovich, 2009; Burkitt, 2006; Chaiklin, 2003; Obukhova & Korepanova, 2009).

In the context of dynamic learning, dynamic assessment is a set of process-focused activities performed by the instructor or the learning system. During the learning process, data on student learning can be collected in a dynamic way, from the non-linear learning activities, in all the dimensions, and at any point on a continuous timeline. Data analytics will then be processed constantly to produce dynamic assessment results. The results can be used to identify the potential problems or weaknesses in student learning, and the level of knowledge and skills students have achieved at the time. Based on the assessment results, dynamic "predictions" and "recommendations" can be made constantly to direct student learning (Liu, Li, & Scherer, 2016; Liu, 2017, Liu, Gibson, & Ifenthaler, 2018).

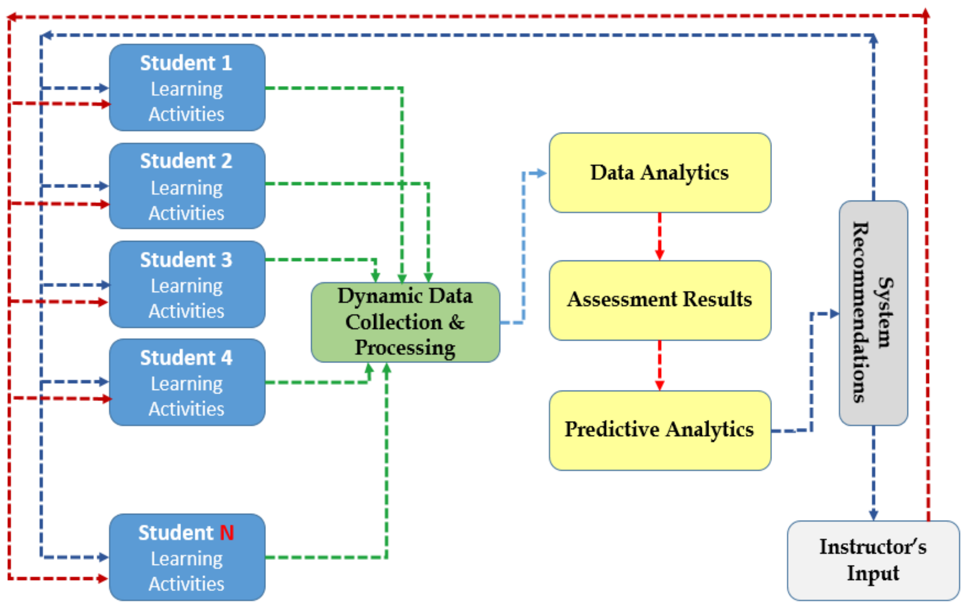

Figure 1. Procedures of Dynamic Assessment

Figure 1 illustrates the procedures and the cycle of dynamic assessment. Researchers and educators have been exploring the methods and technology tools to perform dynamic assessment such as a social network, or any cyberinfrastructure systems (Dawson, Macfadyen, Lockyer, & Mazzochi-Jones, 2011; Gibson & Ifenthaler, 2017). This article introduces a *recommender system* embedded in the online learning platform. It is a tool to collect and process dynamic data, and make dynamic prediction and recommendations to the learners.

RECOMMENDER SYSTEM FOR DYNAMIC ASSESSMENT

WHAT IS A RECOMMENDER SYSTEM?

A recommender system is a personalized information filtering technology used to identify a set of content items or features of interest to a certain user (Deshpande, & Karypis, 2004), and recommends useful information or suggests strategies for users to achieve their apparent goals (Bullet, 2004). Currently, applications of *recommender system* range from e-business, Internet services, to e-learning and many other fields in industry and education (Koren & Bell, 2011; Linden, Smith & York, 2003).

A very common example in business is the use of an e-commercial recommender system. When you search and purchase a book online from Amazon, for example, the recommender system will then provide you a list of suggested books with relevant themes. The recommender system generates this list by analyzing your search behaviors (the books you looked at the time), search history (what you searched before), purchase history, and any information that shows the nature or field of your interest (e.g., a biology professor in a university, or a software engineer in an IT firm). The recommender system also analyzes the search/purchase behaviors of other buyers who purchased the similar books to generate the suggested list. Sometime, from the suggested list you may find one that is definitely more accurately targeting at your need than the original one you searched.

In the field of education, more and more online or network teaching platforms use recommender system to determine learners' individual demands in learning, and help them gain knowledge more efficiently (Liang & Chen, 2014). Learners' potential and individualized learning-demands can be identified from searching, analyzing, summarizing and generating their behavior patterns (Su, Chang, Chiu, & Hsieh, 2015). In a teaching platform such as an instructional website or online learning environment, learners can take an initiative to finish the tasks assigned by the instructors. The system will then automatically evaluate the completion of the task and count correlated data. Generally, tasks assigned to the learners by the teaching platform are mostly the same, either selected by the instructors or randomly drawn from task library. As educators all know, different learners have different knowledge background and comprehension ability as well as different learning style. Assigning the same tasks to every learner may cause them to lose interest or feel less motivated in learning. In the era where individualized learning is emphasized, using a recommender system to further improve the learning efficiency has thus become a new goal when designing a teaching platform.

USING A TASK RECOMMENDER SYSTEM IN DYNAMIC ASSESSMENT

A task recommender system can be used to perform dynamic assessment through the following procedures (as shown in Figure 1):

1. collecting and processing dynamic data,
2. conducting data analytics,
3. producing assessment results,
4. performing predictive analytics, and
5. making recommendations.

Information from each learner on each learning activities or tasks can be collected and analyzed constantly at the time they are completed. The predictions and recommendation of the "next" learning task/activity that fits each learner's level and condition will be provided.

COLLABORATIVE FILTERING AND THE PROBLEM OF DATA SPARSENESS

Collaborative filtering is a technique commonly used in a recommender system to build personalized recommendations, and algorithms are used to make automatic predictions about a user's interest by compiling preferences from several other users. A common problem in traditional collaborative filtering is data sparseness; a recommender system could not provide appropriate recommendations unless necessary and completed information from the user input is available (Lee, Battle, Raina, & Ng, 2007). For example, the problem of data sparseness occurred in e-business recommender system mostly because too much commodity information is available, but the users may only grade a few commodities (Li, Yang, & Xue, 2009).

Similarly, data sparseness exists in task recommender system in a teaching platform, when the number of learners who completed the exercises or tasks is too small, or when there is no sufficient information from the students while starting the system. In addition, if the learner's performance score is the only variable to be analyzed, the trend in task completion calculated by the system may not be predictive. Comparatively, the recommender system may generate a more accurate trend with additional information such as the time spent to complete the task, or the order each learner chose to complete the tasks.

In fact, to collect, process, and analyze dynamic data for dynamic assessment, the filtering method for an e-business recommender system cannot be used as it is in an e-learning recommender system. In the next sections, we will introduce two methods of learner-based collaborative filtering in a task recommender system for an online teaching platform:

1. *Traditional* collaborative filtering method (Collaborative Filtering with *Scores*): this method only uses the learners' performance scores to calculate similarity degree and to predict scores.
2. *Improved* collaborative filtering method (Collaborative Filtering with *Scores* and *Time*): this method uses the learners' performance scores as well as the time they spent to complete a task to calculate similarity degree and to predict scores.

TRADITIOANAL LEARNER-BASED COLLABORATIVE FILTERING

The traditional method of *learner-based collaborative filtering* can be used to predict targeted learners' scores on particular exercises or tasks. The prediction is based on the analysis of the scoring records from other learners. Through the collaborative filtering, a recommender system analyzes learners' common scoring to calculate the *similarity* among the learners, and then predicts targeted learners' scores using *similar learners*' scoring. Based on the similarity of learners, new exercises or tasks will be recommended. The following are the procedures of the traditional learner-based collaborative filtering (Koren & Bell, 2011; Kovács, 2010; Mohan, 2013).

CONFIRMING THE SIMILAR LEARNERS

The key to confirm the similar learners is to count the similarity of the learners (Mohan, 2013). One way to calculate the similarity of learners is to use the collaborative filtering recommender based on the *exercises*. To count the *similarities of two exercises* is to find out the learners who did and scored the two exercises at the same time, and then use similarity calculation method to calculate the similarity between the exercises. The other way is to use collaborative filtering recommender based on the *learners*, it counts the *similarity of the two learners* who scored the same exercises.

There are many ways to calculate the similarity. As proposed by Deshpande and Karypis (2004), the two mainstream methods widely used in the field are:

1. *Pearson Correlation*: Pearson correlation calculation is used to measure the similarity among the learners or the similarity among the tasks. It examines linear correlation of two variables (for example, they can be measures on two tasks, or two learners). At the same time, it also considers the scoring trends of learners so more accurate similarity results can be obtained (Kovács, 2010).
2. *Cosine Similarity*: Usually the learners' scoring to the exercises can be described with an A by B (A * B) learner exercise matrix. "A" stands for the number of the learners, and "B" stands for the number of the exercises. Then the similarity of the

two learners can be obtained by counting the Cosine value of dimension vectors constructed by the correspondent exercises scoring (Salton & McGill, 1983).

In real life, different learners have their own scoring trends, one may tend to have high scores, and some others may be likely to have low scores. Vector Cosine similarity calculation doesn't consider the diversity of learners' scoring trends. In order to solve this problem, an adjusted Cosine calculation method is proposed, combined with the feature of Pearson similarity calculation method (Koren & Bell, 2011).

PREDICTING SCORES

After getting the similarity of learners, we can predict the targeted learners' scores to the task. Score-predicting is the key procedure in collaborative filtering recommender; different predicting method will lead to different prediction results and recommendation effects. Three score-predicting methods are often mentioned and compared as seen in the literature:

1. *Averaged prediction* is the most simple and clear method to calculate predicted scores of a targeted task, based on the scores to the task given by all the similar learners. The defect is that it considers neither the targeted learners' scoring trends, nor the scoring trends of similar learners. It doesn't reflect the similarity degree of the learners, thus there is a relatively larger error in the prediction results. (Zemke, 2003).
2. *Simple weighted summation average* considers the learners' similarity on the basis of *averaged prediction*. This method is more complex than the first one, as it also considers the similarity degree, or the distance of similarity, among the learners. The more similar the learners are, the more influence the scores of the targeted task will have on the final predicted scores, and thus the predictive results would have fewer errors. (Frederick, 2006).
3. *Complex weighted summation average* considers both the targeted and similar learners' scoring trends on the basis of the *simple weighted summation average*. At present, this prediction method in recommender system is the most widely used. Unlike the second method that uses only the absolute scores without considering the scale of learners' scoring, the *complex weighted summation average* method conducts calculations with scoring deviation, and its prediction results have the least errors among the three methods. (Radicchi, Ramasco, & Fortunato, 2011).

RECOMMENDING TASKS

Recommender system will recommend further tasks to the targeted learners in accordance with the predicted results, usually adopting Top-N recommender algorithm (Sarwar, Karypis, Konstan, & Riedl, 2001). In the Top-N recommender algorithm, the predicted scores are proportional to the learning trend of the learners. The higher the scores, the more likely the learners are to finish the task. Then the recommender system can recommend those tasks scored as Top-N from the unfinished tasks (Deshpande, & Karypis, 2004; Kang, Peng, & Cheng, 2016).

AN IMPROVED LEARNER-BASED COLLABORATIVE FILTERING

Knowing how traditional collaborative filtering works, in this section, we will introduce an improved learner-based collaborative filtering method that combines two types of learners' behavior data (*score* received on a task and *time* spent to complete the task) to count the similarity of learners. Based on the similarity of learners, the recommender system will then complete the collaborative filtering to make predictions and recommendations.

SCORE AND TIME ON TASK

In an online learning system, the interactions between the learners and tasks have many ways of expression, such as the status of task, and task completion. The task completion in online learning is quite different from the purchase situation in an e-business network. The task completion in an e-business system has only two results: “buy” or “not buy” (Herlocker, Konstan, Terveen, & Riedl, 2004). However, after a learner completes a task in an online learning system, there will be a correspondent score of comprehensive learning behaviors. These behaviors are related to whether the task is interesting to the learners. If the learners finish the task, the task is absolutely or mostly meet the expectation of the learners, or the task can meet the highest expectation value among the many expectations of the learners. Whether or not the learner can finish the task, the scores of completion, and the time he/she takes should all be considered in recommender system (Karydi & Margaritis, 2016).

Traditionally, the recommender system in an online learning system (e.g., an instructional website, or an online teaching platform) only uses the task score records to complete the recommendations. However, some task completion packets may be sparse, as some learners did not exit the system after completing the task (or exit the system before completing the task), which leads to the difficulty in counting similarity of the learners, and hence such data cannot accurately reflect the similarity condition of the learners (Adomavicius, & Tuzhilin, 2005). Additionally, only using the learners’ task score records may not be able to completely reflect the level the learners mastered the knowledge.

For example, Jane and Bob both scored 90 points in completing a Word exercise, but Jane spent 20 minutes on it, and Bob spent only 10 minutes. Though they have the same score, they may have different levels of mastering the knowledge. The situation might be that Jane completed the exercise with some online help or aids from others, but Bob’s completion is straightforward. In this case Jane and Bob may not be similar learners. Therefore, when we record the learners’ scores for the task, we may also include the time spent on the task. Combining the two types of learners’ behavior data to count the similarity of the learners, the recommender system can then complete the collaborative filtering to make predictions (Ali & Stam, 2004; Bellogín, Cantador, & Castells, 2013).

DATA RECODING

We need to recode the learner-task raw data obtained from the online learning system into the data code that the recommender system can process and calculate in order to make recommendations later. The learner-task data includes (a) *score* data: scores on the tasks, and (b) *time* data: time spent to complete the tasks.

Formula (1) is the data recoding-criteria of *score* data. $S_{\mathrm{u,c}}$ stands for the code of a learner’s score on a task, and it also represents the conditions of task completion. If the learner answers less than 60% questions correctly; he or she gets a code of 0. If the learner answers 60%, or more than 60% but less than 90%, he or she gets a code of 1. A learner who answers more than 90% will get a code of 2.

$$S_{\mathrm{u,c}} = \begin{cases} 0, & \text{Learners u correct rate} \quad \mathrm{c} < 60\% \\ 1, & \text{Learners u correct rate} \quad 60\% \leq \mathrm{c} < 90\% \\ 2, & \text{Learners u correct rate} \quad \mathrm{c} > 90\% \end{cases} \qquad (1)$$

Formula (2) is the data recoding-criteria of *time* data. $T_{U,C}$ is used to represent the code of the time a learner spent on a task. Each task has a time limit. If the time a learner spent on a task is more than 90% of the time allowed, he or she gets a code of 0, if the time is more than 60%, but less than 90%, a code of 1 is given. If the time is less than 60%, he or she gets a code of 2.

$$T_{U,C} = \begin{cases} 0, & \text{Learners u time rate} \quad c > 90\% \\ 1, & \text{Learners u time rate} \quad 60\% \leq c < 90\% \\ 2, & \text{Learners u time rate} \quad c < 60\% \end{cases} \qquad (2)$$

SIMILARITY CALCULATION

If the similarity status of the learners is more accurately calculated, the knowledge consolidation status of the learners will be more accurately predicted. Then better recommender effects could be generated, and the online learning system's service may better meet with the needs and interests of the learners. In our methods, we revised the traditional method of similarity calculation, by linearly combining Cosine similarity degree based on *score* record and Pearson similarity based on *time* record. Formula (3) is used to measure the similarity status of two learners.

$$F_{U,C} = \alpha * F_{score} + (1 - \alpha) * F_{\text{time}} \qquad (3)$$

In formula (3), $F_{U,C}$ stands for the final similarity degree, F_{score} stands for the Cosine similarity degree based on *score* record, and F_{time} stands for Pearson similarity degree based on *time* record. α is the weighting factor, the higher the α value, the more influence Cosine similarity result based on *score* record between the learners would have on the final similarity status of the learners, and the less influence Pearson similarity degree based on *time* record would have on the final similarity status (Kovács, 2010; Koren & Bell, 2011). The weighting factor α, the score-time ratio in formula (3), usually is determined by the instructors according to their teaching experiences. Hereby α value of 0.8 is set and used in the pilot test described in the next section.

PREDITION METHOD

After confirming similar learners, the recommender system will then predict the completion status of the targeted learners to the task. Again, the three commonly used predictive methods are: *averaged prediction*, *simple weighted summation average*, and *complex weighted summation average* (Radicchi, Ramasco, & Fortunato, 2011). In the following pilot test, we started with the *simple weighted summation average* method. It is expected that using this improved collaborative filtering method would achieve a relatively better predictive accuracy and recommender effect.

AN EXAMPLE: ASSESSING THE RECOMMENDER EFFECT

In this section, we will demonstrate the methods and procedures to determine and assess the recommender effect of a learner-based collaborative filtering system. A pilot test was conducted to explore to what extent using the improved collaborative filtering method in the online teaching system, when data sparseness exists, can achieve relatively better predictive accuracy as suggested in the literature (Bifet, Morales, Read, Holmes, & Pfahringer, 2015; Geetika, 2014).

The pilot test was carried out in the *exercise* section of the online teaching system for a course *Fundamentals of Computers* offered by a southwest university in China. A small portion of data was used to pilot the recommender system, while the problem of data sparseness exists.

THE DATA SET

The sample data set was derived from the data in the online teaching platform of the course *Fundamentals of Computers*. The Office of Teaching Affairs in the university added the *exercise* component in the teaching platform of the course to help students practice and master the contents. The rear end of the platform recorded learners' data, including learners' demographic data, task completion records, browsing track records, score data, and time data. To verify the algorithm of the collaborative filtering, two regular classes were randomly chosen, and data from students in the two classes who did the exercises were randomly chosen. For the purpose of the pilot test, only learners' *score* data and their *time* data were used (Hunt, 2015; Indranath, 2016).

The original data of *score* and *time* totally included 61 learners, 32 exercises, and 410 task completion records. Data screening revealed some missing and incomplete data. As during the test, the client-server had some unexpected crash interruption, so some data set only had starting time without ending time, which could not be recoded as described in formula (2). Therefore, the following data screening procedures were done:

1. Eliminating the data that had only starting time but no ending time.
2. Confirming valid learners. Valid learners were those with completed scores in both task scores and time records.
3. Confirming valid tasks. Valid tasks were those that had been finished more than 3 times. Tasks that were completed less than 3 times were also eliminated.

After the data screening, the data set of *score* and *time* included 60 learners, 30 tasks, and 403 records of learner performance for further analysis.

PROCEDURES

Again, to evaluate whether using the improved collaborative filtering method can more accurately reflect the similarity degree of the learners, and provide higher quality recommendation, the two collaborative filtering methods described above were used in the pilot test:

1. *Traditional* collaborative filtering method: Collaborative Filtering with *Scores* (CF-S).
2. *Improved* collaborative filtering method: Collaborative Filtering with *Scores* and *Time* (CF-ST).

The assessment of the recommender effect was performed with the following procedures:

1. Selecting performance score data from valid learners and their valid tasks, using formula (1) to recode and construct the *score* data.
2. Selecting performance time data from valid learners and their valid tasks, and using formula (2) to recode and construct the *time* data.
3. Using vector Cosine similarity degree calculation and formula (3) to calculate the similarity degree among the learners.
4. Using the *simple weighted summation averaging* to test task completion status of the learners.

Different data values were selected with different random sparseness levels. Then, prediction accuracy produced with the two methods (*traditional* CF-S, and *improved* CF-ST) were compared. Four tests were undertaken in four different levels of sparseness (with

96.67%, 96.11%, 95.56% and 95.00% of sparsity). The results from the tests are reported in Table 1.

ASSESSING THE RECOMMENDER EFFECT

In the test, *Mean Absolute Error* (MAE) and *Root Mean Squared Error* (RMSE) were used as the "index" to measure the accuracy of the predicted result. MAE measures the average magnitude of the errors in a set of predictions. It is the average over the test sample of absolute differences between prediction and actual observations (Quellmalz, Thontteh, & Chen, 2012). RMSE is the measure of the differences between values (sample or population values) predicted by a model or an estimator and the value observed (Willmott & Matsuura, 2006). They are two of the most commonly used measurements in recommender systems. They are both calculated to compare the diversity between the learners' actual scores and the scores predicted for each of them by the recommender system. In the comparison of actual scores and predicted scores, *the less the value of MAE* or *RMSE, the more accurate the prediction is, and the better recommender effect is achieved* (Myttenaere, Golden, Grand, & Rossi, 2015; Willmott & Matsuura, 2005).

Table 1. *The Pilot Test Data*

Test	N_u	N_s	N_r	Sparsity	MAE		RMSE	
					CF-S	CF-ST	CF-S	CF-ST
1	60	30	60	96.67%	0.7623	0.7371	0.8681	0.8574
2	60	30	70	96.11%	0.7029	0.7007	0.825	0.8209
3	60	30	80	95.56%	0.6938	0.6622	0.8012	0.8051
4	60	30	90	95.00%	0.6593	0.6277	0.776	0.7627

In Table 1, N_u stands for the number of *learners* in the pilot test; N_s stands for the number of *tasks* selected from the learners' performance data; and N_r stands for the number of *performance* records on the tasks selected from the data set. For example, in test 1, 60 performance records on 30 learning tasks from 60 learners are randomly selected with the data sparsity of 96.67%. The data process and analysis were conducted with Weka, a suite of machine learning software for data mining tasks. Table 1 also shows the MAE values of CF-ST and CF-S under different sparseness status, and the RMSE values of CF-ST and CF-S under different sparseness status as they are often reported in the literature (Treerattanapitak & Jaruskulchai, 2012).

The calculation formula of data sparseness is as follows.

$$\text{Sparsity} = 1 - \frac{N_r}{N_u * N_S} \qquad (4)$$

The same as in Table 1, in Formula (4) N_r stands for the number of records of performance in the data set; N_u stands for the number of users/learners in the data set. N_s stands for the number of tasks of user performance in the data set.

As shown in Table 1, under the same data sparseness, the recommender effect of CF-ST when using MAE and RMSE to measure is relatively better than the effect of CF-S with the same measurements. For example, in test 1, with the data sparsity of 96.67%, the MAE value of using CF-ST method (0.7371) is smaller than that of using CF-S method (0.7623), and the RMSE value of using CF-ST method (0.8574) is smaller than that of using CF-S method (0.8681).

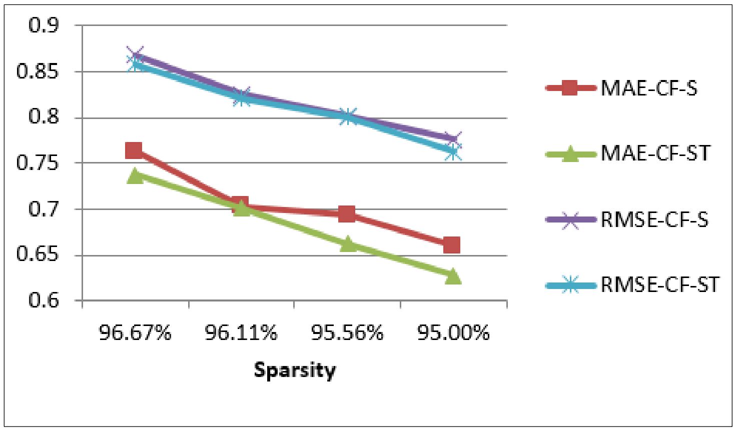

Figure 2. Value of MAE and RMSE from the Pilot Test

Figure 2 shows the status of MAE of CF-ST and CF-S, as well as the status of RMSE of CF-ST and CF-S: (a) the values of MAE and RMSE of CF-ST and CF-S decreases with less data sparseness, and (b) under the same sparseness, the values of MAE and RMSE of CF-ST are *lower* than that of CF-S correspondingly. Although the index differences between the two methods are very small (e.g., 0.7371 versus 0.7623, and 0.8574 versus 0.8681, as described above), the results from improved collaborative filtering method did demonstrate a trend to better reflect the similarity status among the learners. The improvement of the effect is also related with data sparseness.

SUMMARIES AND DISCUSSIONS

In summary, this article introduces the methods of collaborative filtering in a recommender system that can be used to perform dynamic assessment in the context of dynamic learning. Two methods are presented: (a) the traditional collaborative filtering with scores (CF-S), and (b) the improved collaborative filtering with scores and time (CF-ST). The pilot test with the data set from the online learning platform demonstrates: (a) the procedures of data selection, screening, and recoding, and (b) the comparison of the CF-S and CS-ST methods regarding the accuracy of their prediction using *Mean Absolute Error* (MAE) and *Root Mean Squared Error* (RMSE). The results indicate that the improved collaborative filtering with *score* and *time* (CF-ST) has the potential to achieve a better recommender effect.

The methods described in the pilot test may have some limitations. First, in the data coding, *scores* and *time* are divided into three categories, which are preliminary classifications. The recommendations may be more accurate if more layers are used for the recode. Second, in the two behavioral data, the *score* and *time* ratio were 0.8 to 0.2, which is based on the instructors' classroom practice and experiences. Different proportion of the ratio may be tested in further studies.

As demonstrated in Figure 1, the central procedures of *Dynamic Data Collection and Processing, Data Analytics, Assessment Results, Predictive Analytics* and *Making Recommendations* can be performed with the learner-based collaborative filtering in task recommender system. However, using such task recommender system to perform dynamic assessment for online learning will need careful design starting from the online learning design, learning materials and assessment criteria, and data analytics strategies, to the interactions between the instructor and students, between the system and learners, and between the system and instructor. It is also critical that the learning tasks and procedures are designed in the way that dynamic data of students' performance or learning behaviors can be collected dynamically. Technical support is another key factor in the design of the task recommender system.

Noticed that in this article we used common language to describe the recommender system and the collaborative filtering methods, and avoided to use too many mathematics or computer science terms. We only conceptually explained the basic principles and procedures. More details could be found in the reference articles cited for those concepts, methods or procedures. Also, the algorithms for the recommender system are not introduced as they are beyond the scope of this article, although they are the most important core contents when discussing the recommender effect. Educators who are interested in this may continue the exploration from related articles in computer science and information system journals.

Further studies are to be conducted (a) with larger size of data, (b) with more other behavioral data such as the rate of corrected answers, or the frequency of learning activities, and (c) to explore the probability that, or to what extent, different sparsity could result in inaccurate recommendation. Comments and suggestions are appreciated.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering, 17*(6), 734-749. DOI:http://dx.doi.org/10. 1109/TKDE.2005.99

Ali, K., & Stam, W. V. (2004). TiVo: Making show recommendations using a distributed collaborative filtering architecture. In Proceedings of the *10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04).* pp.394--401. DOI:http://dx.doi.org/10.1145/1014052.1014097

Bellogín, A., Cantador, I., & Castells, P. (2013). A comparative study of heterogeneous item recommendations in social systems. *Information Sciences*, *221*, 142-169. DOI:http://dx.doi.org/10.1016/j.ins.2012.09.039

Bifet, A., Morales, G. D. F., Read, J., Holmes, G., & Pfahringer, B. (2015). Efficient online evaluation of big data stream classifiers. In Proceedings, *21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 59-68.

Bozhovich, E. D. (2009). Zone of Proximal Development: The Diagnostic Capabilities and Limitations of Indirect Collaboration. Journal of Russian & East European Psychology, 47(6), 48-69.

Bullet, D. (2004). Provider of usability and Web design services. Retrieved May 20, 2004, from http://www.usabilityfirst.com/glossary/term_962.txl

Burkitt, E. (2006). Zone of proximal development. In *Encyclopaedic dictionary of psychology*. Retrieved from Credo Reference database.

Chaiklin, S. (2003). The Zone of Proximal Development in Vygotsky's analysis of learning and instruction. In Kozulin, A., Gindis, B., Ageyev, V. & Miller, S. (Eds.) *Vygotsky's educational theory and practice in cultural context*. (pp. 39-64). Cambridge: Cambridge University.

Datnow, A., & Hubbard, L. (2014, April). *Teachers' use of assessment data to inform instruction: Lessons from the past and prospects for the future*. Paper presented at the Annual Meeting of the American Educational Research Association, Philadelphia, PA.

Dawson, S., Macfadyen, L., Lockyer, L., & Mazzochi-Jones, D. (2011). Using social network metrics to assess the effectiveness of broad-based admission practices. *Australasian Journal of Educational Technology, 27*(1), 16-27.

Deshpande, M., & Karypis, G. (2004). Search results: Item-based Top-N recommenddation algorithms. *ACM Transcactions Information Systems*, *22*(1), 143-177.

Frederick, J. (2006). Statistical methods in experimental physics (2nd ed.). Singapore: World Scientific.

Geetika, S. (2014). Survey on classification methods using WEKA, *International Journal of Computer Applications*, *86*(18), -19. DOI: 10.5120/15085-3330

Gibson, D. C., & Ifenthaler, D. (2017). Preparing the next generation of education researchers for big data in higher education. In B. Kei Daniel (Ed.), *Big data and learning analytics: Current theory and practice in higher education* (pp. 29-42). New York, NY: Springer.

Haywood, H. C., & Lidz, C. S. (2007). *Dynamic Assessment in Practice: Clinical And Educational Applications*. Cambridge, England: Cambridge University Press.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems, 22*(1), 5-53.

Hunt, L. (2015). Missing data imputation and its effect on the accuracy of classification. In *Conference of the International Federation of Classification Societies, Book of Abstracts,* pp.105-105.

Indranath, C. (2016). Data mining: An experimental approach with WEKA on UCI Dataset, *International journal of computer applications*, *138* (13), 23-28, DOI: 10.5120/ijca2016909050

Kang, Z., Peng, C., & Cheng, Q. (2016). Top-N recommender system via matrix completion. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), (pp. 179-185). Association for the Advancement of Artificial Intelligence (www.aaai.org).

Karydi, E., & Margaritis, K. (2016). Parallel and distributed collaborative filtering: A survey. *ACM Computing Surveys, 49*(2), Issue-in-Progress Article No. 37.

Koren Y, & Bell, R. (2011). Advanced in collaborative filtering. *Recommender Systems Handbook (1st edition), Springer*, pp.145–186.

Kovács, B. (2010). A generalized model of relational similarity. *Social Networks*, *32*(3), 97-211.

Lee, H., Battle, A., Raina, R., & Ng, A. Y. (2007). Efficient sparse coding algorithms. *Proceedings of 19th Annual Conference Neural Information Processing Systems*, pp.801-808.

Li, B., Yang, Q., & Xue, X. (2009). Can movies and books collaborate? Cross-domain collaborative filtering for sparsity reduction. *21st International Joint Conference on Artificial Intelligence*, Pasadena, United States, 11-17 Jul 2009, pp. 2052-2057.

Liang, Y., & Chen, H. (2014). The Research of video resource personalized recommendation system based on education website. *The 9th International Conference on Computer Science & Education*. SaP8.13/898.

Linden G, Smith B, & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing, 7*(1), 76-80.

Liu, L. (2017) An initial model of dynamic instructional design and learning analytics: The power of dynamic assessment and prediction, keynote speech and in *Proceedings* of AECT/SICET, Jacksonville, FL, November 2017.

Liu, L., Gibson, D., & Ifenthaler, D. (2018). A model of dynamic instructional design and dynamic assessment. Panel accepted for 2018 SITE International Conferences, Washington DC.

Liu, L., Li, W., & Scherer, R. (2016). Technology in counseling education: Case analysis and a dynamic course design. . *International Journal of Technology in Teaching and Learning, 12*(2), 123-138.

Liu, L., & Maddux, C. (2005). Influences of course design on student evaluations: An initial logistic prediction model. *Journal of Excellence in College Teaching, 16*(1),125-148.

Mohan, T. M. (2013). Recommended system for neighborhoo-based collaborative filtering algorithm using Pearson correlation. *International Journal of Electronics Communication and Computer Engineering*, *4*(6), 1627-1632.

Myttenaere, A. D., Golden, B., Grand, B. L., & Rossi, F. (2015). Using the mean absolute percentage error for regression models. *Proceedings of the 23th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning* (ESANN 2015). Retrieved from: https://arxiv.org/abs/1506. 04176

Obukhova, L. F., & Korepanova, I. A. (2009). The Zone of Proximal Development: A Spatiotemporal Model. *Journal of Russian & East European Psychology*, 47(6), 25-47.

Pontius Jr., R. G., Thontteh, O., Chen, H. (2008). Components of information for multiple resolution comparison between maps that share a real variable. Environmental and Ecological Statistics. 15, 111-142.

Quellmalz, E. S., Timms, M. J., Buckley, B. C., Davenport, J., Loveland, M., & Silberglitt, M. D. (2012). 21st century dynamic assessment. In M. C. Mayrath (Ed.), *Technology-based assessments for 21st century skills: Theoretical and Practical Implications from Modern Research* (pp. 55-90). Charlotte, NC: Information Age Publishing, Inc.

Radicchi, F., Ramasco, J. J., & Fortunato, S. (2011). Information filtering in complex weighted networks. *Phusical Review, 83*, 046101-1-8. DOI: 10.1103/PhysRevE. 83. 046101

Rotherham, A. J., & Willingham, D. T. (2010). 21st-century skills: Not new, but a worthy challenge. *American Education, 34*(1), 17-20.

Saavedra, A. R., & Opfer, V. D. (2012). Learning 21st-century skills requires 21st-century teaching. *The Phi Delta Kappan, 94*(2), 8-13.

Salton, G., & McGill, M. J. (1983). Introduction to modern information retrieval. New York: Mc Graw Hill.

Sarwar, B, Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. Copyright is held by the author/owner. WWW10, May 1-5, 2001, Hong Kong. ACM 1-58113-348-0/01/0005, p285-295.

Silva, E. (2009). Measureing skills for 21st-century learning. *Phi Delta Kappan*, *90*(9):630-634. DOI:10.117 7/003172170909000905.

Sternberg, R.J. & Grigorenko, E.L. (2002). Dynamic testing: The nature and measurement of learning potential. Cambridge (UK): University of Cambridge.

Su, C. Y., Chang, J., Chiu, T., Hsieh, T. (2015). A Cluster-based personalized item recommended approach on the educational assessment system. http://online-journals.org/index.php/i-jet/article/view/4807, *10* (5).

Treerattanapitak, K., & Jaruskulchai, C. (2012). Exponential fuzzy C-Means for collaborative filtering, *Journal of Computer Science and Technology*, *27*(3), 567-576.

Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance, Published in CR Vol. 30, No. 1. Online publication date: December 19, 2005, Print ISSN: 0936-577X; Online ISSN: 1616-1572

Willmott, C., & Matsuura, K. (2006). On the use of dimensioned measures of error to evaluate the performance of spatial interpolators. *International Journal of Geographic Information Science. 20*, 89-102.

Zemke, S. (2003). Data mining for prediction: Financial series case. Dissertation, Department of Computer and Systems Science. The Royal Institute of Technology.

ACKNOWLEDGMENTS

This research is supported by the China Scholarship Council and Fundamental Research Funds for the Central Universities (XDJK2014C143).