On Supporting Game-Based Learning via Recommendations

Aytuna Yamaç, Mehmet Yamaç and Kostas Stefanidis
Tampere University, Finland

aytuna.yamac@tuni.fi

Abstract: Over the last two decades, game-based learning has gained increasing popularity. In today's world, teachers are expected to utilize technological tools such as digital games as learning aids. Despite the multitude of studies examining the benefits of game-based learning, finding the most convenient game for a particular teaching purpose can be a challenging task given the vast number of similar games that are available on the market. With this study, we aim to provide teachers with a recommendation system that will assist them in selecting appropriate games from all the web-based game materials available. A key theoretical premise behind this work is to examine teaching from the perspective of teachers to develop their ability to teach. The purpose of this study is to develop a recommendation system that will assist teachers in selecting educational games based on the subjects they teach, that will be both personalized and use the experience of other researchers at the same time. We propose a system that utilizes the latest developments in signal processing and machine learning, specifically the tensor completion method. This is a machine learning technique from the family of collaborative filtering methods that fills in missing values in a dataset by analyzing its existing patterns. According to our knowledge, this is the first study to modify a collaborative filtering approach to develop a recommendation system for game-based learning. Whenever a teacher requires a recommendation, the method leverages other users' ratings on games, while also considering the teacher's previous experience with the system. Accordingly, the system selects the best games for users based on their previous preferences as well as the experiences of other users. It appears that this system has a reasonable chance of working properly without excessive training samples, which could not be achieved through other advanced machine-learning techniques. The experimental section demonstrates the potential for such a proof-of-concept technique, which uses tensor completion; even with a small number of collected data, performance starts to increase in suggesting games to the specific user, and this trend is promising for big data.

Keywords: Game-based Learning, Teacher education, Recommendation systems, Interactive learning

1. Introduction

A closer examination of the traditional methods of education, such as direct instruction and textbooks, reveals that these methods dominated the education process before the 21st century. Since technology has gradually transformed instructional methods over the last 20 years, educators now have many tools to use technology effectively in the classroom. Consequently, game-based learning has become an influential part of modern education. Games for students are not particularly new and have been used for many years mostly in early childhood education, but the concern that arises in this domain is the need to integrate games into the design of modern educational approaches at all levels. In addition to the game factors themselves, researchers and practitioners should focus on the instructional support of the game. Goals can be set to promote learning or to guide students to visualize learning outcomes (Pivec 2007).

The web itself cannot address the needs of students due to a large amount of inaccurate information on the web. Computers, on the other hand, cannot motivate students because technology is already at the center of their lives. Moreover, games are commonly linked with fun, but researchers indicate the need for game-based learning is not related to its fun feature (Mattheiss et al., 2009). Educational games could be used to engage students to test theories and achieve objectives. All these technologies (web, computers, digital games) can be used in collaboration to make game-based learning, particularly valuable (Kiili 2005).

The most challenging point in game-based learning integration is that teachers are outraged when considering how to find the most convenient games for their classrooms. The main practical problem is that there are countless games claimed to be educational. However, teachers do not have time to search them all and find the appropriate ones. This research targets at finding a solution to this problem to make game-based learning to be more favorable for teachers.

Dealing with educational games from teachers' point of view, there are challenges, and the efficiency of game-based learning depends mainly on the acceptance by the teachers (Bourgonjon et al. 2013). Therefore, it is important to motivate teachers to implement games regularly to obtain a more engaging learning environment. Hence, our goal in this study is to pursue a recommendation method for teachers to perceive the appropriate games for their students among the huge number of instructional games on the internet.

A major challenge in providing a recommendation system is the vast number of course topics and games. While teachers can differentiate between many categories of courses, it's difficult for them to manually categorize all the relevant games for each topic. Additionally, each teacher has their own teaching style, which can affect their preference for certain games even within a single course topic. It's not possible to assign a game to only one specific course topic. Therefore, using a rule-based approach to prune irrelevant games based on assumed differences between course topics is not practical.

Indeed, in this proof-of-concept study, we collected data from physics teachers to score games on 19 course topics. As can be seen in Figure 2, their response may differ for some games and course topics (details in the experimental setup section). Last, but not least, almost every day, a new educational game may be developed and released. It is practically infeasible to maintain manually a way of identifying games according to your needs. To sum up, in this paper, we focus on designing a *personalized automatic recommendation system*, which could both use other users' experience with the games to be proposed, and a user's previous experience with other games in suggesting games according to course topics and teaching style of the user.

Only a few studies focused on recommendation systems for educational purposes, for example, personalized learning plans (Hulpus, Hayes, and Fradinho 2014) reading materials (Hsu, Hwang, and Chang 2010), students' academic performance (M. Goga, Kuyoro, and N. Goga 2015). To our knowledge, this study is the first in the literature that shows a recommendation system that can be designed to help teachers in finding the appropriate games easily according to the chosen set of topics by the teacher.

This research constitutes a relatively novel area for game-based learning with a breakthrough method for teachers and instructors. We suggest the use of matrix and tensor completion techniques to build a model that could recommend online games, and simulations, in order to assist teachers to incorporate game-based learning. In the proposed recommendation system, a teacher chooses a set of course objects, then, according to this choice, the system retrieves a list of games. To do this, the system first predicts the scores for the new games, i.e., the ones the user has not experienced yet. By using both predicted and known scores, the system returns a list of games with the best scores. While calculating the scores for unseen games, the technique both uses the scores of other users' scoring on the game of interests and the teacher's own scoring data on other games. In that sense, the system is personalized, the retrieved games highly depend on the teacher's own teaching style. Our model-based technique, exploiting ideas from signal processing, can work with a small piece of data (even one additional sample is enough), although preserving the increasing trend in accuracy when we increase training sample. This is also a clear advantage over more advanced machine learning tools, which requires a large number of training samples.

In the experimental part, we compare the proposed recommendation system, with a baseline in which only other users scoring is used. In that case, averaging of other users' scores about the game that is unseen by the teacher will be the technique we will compare with our algorithm. The experimental results show a clear advantage of such a personalized system over a system that is not personalized.

2. Background

The challenging part of using a recommendation system for education is that in a classroom there are a variety of students with different abilities and interests. Even if they study the same subject, their understanding levels are different. It is very critical to find a game that addresses all students' attention in a lecture to be able to incorporate them. Game-based learning provides a solution for K-12 students, but it is worthwhile to know what research says about the use of technology to guide a game-based course plan (Barnett 2001).

Lack of competence among teachers has always been a contributing factor to failures to integrate technology into the classroom over the last few decades (Eteokleous 2008). On the other hand, the number of games that can be used in classes depends on technological opportunities in that school. Teachers benefit from digital games for instructional preparation, during the process, or as a learning tool. Classroom activities, instructional materials, group work, homework and lesson plans are some of the main areas that teachers can integrate a game. While it is generally agreed that game-based learning can be beneficial for K-12 students, there has been argued that most of the games have failed in helping students to develop knowledge. On the other hand, some games have not been well designed, shallow, not attractive, or repetitive, and do not help students to find any possibilities for active exploration (Kirriemuir and McFarlane 2004). Many alternative methods are available for solving this problem, but they all require teachers’ time, which is typically not enough, or even available.

Our work aims to develop a generic recommendation method that can solve a range of problems. In recent years, there has been rapid development in recommendation systems designed for specific learning groups. This has brought great opportunities in education. For teachers, an automated system that can guide activities and intelligently recommend online resources to enhance the learning process would be very useful. While recommendation systems are typically used to increase profit by displaying services a consumer is likely to be interested in, our study focuses on student achievement and integration rather than satisfaction and profit.

3. Preliminaries

Collaborative Filtering and Matrix Completion

Collaborative filtering is a technique utilized by recommendation systems (Ricci, Rokach, and Shapira 2010). Specifically, collaborative filtering is a method to make automatic predictions about the interests of a user by collecting information from many users (Kang, Peng, and Cheng 2016).

It is based on the idea that people who agreed in their evaluation of certain data items in the past are likely to agree again in the future. For example, a person who wants to see a movie might ask for recommendations from users with similar taste in movies. The main steps of a collaborative filtering algorithm for producing a list of recommendations for a user are: (i) Find the most similar users to the user in question using a similarity function, (ii) predict a relevance score for each item not rated by the user based on his/her similar users, and (iii) recommend the top- n items with the highest relevance score.

Recommendation systems that include collaborative filtering typically require very large data sets, such as in the case of the Netflix movie recommendation system (Bennett and Lanning 2007). Collaborative filtering algorithms require the active participation of users; they consist of a system that reflects interest of users, and algorithms that are capable of matching up people with similar concerns (Koren, Bell, and Volinsky 2009).

Matrix completion is one of the most popular methods for recommendation systems and has become superior among collaborative filtering approaches. Matrix completion in recommendation systems may broadly be defined as the completion of a matrix with one dimension indexing users and the other dimension representing data items, typically products, to be scored. Following the dimensions, the elements in the matrix are also the rates that given by other users. To complete the matrix, the main focus is to find missing elements which are needed to be filled automatically by the algorithm because not every user is able to or wants to score every element in the matrix. Thus, there will be recommendations that make sense for users, depending on the items they scored previously and other people's scores on the current item of interest.

4. The Problem

As mentioned above, in collaborative filtering-based recommendation studies the most common problem definition is that the matrix to be completed is defined as a matrix with one dimension indexing users and the other dimension represents the products to be scored. In this work, we modified the classical target matrix construction strategy so that one dimension of the matrix is the context (topics to be learned during lectures), and the other dimension is the product to be scored (e.g., candidate games). This matrix is created for one specific user (teacher in this concept), then it can be completed by the help of other users' scoring matrices. This pre-assumption takes us from matrix representation to *tensor representation*. One advantage of defining such a model is that the system is able to give a satisfactory result even with a limited number of available scoring matrices, e.g., only several users with given scores to the candidate games. In the sequel, we will give a more formal definition of the proposed scheme. Then, we will discuss the mathematical formulation of the problem and the proposed solution.

Mathematically speaking, let's assume N number of games are needed to be scored with the pre-defined range of grades. We fixed these points from 1 to 10 according to what rate they imply each of the M numbers of course topics. The proposed recommendation system can be summarized as follows: A user (in our context, a teacher i) gives scores to a small subset of games corresponding to their involvement with the course topics. It is not feasible for a user (a teacher) to evaluate N number of games for M number of lesson topics, because the number of topics, M, and the number of games, N, may be excessive.

There are at least two possible cases that we can encounter in a realistic scenario. In the first case, even if a specific user able to score all the N numbers of games, he/she may leave empty most of the matrix elements, due to lack of attention, time, etc. Fortunately, the score of the course topics reflects very high correlations. Therefore, even if the user cannot score all the candidate games according to the topic, that is to say, do not fill the whole scoring matrix; it is possible to estimate the unfilled scores in the matrix using the filled ones.

In the second scenario, which will be our main concern, the teacher can score a limited number of games and may want to be suggested new games based on the topic he/she wants to teach in a course. In a real use case, we can assume that the number of candidate games N is too large. For this reason, we want to take advantage of the scores of other users (teachers) and hence their scoring matrices. Let's assume that another user, teacher j, had already experienced some or all the candidate games that teacher i has not. If we can predict the possible scores that teacher i may give to these games by leveraging the data of teacher j, which is the scoring matrix of the teacher j, then we can recommend or not recommend these games to teacher i based on these estimated scores. Moreover, we can extend this scenario so as to have access to T number of scoring matrices of different users, to be used for estimating missing scores between each other. In this case, we have a tensor of data, whose each layer is nothing but a specific user's (teacher) scoring matrix. The corresponding tensor representation can be illustrated in Figure 1.

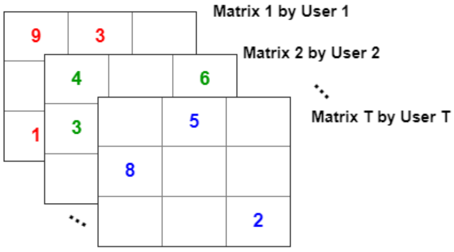

Figure 1: A representative tensor created for multiple users

5. The Method

The matrix completion problem is a well-studied problem in the signal processing literature (Cai, Candes, and Shen 2010), which is used extensively in recommendation systems (Koren, Bell, and Volinsky 2009). Specifically, let $X^{\{i\}} \in R^{M \times N}$ be the scoring matrix of the user i with available elements $\{X^{\{i\}}{}_{(g,k)}: (g,k) \in \Omega\}$, where Ω is the set of filled elements (scores) in the matrix. We can then define the missing elements, which are to be estimated, as the ones $\{X^{\{i\}}{}_{(g,k)}: (g,k) \in \Omega^C\}$, where Ω^C is the complement set of the set of available scores, Ω. One of the state-of-the-art matrix completion strategies is to solve the following convex optimization problem, which is also known as nuclear norm minimization,

$$\min_{M} ||M||_* \text{ subject to } M^{\{i\}}{}_{(g,k)} = X^{\{i\}}{}_{(g,k)}, (g,k) \in \Omega, \quad (1)$$

where $||M||_*$ is the nuclear norm of matrix $M \in R^{M \times N}$ to be estimated. The nuclear norm of a matrix is indeed nothing but the sum of the singular values of it. The importance of the usage of the optimization problem in Equation (1) is that it is the closest convex relaxation to the following NP-hard optimization problem:

$$\min_{M} \text{rank } (M) \text{ subject to } M^{\{i\}}{}_{(g,k)} = X^{\{i\}}{}_{(g,k)}, (g,k) \in, \quad (2)$$

which should spit out a solution matrix M which has maximum correlation between elements from all possible matrices with available elements are indexed by set Ω and missing elements with Ω^C.

In this work, we consider the tensor completion problem from the scored elements of tensor $\mathcal{X} \in R^{M \times N \times T}$, where i^{th} slice of $\mathcal{X}$ is the scoring matrix, $X^{\{i\}} \in R^{M \times N}$, of i^{th} user. By leveraging the recently proposed tensor nuclear norm

(Lu et al. 2019), we can define the following tensor completion problem,

$$\min_{\mathcal{M}} ||\mathcal{M}||_* \text{ subject to } \mathcal{M}_{(g,k,l)} = \mathcal{X}_{(g,k,l)} \textit{ and } (g,k,l) \in \Omega, \quad (3)$$

where the solution tensor $\mathcal{M}$ is completed tensor with estimation of mixing elements, and $\mathcal{X}$ is scoring tensor that compose of T number of users with available elements $\{\mathcal{X}_{g,k,\ell}: (g,k,l) \in \Omega\}$. In this concept, Ω is the set of filled elements (scores) of $\mathcal{X}$. To solve the optimization problem given in Equation (3), we derived an alternating direction method of multipliers-based algorithm (Boyd et al. 2011).

For the sake of brevity, we omit the details of its derivation, but it can be seen in Algorithm 1. The steps of algorithm are explicit except the sub-problem we solve, namely:

$$\arg\min_{\mathcal{Z}} \left\{ ||\mathcal{Z}||_* + \frac{\mu}{2} ||\mathcal{Z} - \left(\mathcal{M} - \frac{\beta}{\mu}\right)||_F^2 \right\}, \quad (4)$$

which can easily be calculated using toolbox given in (Lu et al. 2019) ("Proximal operator of nuclear norm").

When a new instructional game is released, a user-specific prediction system first estimates the possible scores that the user of interest may give for these untested games (whose scores are represented as new columns in the scoring matrix of the user) via Algorithm 1. Then, if the user has already determined his/her topics of interest, Algorithm 2 recommends a pre-defined n number of games according to the maximum n relevance scores of all games. The mentioned relevance scores of a specific game are obtained by the summation of the scores corresponding to the determined topics (if multiple selected) within the column of the scoring matrix. Note that the retrieved recommended games can be from both already seen and scored ones, or unseen (new) ones.

Algorithm 1 Tensor Completion Algorithm

Input: $\mathcal{X}_\Omega$
Auxiliary variables: $\beta, \mathcal{Z} = 0 \in \mathbb{R}^{M\times N\times N}$
Tuning Parameters: μ
Initialize: $\mathcal{M}_\Omega = \mathcal{X}_\Omega, \mathcal{M}_{\Omega^C} = 0$
repeat

$$\mathcal{Z} \leftarrow \arg\min_{\mathcal{Z}} \left\{ \|\mathcal{Z}\|_* + \frac{\mu}{2} \left\| \mathcal{Z} - \left(\mathcal{M} - \frac{\beta}{\mu} \right) \right\|_F^2 \right\}$$

$$\mathcal{M}_{g,k,l} \leftarrow \begin{cases} 1 & \text{if } \mathcal{Z}_{g,k,l} < 1 \\ \mathcal{Z}_{g,k,l} & \text{if } 1 \le \mathcal{Z}_{g,k,l} \le 10, \text{(Projection Part I)} \\ 10 & \text{if } \mathcal{Z}_{g,k,l} > 10 \end{cases}$$

$\mathcal{M}_\Omega \leftarrow \mathcal{X}_\Omega$ (Projection Part II)
$\beta \leftarrow \beta + \mu * (\mathcal{Z} - \mathcal{M})$
until Convergence
return $\mathcal{M}$

Algorithm 2 Game Recommender System

Inputs:
$\mathcal{X}_\Omega$:= Scored elements in tensor form,
Λ:= Set of target topics for the lecture,
i:= User index to be recommended,
n:= Number of games to be recommended.
1. Obtain $M^i_{(\Omega^C)}$ using Algorithm 1.
2. $\tilde{M}^i_{(\Omega^C)} \leftarrow$ Column-wise rescaling (to 1-10) of $M^i_{(\Omega^C)}$
3. Calculate the score of games, i.e., $\text{scores}(k) = \sum_\Lambda \tilde{M}^i_{(\Lambda,k)}$.
return the indices of games with top-n scores

In plain English, we consider the strong connection between topics, games, and user preferences during game selection. Algorithm 1 helps fill in missing elements in a tensor (specifically, unscored games), optimizing for maximum correlation while preserving user-provided scores.

When a user seeks game recommendations based on a topic or topics, Algorithm 2 retrieves the user's corresponding matrix from the estimated tensor. It identifies the column (representing a game) with the highest value for the selected row (representing a topic). If multiple topics are chosen, the algorithm finds the column with the highest average score across the rows. As Algorithm 1's estimation may not be perfect, Algorithm 2 normalizes the estimations column-wise before making recommendations.

6. Experimental Evaluation

Data Collection

19 universally common high school physics topics were determined in this study. As shown in Table 1. In addition, 10 arbitrary educational games were selected from 10 arbitrary websites. For ethical concerns, the names of the games are anonymized. Interested readers can contact the corresponding author. Furthermore, several teachers are asked to test the games and evaluate the efficiency of the games according to specified topics. 6 teachers gave feedback and fulfilled an Excel file. Participant teachers' level of experience is between 3 to 6 years, and

they all are 28 years old. Participant teachers are asked whether they used at least one educational game in their classroom before. The collected data can be seen in Figure 2.

Table 1
Topics that are used in the experimental evaluation part.

TOPICS	10. Potential & Capacitance
1. Vectors	11. Direct Current
2. Kinematics	12. Magnetism
3. Newton's laws	13. Wave, Sound, Light
4. Work, Energy, Power	14. Atomic Nuclear Physics
5. Linear Momentum	15. Fluid Mechanics
6. Rotational Motion	16. Thermodynamics
7. Circular Motion & Gravitation	17. Universe
8. Oscillations	18. Relativity & Quantum
9. Electrical Forces & Fields	19. Matter and Light

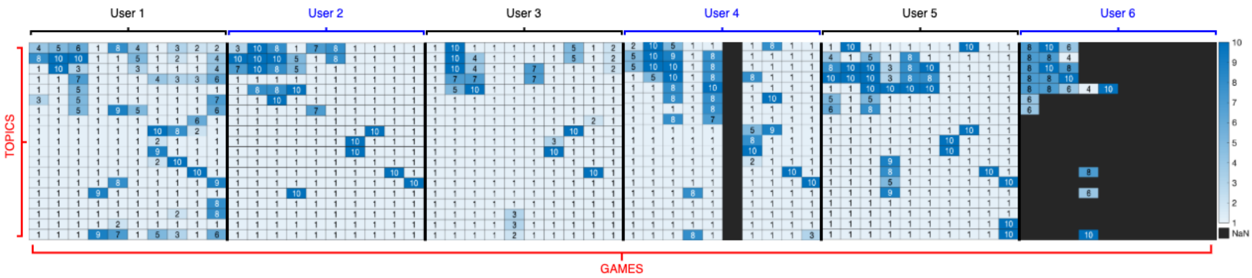

Figure 2: Data used in experimental evaluations. Six physics teachers played the games and sent their scoring tables, which are given above

7. Experimental Design

In order to test the proposed method, three columns were deleted from all matrices. Therefore, we mimic the situation such that the selected user never tried these games with corresponding deleted columns. The elements of the deleted columns were predicted by using Algorithm 1. Then, with these predicted scores, the games are recommended via Algorithm 2. Please note that Algorithm 2 recommends the n-number of games with top n scores. The number of recommended games, n can be pre-defined.

For the prediction of missing elements of matrix of user i, at least one additional user's data, who already scored these missing games, is used. The main purpose here is to see how the performance of the proposed method will change while the number of users who tried one or more games is increased. Thereafter to be able to predict the matrix of the user i, increasing number of matrices from other users were applied. By this means, the performance report was expected to be higher when five users' matrices are used to predict than the one user's matrix is used. These trends can be easily observed in reported performance given in Figure 3. The performance measure that can be seen herein for Algorithm 1, is determined as the Root Mean Square Error (RMSE) for the estimation of missing columns of scoring matrix of a specific user, namely teacher i. The RMSE value is calculated as follows.

$$\text{RMSE} = \sqrt{\frac{1}{|\Omega^C|}\sum_{j=1}^{|\Omega^C|}\left(\mathcal{M}_{\Omega^C}^{\{i\}}(:)_j - \mathcal{X}_{\Omega^C}^{\{i\}}(:)_j\right)^2}, \quad (5)$$

where Ω^C is the set of missing elements in the scoring matrix, thereby $|\Omega^C|$ is the total number of missing elements, and also $\mathcal{M}_{\Omega^C}^{\{i\}}(:)_j$, and $\mathcal{X}_{\Omega^C}^{\{i\}}(:)_j$ are j^{th} elements of vectorized version of the predicted scores of i^{th} user and true values of them, respectively. RMSE is a widely employed performance metric for assessing the quality of estimations used in recommendation systems. A lower RMSE value signifies a superior estimate.

In order to show the performance of Algorithm 2, three random topics are selected and n number of suggestions are retrieved with the maximum scores which are calculated as relevance scores corresponding to selected topics. Recall that the games that are deleted were also selected randomly. While we get performance scores for both Algorithm 1 and 2, each experiment repeated thousand times and an average result is obtained for every one of them. In all the experiments, the tuning parameter, μ in Algorithm 1, was set as 0.015.

As a performance metric for Algorithm 2, we determined the probability of the success rate in n retrieval. The success in n-retrieval is the case where Algorithm 2 returns the game which is tried to be predicted (the game that gets the higher relevance score when the true scoring matrix of the user is available) within these n retrieved ones. As an example, when $n = 2$, the probability of success is the ratio of the cases where the game which should be recommended is returned within 2 suggestions given by Algorithm 2, to the number of trials, which is 1000 in our setup as mentioned.

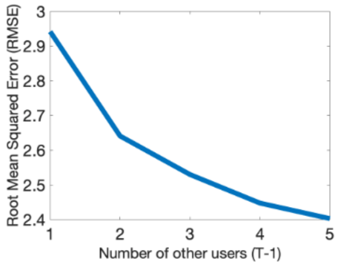

Figure 3: Algorithm 1 performance is reported as RMSE in estimation of missing columns of scoring matrix using data other users. When the number of other users increase (tensor size T), error in the prediction decreases rapidly

When we want to give suggestions to the i^{th} user about arbitrary any topics in the list, we can use the data of other users. We expect that when we increase the number of additional users' data (therefore we increase the tensor size, T), the probability of success also increases. This situation can be seen in Figure 4. Figure 4-(a) shows the probability of retrieving the game that should be recommended when Algorithm 2 suggests only one game. Figure 4-(b) also shows the probability of retrieving this game when Algorithm 2 suggests two games. Likewise, Figure 4-(c) illustrates the probability of success when Algorithm 2 returns three games. The experimental results show that when the number of other users, which already scored the game of interest, increases the performance is also increases. And even with only 5 additional users, it can reach satisfactory results especially for top-3 suggestions are returned. The performance reports in Figure 4 are given as the average performance of situations where user i, whose missing columns need to be estimated, are chosen from the first 5 users, whose data are given in Figure 2. In averaging, User 6 is excluded because most of the columns were missing from the matrix. Nevertheless, User 6, matrix is still use-full in estimation of the other's matrices.

As a competitor method (baseline), we choose the method, where the scoring matrix's missing column estimation for i^{th} user is done by getting an average of the columns that belong to other users who already scored the games of interest. As Figure 4 shows, for all the cases, the proposed solution surpasses this baseline method. The reason why the proposed scheme gives a higher score is that the simple averaging (baseline) technique only uses the correlation among the scoring matrices of different users. However, the proposed scheme both implicitly leverages this outer correlation, but also the correlation within the matrix of the i^{th} user itself. Therefore, the technique is user-specific.

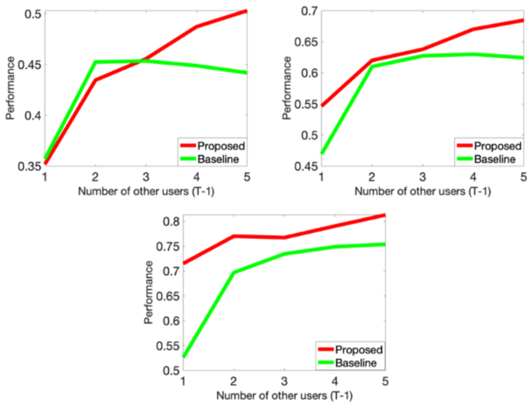

Figure 4: Performance reports of Algorithm 2 vs baseline method. (a) The probability of retrieving the game that should be recommended when Algorithm 2 suggests only one game. (b) The probability of retrieving this game when Algorithm 2 suggests two games. (c) The probability of success when Algorithm 2 returns three games

8. Conclusion

Technology has become increasingly important in education, but it cannot replace the role of teachers in classrooms. Game-based learning has emerged as a new learning method to motivate students and create an active learning environment. However, finding the most relevant games to support a specific subject can be a challenge for teachers due to the vast number of games available. This research proposes a recommendation method for teachers to choose the most suitable games for their students, with a focus on high school physics curriculum. The recommended technique, based on tensor completion, shows promise even with a small amount of collected data, and could achieve high performance with enough data. This study fills a gap in the literature by suggesting a recommendation method for game-based learning.

References

Barnett, H., 2001. Successful K-12 Technology Planning: Ten Essentials Elements. ERIC Clearinghouse on Information and Technology.

Bennett, J. and Lanning, S., 2007. Proceedings of KDD cup and workshop.

Bitner, N. and Bitner, J.O.E., 2002. Integrating technology into the classroom: Eight keys to success. Journal of technology and teacher education, 10(1), pp.95-100.

Bourgonjon, J., De Grove, F., De Smet, C., Van Looy, J., Soetaert, R. and Valcke, M., 2013. Acceptance of game-based learning by secondary school teachers. Computers & education, 67, pp.21-35.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning, 3(1), pp.1-122.

Cai, J.F., Candès, E.J. and Shen, Z., 2010. A singular value thresholding algorithm for matrix completion. SIAM Journal on optimization, 20(4), pp.1956-1982.

Eteokleous, N., 2008. Evaluating computer technology integration in a centralized school system. Computers & Education, 51(2), pp.669-686.

Feldman, A. and Kropf, A., 1999. Teachers as curriculum decision makers: the selection of topics for high school physics. Journal of Curriculum and Supervision, 14(3), p.241.

Goga, M., Kuyoro, S. and Goga, N., 2015. A recommender for improving the student academic performance. Procedia-Social and Behavioral Sciences, 180, pp.1481-1488.

Hsu, C.K., Hwang, G.J. and Chang, C.K., 2010. Development of a reading material recommendation system based on a knowledge engineering approach. Computers & Education, 55(1), pp.76-83.

Hulpuş, I., Hayes, C. and Fradinho, M.O., 2014. A framework for personalised Learning-plan recommendations in Game-based learning. Recommender Systems for Technology Enhanced Learning: Research Trends and Applications, pp.99-122.

Inan, F.A. and Lowther, D.L., 2010. Factors affecting technology integration in K-12 classrooms: A path model. Educational technology research and development, 58, pp.137-154.

Kang, Z., Peng, C. and Cheng, Q., 2016, February. Top-n recommender system via matrix completion. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 30, No. 1).

Kiili, K., 2005. Digital game-based learning: Towards an experiential gaming model. The Internet and higher education, 8(1), pp.13-24.

Kirriemuir, J. and McFarlane, A., 2004. Literature review in games and learning.

Koren, Y., Bell, R. and Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer, 42(8), pp.30-37.

Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z. and Yan, S., 2019. Tensor robust principal component analysis with a new tensor nuclear norm. IEEE transactions on pattern analysis and machine intelligence, 42(4), pp.925-938.

Mattheiss, E. E., Kickmeier-Rust, M. D., Steiner, C. M., & Albert, D. (2009). Motivation in Game-Based Learning: It's More than'Flow'. In DeLFI Workshops (Vol. 7, pp. 77-84).

Pivec, M., 2007. Play and learn potentials of game-based learning. British journal of educational technology, 38(3), pp.387-393.

Ricci, F., Rokach, L. and Shapira, B., 2010. Introduction to recommender systems handbook. In Recommender systems handbook (pp. 1-35). Boston, MA: springer US.

Sudzina, M.R., 1993. Technology, Teachers, Educational Reform: Implications for Teacher Preparation.

Zaiane, O.R., 2002, December. Building a recommender agent for e-learning systems. In International Conference on Computers in Education, 2002. Proceedings. (pp. 55-59). IEEE.