ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse320942020.pdf
https://doi.org/10.30534/ijatcse/2020/320942020

Integrating courses' relationship into predicting student performance

Thanh-Nhan Huynh-Ly[1*], Huy-Thap Le[2], Thai-Nghe Nguyen[3]
[1,2]Department of Information Technology, Lac Hong University, Vietnam
[1]Department of Information Technology, An Giang University, Vietnam
[1]Vietnam National University Ho Chi Minh City, Vietnam
[3]Department of Information Systems, Can Tho University, Vietnam,
[1*]hltnhan@agu.edu.vn, [2]lhthap1007@gmail.com,[3]ntnghe@cit.ctu.edu.vn

ABSTRACT

In Intelligent Tutoring System (ITS) as well as the E-learning system at the university, predicting student learning performance to suggest courses is an essential task of an academic advisor. Many kinds of research address to solve this problem with diverse approaches such as classification, regression, association rules, and recommender systems. Recently, it was a measurable success in using collaborative filtering in the recommender system, especially the matrix factorization technique, to build the courses' recommendation system. There are many advances to improve the accuracy of the prediction, such as using student profiles, course properties, or course relationships; however, they have not been mined. This study proposes an approach which integrates the course relationships into the courses' recommendation system to improve the prediction accuracy. Experimental results of the proposed approach are positive when we validate the published educational datasets.

Key words: Predicting student performance, course recommendation system, educational data mining.

1. INTRODUCTION

The reason for the original intention of designing and developing such systems was the vision that Artificial Intelligence (AI) could produce a promising solution to the limitations of educational professionals facing. In 1984, Bloom et al. [1] conducted experiments comparing student learning under conditions. Thirty students per teacher class versus one-to-one tutoring, and found that individual training is much more effective as group teaching. Accordingly, the personalized prediction is much more useful than general rule prediction for the whole group of students.

Therefore, AI scientists were keenly seeking a meaningful venue for their enthusiasm to spread the power of AI in several traditional fields when AI was growing. Computer scientists, cognitive scientists, educational professionals noticed the newborn Intelligent Tutoring Systems (ITS) fulfill their various goals. Ali, in paper [2], reviewed the historical survey of ITS development. ITS uses AI methods and support quality learning for individuals with no or a little human assistance.

Although different ITSs may have diverse structures [3], the principal structure of an ITS contains four components (called modules or models) such as Student-Model, Tutoring-Model, Domain-Model, and User-Interface. The student model is a vital component of any ITS. It observes student behaviors in the tutor and creates a quantitative representation of student properties of interest necessary to customize instruction, respond effectively, engage students'attention, and promote learning. To ensure the positive feedback feature for learners in the student model, predicting student performance (PSP) is first raised for research. To address the PSP problem, many types of studies may be using the learner's behavior or learner's grade [4]. Using learner behavior is an implicit method in which researchers can predict student performance by observes the learning activities of the student through the application system. Nevertheless, using the grade or mark of students is an explicit and straightforward method because all schools had a student grading management system. Therefore, this approach is widely used and in this article too.

A web-based math tutoring system was first created in 2004 as joint research conducted by Worcester Polytechnic Institute and Carnegie Mellon University. It is ASSISTmentsthat came from the idea of combining assisting the student with the automated assessment of the student's proficiency at a fine-grained level [5]. In 2010-2011, there were over 20,000 students and 500 teachers using the system. Students have solved more than 20,000,000 problems in one year. Because of a large number of students as well as problems, the data for mining should be increasingly more abundant and useful.

Because of the rapidly growing amount of online social networks like Facebook, Twitter, so many researchers have considered the approaches for recommender systems based on social networks. Several experiments confirmed that the social network providesinformation from independent sources, which can improve the quality of recommendations. Thus, the authors of [6] integrated the relationship between members of the classroom into the training model, which made the prediction is better accurate than standard methods.

Following the success of integrating social networks, this research continues adding the courses' relationship to the prediction model because of a similar hypothesis (learners have similar competencies when they learn the related subjects). The experimental results of the study are very positive because of utilizing and exploiting meta-data, increasing the accuracy of prediction.

2. RELATED WORK

There have been many researchersaddress predicting student performance by data mining methods. However, each method has both advantages and disadvantages. Therefore, they proposed many approaches from traditional data mining algorithms to start-of-the-art methods such as decision tree, k-NN, Bayesian network, case-based reasoning, support vector machines, neural networks, association rules, sequential rules, game theory, deep learning, and genetic algorithm.

The authors of the paper [7] listed and compared the results of implementing the different algorithms. They showed J48 is the best decision tree algorithm that can be predictingstudent's dropoutindicators. In another work [8], theauthors analyze the capability of data mining techniques, particularly on the performance of Naive Bayes and C4.5 algorithms to achieve the model of academic performance. Thus, many comparative results are also interested in and studied by researchers. For example, [9] compared two methods between Decision Tree and Bayesian Network algorithms for predicting the academic performance of students at two academic institutes.In [10], the intelligent course recommendation system uses association rules that can recommend courses to the student by common rule; however, this system is not personalized for each student. Moreover, Huu-Quang Nguyen et al. [11] have used the sequential rules algorithm applied to the problem of predicting student performance to give suggestions for students to choose elective courses. In another study [12], they proposed a system for academic advising using case-based reasoning (CBR) that suggests to the student the most suitable major in his case, after comparing the historical case by the student case.

Recently, it has been popular to transfer knowledge from one domain to another has gained much consideration among scientists. Tsiakmaki M. et al. [13] used the transfer learning, which a machine learning approach (deep neural networks) is aiming to exploit the knowledge retrieved from one problem for improving the predictive performance of a learning model. In [14], the authors focus on designing a recommender system that recommends a set of learning objects to multiple students. Moreover, to deal with the issue of multi-decision group recommendation, they model the recommendation process as a non-cooperative game to achieve Nash equilibrium and prove the effectiveness of their proposed model with a case study experiment. Furthermore, they built the system to help students choose elective courses by using a hybrid multi-criteria recommender system with genetic optimization [15].

Rivera A.C. et al. [16] had a systematic mapping study about recommender systems (RS) in education. Thus, they have several statistical methods to address the problem of predicting student performance by using RS. In [17], they proposed another multi-relational approach for recommender systems that can be applied for predicting student performance and assessed the applied model. However, the study depends on the availability of data for the experiment.

In the paper [18], the authors proposed some methods for building course recommendation systems. Those methods are analyzed and validated by using anunpublish dataset before selecting the appropriate techniques, and they presented the framework for developing the course recommendation system. However, this study focuses on application systems and use baseline methods.Similarly, in the paper [19], the authors proposed to exploit multiple relationships by using multi-relational factorization models (MRMF) to improve accuracy for the PSP problems in Student-Model. However, these studies have not taken advantage of social relations.

Several works considered integrating social networks into RS, e.g., [20] improved the prediction accuracy by utilizing social networks of users in many ways. In the study [21], the authors have made a comparison of methods to integrate social networks into the MF. However, the algorithm is restricted only used for data sets that have user relationships. Rui Chen et al. [22] proposed a novel social matrix factorization-based recommendation method, which improves the recommendation quality by fusing user's social status and homophiles. Experimental results of these studies show that when integrating user relationships, predictive models can improve accuracy.

Recently, researchers are also interested in integrating item relationships into the RS. In the paper [23], the authors described the way to exploit information from an item and made the predicting with a better result. Tu Minh Phuong et al., in the paper [24], presented a method for making context-aware recommendations, which exploits the transitivity of the interactions between users and items on the user-item graph to augment the direct communications, so it reduced the negative effect of sparse data. However, these studies focus on the problems in the e-commerce and entertainment fields.

In terms of application for recommending in university, the authors of the paper [25] concluded with "smart collaborative learning" as a relevant concept that adopts smart interactions to promotes modern methods of collaboration between teams of smart learners.

Summarily, in this work, we propose an approach to gather the relationships of the courses (e.g., knowledge/skills) and use them for integrating into the Matrix Factorization for solving the PSP problem in the ITS.

3. PROPOSED METHOD

3.1Problem Definition

Student grading management systems seem to be available to all universities, but they have not yet exploited them effectively. It is useful to exploit them to predict student performance by using computer science methods. Several datasets are published on the internet,such as Algebra, Bridge to Algebra 2008-2009, and ASSISTments. Although different datasets may have diverse structures, the principal structure contains three main fields (Student/User Id, Course/ProblemId, Performance/Correct). Figure 1 shows the snapshot of the ASSISTments dataset. Some fields are necessary for mining, such as "User_id", "Student_class_id", "Problem_id", "Skill_id", and "Correct".

	A	B	C	D	E	F	G
1	user_id	student_class_id	assignment_id	assistment_id	problem_id	skill_id	correct
2	77759	12138	245748	12914	12914	231	1
3	77759	12138	245748	15320	15320	231	1
4	77759	12138	245748	14529	14529	231	1
5	77912	12138	245698	1159	1159	100	0
6	77912	12138	245698	1647	1647	93	1
7	77912	12138	245698	2705	2705	100	1
8	77912	12138	245698	2186	2186	35	1
9	77912	12138	245698	1653	1653	35	1

Figure 1: A snapshot of the data sample

The RS has three main terms asthe user, item, and rating, andthe task of RS is to predict the rating score that the user would provide for all un-rated items, then recommending the top-N highest predicted rating to the user. Similarly, the PSP problem contains three essential objects: student, course, and performance. In the setting of PSP, the task is predicting the course's result that the students have not learned or solved. In figure 2, there is a mapping between the PSP and RS. Where students, course, and grading would be correspondingtothe user, item, and rating, respectively.

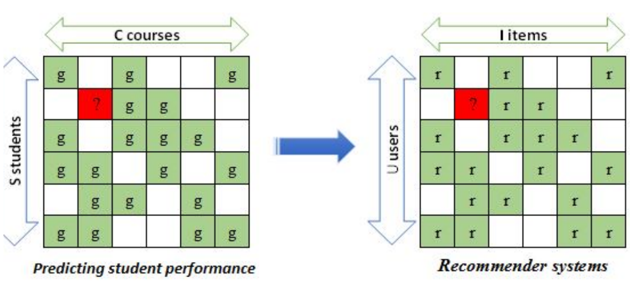

Figure 2: The similar mappingbetween PSP and RS.

Figure 3 shows an example of how we can factorize the students and problems (the performance is correct set be '1' and incorrect set be '0'). From this point to the rest of the paper, we call course, problem, exercise, question, task interchangeably.

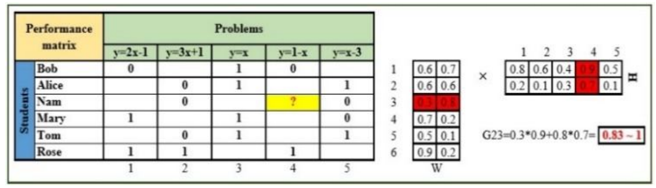

Figure 3: An example of factorizing on students and problems

For improving the accuracy of prediction, many researchers integrated some information from independent sources. These studies have a better result.The paper [6] proposed an approach for integrating information of the social networks into the ITSs that can utilize the advantages of social networks information (e.g., classmate, course-mate, roommate) for the prediction models. They assume that "If student s1 is student s2's friend and s1 study well (good performance), then s2 isaffected" and backward. Indeed, these results are very accurate.Following the achievement, we proposed the same assumption. The content of a subject may relate to other subjects. As such, the subjects have a close relationship with each other. With the same idea: "If problems' A' and 'B'require some same skills, and a student solved the problem 'A' sohe or she can also solve the problem 'B.'" Therefore, if we exploit well this relationship, we can improve the efficiency of students'learning ability prediction.

To apply this approach, we need to transform the relationship data into a matrix called the relationship matrix. For instance, in figure 1, three problems ("12914", "15320", "14529") have the same skill "231". These relationship matrices are a binary matrix that both rows and columns are the student objects for the social networks, the course objects for the course relationships. If two objects are related, then the cell's value is set '1', otherwise '0' or null. Figure 4 shows an example of the relationship matrices.

T	S_2	S_2	S_3	S_4	S_5	S_6	S_7	S_8
S_1	1	1			1			
S_2	1	1			1			
S_3			1			1		
S_4				1			1	1
S_5	1	1			1			
S_6			1			1		
S_7				1			1	1
S_8				1			1	1

THE SOCIAL NETWORKS MATRIX

R	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_1	1		1		1	1		
C_2		1			1			1
C_3	1		1			1		
C_4				1			1	1
C_5	1	1			1			
C_6	1		1			1		
C_7				1			1	1
C_8		1		1			1	1

THE COURSE RELATIONSHIPS MATRIX

Figure 4: An example of the relationship matrices

3.2 Basic Methods

Recommender systems typically suggest a list of recommendations by using collaborative filtering or content-based filtering, or a hybrid approach. Collaborative filtering methods are categorized as memory-based and model-based collaborative filtering. A well-known example of memory-based approaches is a user-based algorithm, and that of model-based approaches is the latent factor models.

A. Student/Course average recommendation

The simplest methods are baselines, such as the global average and student or course average [18]. Precisely, for the global average, we can compute an average gradeon the training set, then use this number as the predicted values for all instances in the test set.

The student or course average is similar to the global average but averaging for each student or course, as in the following equation, where *g*, *s*, and *c*are denoted for grade/mark, student, and course, respectively.

$$\hat{g}_s = \frac{\sum_{(s',c,g)\in D^{train}|s'=s}(g)}{|\{(s',c,g)\in D^{train}|s'=s\}|} \quad (1)$$

B. Student/Course k-NNs collaborative filtering

In collaborative filtering methods of the recommender systems, we usually assume that "similar users" may like "similar items" and vice versa. Likewise, in the education domain, we also assume that "similar students" may have analogous performances on "similar courses" [18]. Thus, the user-based or item-based collaborative filtering would be a simple choice for taking into account correlations between the students and the courses in student performance prediction. We briefly describe how to use the k-nearest neighbor's collaborative filtering in the following. This method is called "Student-kNNs."

In this method, the predicted mark $\hat{g}_{sc}$ of the student s on the course c is based on the mark of its nearest neighbors (students) on that course. The prediction function is determined by:

$$\hat{g}_{sc} = \frac{\sum_{s'\in K_s} sim(s,s')g_{s'c}}{\sum_{s'\in K_s}|sim(s,s')|} \quad (2)$$

Where K_s is the set of k nearest neighbors of a student s, and $sim(s,s')$ is the similarity between student s and s' which can be computed by using the Cosine or Pearson similarity:

$$sim_{cosine}(s,s') = \frac{\sum_{c\in C_{ss'}} g_{s'c} \cdot g_{sc}}{\sqrt{\sum_{c\in C_{ss'}} g_{sc}^2 \sum_{c\in ss'} g_{s'c}^2}} \quad (3)$$

$$sim_{pearson}(s,s') = \frac{\sum_{c\in C_{ss'}}(g_{sc}-\bar{g}_s)(g_{s'c}-\bar{g}_{s'})}{\sqrt{\sum_{c\in C_{ss'}}(g_{sc}-\bar{g}_s)^2 \sum_{c\in C_{ss'}}(g_{s'c}-\bar{g}_{s'})^2}} \quad (4)$$

Where $c_{ss'}$ is a set of courses performed by both student s and students$'$; $\bar{g}_s$ and $\bar{g}_{s'}$ are the means (average) performance over all the courses of student s ands', respectively. Another prediction method, instead of using the weighted sum, one could also use the prediction using deviations from the user (student) mean. Using deviation to determine the performance of the student s on courses $c$by:

$$\widehat{g_{sc}} = \bar{g}_s + \frac{\sum_{s'\in K_s} sim(s,s')(g_{s'c}-\bar{g}_s)}{\sum_{s'\in K_s}|sim(s,s')|} \quad (5)$$

C. Matrix factorization method

The matrix factorization is flexibility model in dealing with various datasets, applications, and fields. Approximating a matrix $X \in R^{|S|\times|C|}$ by a product of two very small matrices W and H is the main idea of matrix factorization.

Figure 5 describes simply the graphical model that factorized the grading matrix. The system gives a predicted grading for the student will learn a course by matrix factorization technique. This model is a benchmark model for integrating social networks as well as course relationships. Indeed, the proposed algorithms later inherited this graphical model. The symbols and concepts in this figure are presented in detail the description of each proposed algorithm.

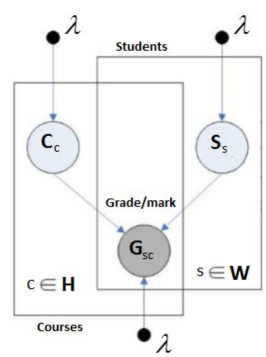

Figure 5: The graph illustrates matrix factorization technique

Based on [18], $X \approx WH^T$,where $W \in R^{|S|\times|K|}$ is amatrix where each row s is a vectorW_s (rendering the student *s) and* has k latent factors. Similarly, $H \in R^{|C|\times|K|}$ is amatrix where each row c is a vectorh_c(Rendering the course c) and has$k$ latent factors.Letw_{sk}andh_{ck} are the elements of two matrices *W* and *H*, respectively. To predict the grade/mark$g$ for a student s to study a course c:

$$\widehat{g_{sc}} = \sum_{k=1}^{K} w_{sk}\, h_{ck} = w_s h_c^T \quad (6)$$

Root Mean Square Error (RMSE) isa criterionto find optimal values for the parameters *W* and *H*. it is determined:

$$RMSE = \sqrt{\frac{1}{|D^{test}|}\sum_{s,c,g\in D^{test}}(g_{si} - \widehat{g_{si}})^2} \quad (7)$$

In the MF technique [18], training the model is to find the optimal parameters W and H. These matrices are initialized with some random values (from the normal distribution). Besides, we addeda termto the error function for preventing over-fitting. The error function is determined:

$$O^{MF} = \sum_{(s,c,g)\in D^{train}} \left(g_{sc} - \sum_{k=1}^{K} \widehat{w_{sk}}\, \widehat{h_{ck}}\right)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2) \quad (8)$$

Where$\|\cdot\|_F^2$is a Frobenius[1]norm, λ is a regularization weight. The valuesofw_{sk} andh_{ck} are updated,respectively.

$$w' = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk}) \quad (9)$$

$$h' = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck}) \quad (10)$$

Where β is the learning rate. We update the values of W and H iteratively until the error converges to its minimum ($O_{n-1}^{MF} - O_n^{MF} < \varepsilon$) or reaching a predefined number of iterations. Finally, the performance of the student s on courses c is now determined by equation (11) and figure 6*:*

$$\widehat{g_{sc}} = \sum_{k=1}^{K} w_{sk}\, h_{ck} = w_s h_c^T \quad (11)$$

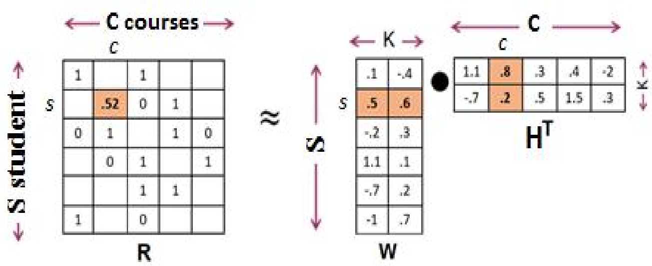

Figure 6: The prediction process of matrix factorization

D. Biased Matrix factorization method

We have presented the standard matrix factorization technology to encode the student/course latent factors in the previous section. Next, we use the biased matrix factorization (BMF) to deal with the problem of "user effect" ("user bias") and "item effect" ("item bias") [18]. In the educational setting, the user and item biases are, respectively, the student and course biases/effects. The student effect (student bias) models how a good/clever/bad student is (i.e., how likely is the student to perform a course correctly). Similarly, the course effect (course bias) models how the difficult/easy course is (i.e., how likely is the course to be performed correctly). With these biases, the prediction function for a student s on the course c is presented by

$$\hat{g}_{sc} = \mu + b_s + b_c + \sum_{k=1}^{K} w_{sk}h_{ck}\ \mu = \frac{\sum_{(s,c,g)\in D^{train}} g}{|D^{train}|} \quad (12)$$

$$\mu = \frac{\sum_{(s,c,g)\in D^{train}} g}{|D^{train}|} \quad (13)$$

$$b_s = \frac{\sum_{(s',c,g)\in D^{train}|s'=s} (g-\mu)}{|\{(s',c,g)\in D^{train}|s'=s\}|} \quad (14)$$

[1]https://en.wikipedia.org/wiki/Matrix_norm#Frobenius_norm

$$b_c = \frac{\sum_{(s,c',g)\in D^{train}|c'=c} (g-\mu)}{|\{(s,c',g)\in D^{train}|c'=c\}|} \quad (15)$$

Moreover, the error function is of BMF also changed by adding these two biases to the regularization:

$$O^{BMF} = \sum_{(s,c,g)\in D^{train}} \left(g_{sc} - \mu - b_s - b_c - \sum_{k=1}^{K} w_{sk}h_{ck}\right)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2 + b_s^2 + b_c^2) \quad (16)$$

After updating the values of the matrix W and H iteratively, we use the predicting function (equation 11) for prediction.

3.3 Proposed Methods

A. Social network Matrix factorization method

The main idea of integrating social networks into the MF technique is to replace the matrix W with$\widehat{W}$, which is the latent factor, is influenced by his direct neighbors. After, we process as a baseline method (matrix factorization).

Because of social influence, his direct neighbors affect the behavior of users or students [6]. The first step in integrating is gathering the trust matrix from the social network information (if two students are in the same class, they are neighbors or classmates). We proceed to browse all student data to build a trust matrix.

Figure 7 shows a graphical example of the incorporating trust matrix into matrix factorization for the latent factor matrix W (Student/User factor). Each student s has a neighbor list, and the value of the influence is $T_{s,n} = 1$ (direct neighbor). The regularization term (regularization weight) λ_T be added for normalizing.

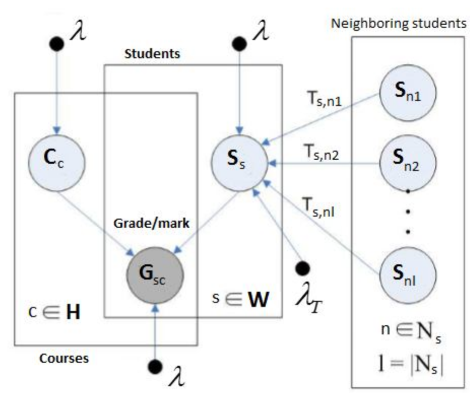

Figure 7: Graphical Model of the Social-MF technique for PSP

In other words, the latent feature vector of s is dependent on the latent feature vectors of all his direct neighbors of n ∈ Ns. We formulate this influence as follows:

$$\widehat{W_s} = \frac{\sum_{n\in N_s} T_{s,n} w_n}{\sum_{n\in N_s} T_{s,n}} = \frac{1}{|N_s|} \sum_{n\in N_s} w_n \quad (17)$$

Where $\widehat{W}_s$ is the estimated latent feature vector of s given the feature vectors of his direct neighbors? Since the relationship matrix $T_{s,n}$ is a binary matrix so that all non-zero values of $T_{s,n} = 1$ and $\sum_{n\in N_s} T_{s,n} = |N_s|$.

To integrate social-networks into the matrix factorization, we only replace W_s by $\widehat{W}_s$ in equation (6), and they become:

$$\hat{g}_{sc} = \sum_{k=1}^{K} \widehat{w}_{sk} h_{ck} = \widehat{w}_s * h_c^T \quad (18)$$

The error function for the Social-MF now becomes:

$$o^{SocialMF} = \sum_{(s,c,g)\in D^{train}} \left(g_{sc} - \sum_{k=1}^{K} \widehat{w_{sk}}\,\widehat{h_{ck}}\right)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2) + \lambda_T \sum_{s=1}^{S} \left(w_s - \frac{1}{|N_s|}\sum_{n\in N_s} w_n\right)^2 \quad (19)$$

With the new error function of the Social-MF, the w_{sk} and h_{ck} are updated by the equations below (where $e_{sc} = g_{sc} - \hat{g}_{sc}$)

$$w'_{sk} = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk}) + \lambda_T \left(w_{sk} - \frac{1}{|N_s|}\sum_{n\in N_s} w_{nk}\right) - \lambda_T \left(\sum_{t\in S_s} \frac{1}{|N_s|}\left(w_{tk} - \frac{1}{|N_s|}\sum_{w\in N_s} w_{wk}\right)\right) \quad (20)$$

$$h'_{ck} = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck}) \quad (21)$$

In function (20), we denote:

$$L = \lambda_T \left(w_{sk} - \frac{1}{|N_s|}\sum_{n\in N_s} w_{nk}\right) \quad (22)$$

$$R = \lambda_T \left(\sum_{t\in S_s} \frac{1}{|N_s|}\left(w_{tk} - \frac{1}{|N_s|}\sum_{w\in N_s} w_{wk}\right)\right) \quad (23)$$

Abridged function (20) becomes:

$$w'_{sk} = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk}) + L - R \quad (24)$$

Likewise, after updating the values of the matrix W and H iteratively, the predicting function (equation 11) is applied.

B. Course relationship Matrix factorization (CRMF)

With the same idea, this section presents the proposed approach that incorporates course relationships to the matrix factorization model for the problem of PSP in the ITS.

Integrating course relationships into the MF technique is to replace the matrix H with $\widehat{H}$, which is the course/item-latent factor, is effected by similar courses. Then, we process as a baseline (matrix factorization technique).

Because of the course effect, the courses that they have studied usually bemore simple for students to study. The first step in integrating is gathering the course relationship matrix from the course information (if two courses requiresimilar the skills/knowledge, they are the same). We proceed to browse all course data to build a relationship matrix.

Figure 8 is a graphical example of incorporating the course relationships into matrix factorization for the latent factor matrix H (Course/Item factor). Each course c has a similar list, and the value of the influence is $R_{c,m} = 1$ (similar). The regularization term (regularization weight) λ_R is added for normalizing.

Similar to the calculation of user influence in the Social-MF method, we formulate the course-effect as follows:

$$\widehat{H_c} = \frac{1}{|M_c|}\sum_{m\in M_c} H_m \quad (25)$$

The error function for the CRMF now becomes:

$$o^{CRMF} = \sum_{(s,c,g)\in D^{train}} \left(g_{sc} - \sum_{k=1}^{K} w_{sk}\,\widehat{h_{ck}}\right)^2 + \lambda(\|W\|_F^2 + \|H\|_F^2) + \lambda_R \sum_{c=1}^{C} \left(h_c - \frac{1}{|M_c|}\sum_{y\in M_c} h_y\right)^2 \quad (26)$$

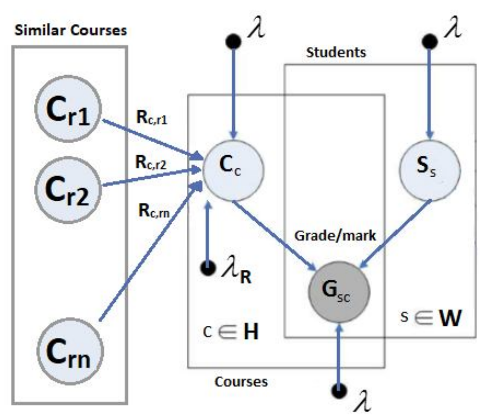

Figure 8: Graphical Model of the CR-MF technique for PSP

With this error function of the CR-MF, the w_{sk} and h_{ck} are updated by the equations below (where $e_{sc} = g_{sc} - \hat{g}_{sc}$)

$$w'_{sk} = w_{sk} + \beta(2e_{sc}h_{ck} - \lambda w_{sk}) \quad (27)$$

$$h'_{ck} = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck}) + \lambda_R \left(h_{ck} - \frac{1}{|M_c|}\sum_{m\in M_c} h_{mk}\right) - \lambda_R \left(\sum_{r=C_c} \frac{1}{|M_c|}\left(h_{rk} - \frac{1}{|M_c|}\sum_{y\in R_c} h_{yk}\right)\right) \quad (28)$$

The function (27) is very long. Therefore, we temporarily set L (left side) and R (right side), and then use this replacing for the procedure below.

$$L1 = \lambda_R \left(h_{ck} - \frac{1}{|M_c|}\sum_{m\in M_c} h_{mk}\right) \quad (29)$$

$$R1 = \lambda_R \left(\sum_{r=C_c} \frac{1}{|M_c|} \left(h_{rk} - \frac{1}{|M_c|} \sum_{y \in R_c} h_{yk} \right) \right) \quad (30)$$

Abridged function (28) becomes:

$$h'_{ck} = h_{ck} + \beta(2e_{sc}w_{sk} - \lambda h_{ck}) + L1 - R1 \quad (31)$$

We still apply the predicting function (equation 11) of the standard MF method for this proposed model.

3.4 Proposed Algorithm

Details of the proposed method that integrates the courses' relationship into Matrix Factorization, are presented in the procedure below "Course-Relationship-Matrix-Factorization - CRMF." This CRMF is factorizing student and course using stochastic gradient descent with K latent factors, β learning rate, λ regularization weight, stopping condition, and R relationship matrix (the binary matrix is constructed independently of another simple procedure), and λ_R regularization weight for the matrix R. For example, in each iteration, we randomly select an instance in the training set $\{s, c, g\}$ then compute the prediction for this student and course, as in lines 10-20. Then, we estimate the error in the iteration and update the values of W and H in lines 21-22

Procedure Course-Relationship-Matrix-Factorization

Input: D^{train}, K, β, R, λ,λ_R, stopping condition

Output: *W*, *H*

1. Let $s \in S$be a student, $c \in C$ a course, $g \in G$ a grade
2. **Let**$W[|S|][K]$, $H[|C|][K]$ be latent factors of students, courses
3. $W \leftarrow N(0, \sigma^2)$
4 $H \leftarrow N(0, \sigma^2)$
5. **while** (Stopping criterion is NOT met) **do**
6. Draw randomly (s, c, g_{sc})fromD^{train}
7. Ns = number of students have friendship in T matrix;
8. $\widehat{g_{sc}} \leftarrow \sum_k^K (W[s][k] * H[c][k])$
9. $e_{sc} = g_{sc} - \widehat{g_{sc}}$
10. for $k = 1$..K do
11. for$m = 1$..Rc do
12. $L1 = L1 - H[m][k]/R_c$
13. end for
14. $L1 = H[c][k] - L1$
15. for$r = 1$.. Cc do
16. for$y = 1$..Rc do
17. $H[r][k] = H[r][k] - H[y][k]/R_c$
18. end for
19. $R1 = R1 + H[t][k]/R_c$
20. end for
21. $W[s][k] = W[s][k] + \beta * (2e_{si} * H[c][k] - \lambda * W[s][k])$
22. $H[c][k] \leftarrow H[c][k] + \beta * (2e_{sc} * W[s][k] - \lambda * H[c][k]) + \lambda_R(L1 - R1)$
23. end for
24. end while
25. return $\{W, H\}$
26. **end procedure**.

4. RESULT

4.1 Dataset

There are several datasets for experimenting the predicting student performance problem such as ASSISTments, Algebra, CTU (gather from the management system of the university). Commonly, we classify into two types of data sets (publish and private dataset). To evaluate the approach method, we should use the publish dataset for more convincing.

The ASSISTments dataset [2], which is published by the ASSISTments Platform, is a web-based tutoring system that assists students in learning math and gives teachers an assessment of the progress of their students. After pre-processing, this dataset contains 8519 students (users), 35978 tasks (items), and 1011079 gradings (ratings).

4.2 Evaluation

In this work, predicting student marks is the task of rating prediction (explicit feedback), so we use a popular measure in RS, which is Root Mean Squared Error (RMSE) for model evaluation. We have used the hold-out approach (use 2/3 of data for training and use 1/3 of data for testing) for experimenting with the models.

Theprediction accuracy depends on the parameters that feed the algorithm. If the parameters were not suitable, the accuracy of the prediction would not be good even though the algorithm is correct. Thus, finding the best parameter is significant.The hyper-parameters search, which is a searching parameter method, is applied for searching all the parameters of the approached models. The hyper-parameters search has two phases, such as raw search (for the long segments) and smooth search (for the short segments). First, the raw search finds the best parameters in the long segments. Next, we use a smooth search to find the nearby parameters. For example, using RMSE as a criterion, the hyper-parameter search results for the models on the ASSISTments dataset are presented in Table 1.

[2]https://sites.google.com/site/assistmentsdata/home

Table 1: Hyper-parameters on ASSISTments dataset

Methods	Hyper Parameter
Student-kNNs	k=5, simMeasure=Cosine
Course-kNNs	k=5, simMeasure=Cosine
MF	β=0.01, #iter=60, K=32, λ=0.1
BMF	β=0.01, #iter=100, K=80, λ=0.15
Social-MF	K=80, λ=3, β=0.01, SocialReg=5, Iter=500
CRMF	K=80, λ=3, β=0.01, ItemReg=3, Iter=300

After having the best parameters, we use them for training and testing each respective model.

4.3 Experimental result

We have compared using course relationships (CRMF) for solving the problem of prediction student performance in the ITS with other methods, such as global average, student average, student-kNNs, course-kNNs, standard MF, BMF, and Social-MF. Fortunately, many open-source librariesimplemented these algorithms, such as LibRec (librec.net) or MyMediaLite (mymedialite.net), that we can inherit from them.

We conduct six experiments, and the experimental results are displayed in figure 9 (please note that we interchangeably call the user as the student and the item as the course). We compared with others, RMSE of the proposed approach (CRMF) is the smallest one (0.423) on the dataset.

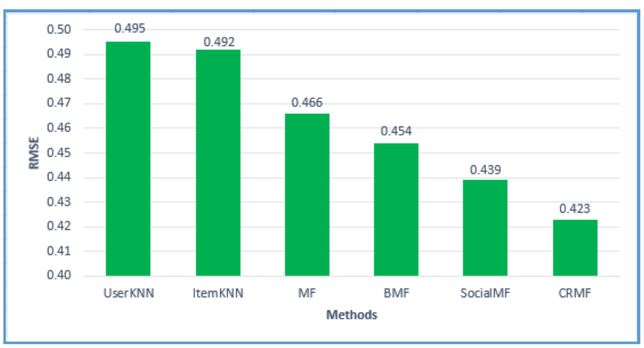

Figure 9: Experiment result on the ASSISTment dataset

5. CONCLUSION

In this paper, we have introduced an approach for integrating the relationships of courses to the intelligent tutoring systems. We used the matrix factorization technique to illustrated embedding the item relationship. With this approach, we can take advantage of the interactions among courses for building the prediction model. Thus, the prediction results can be improved significantly. Conducting experiments on real datasets shows that the proposed approach works well.

In terms of the relationship between the subjects, it is likely to be found in many ways. This work only uses existing information (knowledge for solving the problems or courses) without using sophisticated techniques. The next research is how to find the relation between these courses by algorithms that are highly effective and fast.

REFERENCES

1. B. S. Bloom. **The 2-sigma problem: The search for methods of group instruction as effective as one-to-one tutoring**,Educational researcher, Vol. 13, no. 6, pp. 4–16, 1984. https://doi.org/10.3102/0013189X013006004
2. Sottilare, R. A., Graesser, A., Hu, X., & Holden, H. **Design recommendations for intelligent tutoring systems**, US Army Research Laboratory, Vol. 1, ch. 5, 2013.
3. Ali Alkhatlan and Jugal Kalita. **Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments**, International Journal of Computer Applications, 181(43), p1-20, 2019.
4. Harrison, B.E., & Roberts, D.L. **A Review of Student Modeling Techniques in Intelligent Tutoring Systems**, in Proceedings of Eighth Artificial Intelligence and Interactive Digital Entertainment Conference, 2012
5. Douglas Selent, Thanaporn Patikorn, and Neil Heffernan. **ASSISTments Dataset from Multiple Randomized Controlled Experiments**. In Proc. the Third. ACM Conference on Learning @ Scale. Association for Computing Machinery, New York, 2016, pp. 181–184. https://doi.org/10.1145/2876034.2893409
6. Huynh-Ly Thanh-Nhan, Le Huy-Thap, and Nguyen Thai-Nghe. **Toward integrating social networks into Intelligent Tutoring Systems**. In Proceedings of the 2017 International Conference on Knowledge and Systems Engineering, IEEE, 2017, pp. 112-117.
7. Jay S. Gil, Allemar Jhone P. Delima, and Ramcis N. Vilchez.**Predicting Students Dropout Indicators in Public School using Data Mining Approaches**. International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 1. pp. 774-778, 2020. https://doi.org/10.30534/ijatcse/2020/110912020
8. Edison L. Polinar, Allemar Jhone P. Delima and Ramcis N. Vilchez, **Students Performance in Board Examination Analysis using Naïve Bayes and C4.5 Algorithms.** International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9, No. 1. pp. 753-758, 2020. https://doi.org/10.30534/ijatcse/2020/107912020
9. Bogarín, A., Cerezo, R. and Romero, C.**A survey on educational process mining**. WIREs Data Mining Knowl Discov, vol. 8, no.1, 2018.
10. P. Rojanavasu.**Educational Data Analytics using Association Rule Mining and Classification**, 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, Thailand, 2019, pp. 142-145.
11. Huu-Quang Nguyen, Thi-Thiet Pham, Van Vo, Bay Vo, and Thanh-Tho Quan. **The predictive modeling for learning student results based on sequential rules**, International journal of innovative computing information and control, vol. 14, no. 6, pp. 2129:2140, 2018

12. Mona Masood and Nur Azlina Mohamed Mokmin.**Case-based Reasoning Intelligent Tutoring System: An Application of Big Data and IoT**, In Proceedings of the 2017 International Conference on Big Data Research (ICBDR 2017). Association for Computing Machinery, New York, NY, USA, 2017, pp. 28–32.
13. Tsiakmaki M, Kostopoulos G, Kotsiantis S, Ragos O. **Transfer Learning from Deep Neural Networks for Predicting Student Performance**, Applied Sciences. 10(6):2145. ISSN: 2076-3417, 2020
14. A. Zia, and M. Usman. **Elective Learning Objects Group Recommendation Using Non-Cooperative Game Theory**, International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 2018, pp. 194-199.
15. A. Esteban, A. Zafra, C. Romero.**Helping university students to choose elective courses by using a hybrid multi-criteria recommendation system with genetic optimization**, Knowledge-Based Systems, Vol. 194, 2020, 105385, ISSN 0950-7051, 2020
16. Rivera A.C., Tapia-Leon M., Lujan-Mora S. **Recommendation Systems in Education: A Systematic Mapping Study**, In Proceedings of the International Conference on Information Technology & Systems (ICITS 2018). Advances in Intelligent Systems and Computing, vol 721. Springer, Cham, 2018
17. Thai-Nghe N., Nhut-Tu M., Nguyen HH. **An Approach for Multi-Relational Data Context in Recommender Systems**. In: Nguyen N., Tojo S., Nguyen L., Trawiński B. (eds) Intelligent Information and Database Systems. ACIIDS 2017. Lecture Notes in Computer Science, vol 10191. Springer, Cham. https://doi.org/10.1007/978-3-319-54472-4_66
18. Huynh-Ly Thanh-Nhan, Huu-Hoa Nguyen, and Nguyen Thai-Nghe. **Methods for building course recommendation systems**. In Proceedings of the 2016 International Conference on Knowledge and Systems Engineering, IEEE Xplore, 2016, pp.163-168.
19. Nguyen Thai-Nghe and Lars Schmidt-Thieme. **Multi-Relational Factorization Models for Student Modeling in Intelligent Tutoring Systems**. In Proceedings of the 2015 Seventh International Conference on Knowledge and Systems Engineering, 2015, pp. 61-66.
20. B. Díaz-Agudo, G. Jimenez-Diaz and J. A. Recio-García, **SocialFan: Integrating Social Networks Into Recommender Systems**, 2018 IEEE 30th International Conference on Tools with Artificial Intelligence, Volos, 2018, pp. 171-176.
21. Zhang M., Hu B., Shi C., Wu B., Wang B. **Matrix Factorization Meets Social Network Embedding for Rating Prediction**. In: Cai Y., Ishikawa Y., Xu J. (eds) Web and Big Data. APWeb-WAIM 2018. Lecture Notes in Computer Science, vol 10987. Springer, Cham, 2018
22. Rui Chen et al., **A Novel Social Recommendation Method Fusing User's Social Status and Homophily Based on Matrix Factorization Techniques**, in IEEE Access, vol. 7, pp. 18783-18798, 2019.
23. Lops, P., Jannach, D., Musto, C. et al. **Trends in content-based recommendation**, User Modeling and User-Adapted Interaction. Springer (2019) ISSN 1573-1391
24. Tu Minh Phuong, Do Thi Lien, Nguyen Duy Phuong,**Graph-based context-aware collaborative filtering**. Expert Systems with Applications, vol. 126, pp. 9-19, 2019
25. Ouidad Akhrif, Chaymae Benfares, Younès El bouzekri el idrissi, and Nabil Hmina. **Collaborative learning services in the smart university environment**, In Proceedings of the 4th International Conference on Smart City Applications (SCA '19). Association for Computing Machinery, New York, NY, USA, Article 42, 1–6, 2019. https://doi.org/10.1145/3368756.3369020