View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by D-Scholarship@Pitt

Tensor Factorization for Student Modeling and Performance Prediction in Unstructured Domain

Shaghayegh Sahebi
Intelligent systems Program
University of Pittsburgh
Pittsburgh, PA
shs106@pitt.edu

Yu-Ru Lin
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA
yurulin@pitt.edu

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA
peterb@pitt.edu

ABSTRACT
We propose a novel tensor factorization approach, Feedback-Driven Tensor Factorization (FDTF), for modeling student learning process and predicting student performance. This approach decomposes a tensor that is built upon students' attempt sequence, while considering the quizzes students select to work with as its feedback. FDTF does not require any prior domain knowledge, such as learning resource skills, concept maps, or Q-matrices. The proposed approach differs significantly from other tensor factorization approaches, as it explicitly models the learning progress of students while interacting with the learning resources. We compare our approach to other state-of-the-art approaches in the task of Predicting Student Performance (PSP). Our experiments show that FDTF performs significantly better compared to baseline methods, including Bayesian Knowledge Tracing and a state-of-the-art tensor factorization approach.

Keywords
Tensor factorization, student modeling, predicting students performance, learning analytics

1. INTRODUCTION
The growth of Massive Open Online Courses (MOOC) has rapidly increased the volume of data on students' education and learning behavior. This abundance of data calls for approaches that can automatically make sense of such data, and that remove the need for manual handling of such massive amounts of data. Predicting students performance and modeling student knowledge are two of the tasks that help researchers to understand such data. The goal in predicting student performance (PSP), is to estimate if a specific target student can handle a learning material successfully – for example, whether the student can succeed or fail at solving a specific quiz. Student knowledge modeling aims to quantify or infer a student's knowledge at each moment in time in each of the possible skills (or concepts) the student may have. The set of skills are defined either manually or automatically based on the learning materials.

Understanding students' attempt data through PSP and student knowledge modeling encourages teachers to design better courses, allows for targeted personalization of course pace, and provides more accurate automatic learning material recommendation to students. Hence, a primary focus in educational data mining literature is on predicting student performance and student knowledge modeling. For example, Bayesian Knowledge Tracing was one of the pioneering approaches that could predict the success or failure of students in solving problems [1].

Recently, other approaches, such as factorization models, have been used for PSP. For example, Performance Factor Analysis (PFA) [5] is another approach to PSP and cognitive modeling. PFA takes into account the effects of the initial difficulty of the skills (knowledge components) and prior successes and failures of a student at learning the skills associated with the current item. These approaches require prior knowledge of the overall domain model – the association between skills and learning material.

More recent approaches have sought to overcome this limitation by using latent factor approaches. For example, Thai-Nghe et al. experimented on a context-aware factorization algorithm, based on collaborative filtering approaches, in the relevant recommender system literature [9]. Sahebi et al. studied various methods of the educational data mining field with matrix and tensor factorization approaches, from the recommender systems literature for PSP [7]. Lan et al. used quantized matrix completion to predict students' performance in SPARFA-Lite [4]. This method solves a convex optimization problem and gives a global optimum solution.

Tensors, or multi-dimensional arrays, have been used in the literature to represent data on student attempts [6]. One of the main reasons that tensors are a suitable representation for modeling educational data is their seamless integration ability and flexibility in representing multiple dimensions of the data, such as students, questions, time, and topic structure. Another reason for using tensors is their capability for decomposing interactions in multi-dimensional data.

While various tensor decomposition models and algorithms already exist in the literature [3], the potential for versatile modeling of tensors in the educational data mining field is under-explored. Although previous tensor factorization models that have been used in the literature have resulted in comparable performance in the task of PSP [6, 8], they are not tailored to educational data. More specifically, these models are built for purposes other than educational data mining (such as recommender systems), and thus do not consider the characteristics of educational data mining challenges.

One of these challenges is increases in student knowledge that occurs while they interact with learning material. As the students learn through quizzes, readings, and other learning resources, they incrementally learn the underlying skills that are present in these resources. Thus, this amount of knowledge increase for a student depends on the material that the student is interacting with. The current tensor factorization approaches that are used for PSP in the literature do not model this interaction.

In this paper, we provide a solution to this problem by proposing a unique tensor factorization-based approach that can account for the constant learning of students. Our proposed tensor factorization model, called *feedback-driven tensor factorization*, directly models the increases in student knowledge by adding a feedback-based constraint on the previous student's knowledge and the current learning material that a student is using. We compare our approach to Bayesian Knowledge Tracing and a baseline tensor factorization algorithm. Our experiments show the superior performance of our proposed approach, as compared to the baseline methods.

2. FEEDBACK-DRIVEN TENSOR FACTORIZATION (FDTF)

As mentioned in the introduction, the goal of our approach is to predict student performance while considering the fact that students are constantly learning. In order to achieve this goal, we represent student activities on learning material as a three-dimensional tensor $\mathcal{Y}$.

Notations. In this paper, tensors are represented by script letters, e.g. $\mathcal{Y}$; Matrices are denoted by boldface capital letters, e.g. $\mathbf{X}$; and vectors are represented by boldface lowercase letters, e.g. $\mathbf{x}$. In addition, we denote the i^{th} row of a matrix $\mathbf{X}$ as $\mathbf{X}_{i,:}$, the j^{th} column as $\mathbf{X}_{:,j}$, and the entry (i, j) as $\mathbf{X}_{i,j}$.

Suppose that students are working with one resource type and are learning from it. To be more specific, suppose that m students are interacting with n quizzes, and that each student can have multiple attempts (at most l) on each quiz. Then, we can represent the students' attempt sequences on all quizzes as a tensor of size $m \times n \times l$. The k^{th} frontal slice of this tensor ($\mathcal{Y}_{:,:,k}$) shows the success or failure of all students on all quizzes in their k^{th} attempt. To abbreviate, we use $\mathcal{Y}_k$ to represent the k^{th} frontal slice of all tensors. Accordingly, $\mathcal{Y}_{i,:,:}$ shows all the attempts of student i on all questions and $\mathcal{Y}_{:,j,:}$ shows all attempts of all students on question j. We assume that each quiz consists of multiple (c) concepts (skills or knowledge components) and that the students should have some knowledge of these concepts in order to solve the quizzes that include such concepts. Some of the elements of $\mathcal{Y}$ are unknown to us because not all of the students try all of the questions as many times. Based on these assumptions, we formulate the problem as a tensor factorization with two phases: the *prediction* phase and the *learning* phase.

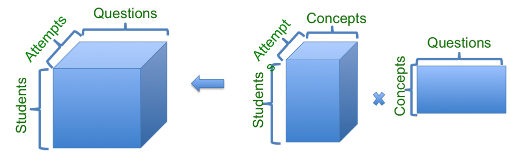

Figure 1: Phase 1: Decomposition of Student Performance into Student Knowledge and Concept-Map

In the prediction phase, we follow the assumption that students' success or failure in quizzes depends on their knowledge and the concepts underlying those quizzes. In this phase, we decompose $\mathcal{Y}$ into a tensor and a matrix: the tensor $\mathcal{T}$ that shows the knowledge of students on the concepts at each of their attempts on the quizzes, and the matrix $\mathbf{Q}$ that shows the concepts that are required to solve each quiz correctly. For each quiz j, $\mathbf{Q}_{:,j}$ shows the importance of each of the discovered concepts in it. Also, $\mathcal{T}_{i,k,l}$ shows the knowledge of student i in concept k at the l^{th} attempt.

Based on this decomposition, we can estimate (predict) the unknown values of $\mathcal{Y}$ using the multiplication of tensor $\mathcal{T}$ and matrix $\mathbf{Q}$, as presented in Equation 1. Figure 1 gives an illustration of this decomposition.

$$\mathcal{Y} = \mathcal{T} \times \mathbf{Q} \tag{1}$$

We suppose that students learn by practicing the quizzes, and that the knowledge of students increases through this practice of the concepts. The learning phase of our tensor factorization approach models student learning, based on the quizzes that they choose to solve in each step. In order to do that, we construct a tensor $\mathcal{X}$ that denotes when a student has or has not chosen to work on a specific problem at a specific time. Equation 2 shows how to build this tensor, based on $\mathcal{Y}$.

$$\mathcal{X}_{i,j,k} = \begin{cases} 1, & \text{if } \mathcal{Y}_{i,j,k} \text{ is observed} \\ 0, & \text{if } \mathcal{Y}_{i,j,k} \text{ is not observed} \end{cases} \tag{2}$$

In the learning phase, we assume that the amount of gained knowledge in each concept is a function of the student's knowledge at the previous attempt, as well as the weight of concepts that are learned in the quiz that the student chooses to solve. Let $f(\cdot)$ be such a function; then the gained knowledge at time t can be expressed as:

$$\mathcal{T}_t = f(\mathcal{T}_{t-1}, \mathcal{X}_t, \mathbf{Q})$$

Since we assume that knowledge of students grows over time, we should choose a monotonically increasing function for

$f(\cdot)$. Also, to keep this knowledge increase from growing too large, this function should be bounded. Based on these assumptions, we model the knowledge growth of students as a logistic regression function that ranges between 0 (for no increase in the knowledge) to $1 - \mathcal{T}_{t-1}$ (for a maximum increase in the knowledge). This allows us to have a bounded amount of knowledge that always stays between zero and one. To add to the flexibility of this function, and to account for different students' rate for learning from the quizzes, we add a factor μ that controls the slope of the logistic regression function. The higher the learning rate (μ), the larger the knowledge increase and the faster the students reach a maximum state of knowledge. This increase can be seen in Equation 3.

$$\mathcal{T}_t = \mathcal{T}_{t-1} + \left(\frac{2(1 - \mathcal{T}_{t-1})}{1 + exp(-\mu \mathcal{X}_t \mathbf{Q}')} - (1 - \mathcal{T}_{t-1})\right), \qquad (3)$$

which can be written as follows:

$$\mathcal{T}_t = 2\mathcal{T}_{t-1} + \frac{2(1 - \mathcal{T}_{t-1})}{1 + exp(-\mu \mathcal{X}_t \mathbf{Q}')} - 1 \qquad (4)$$

Based on this model, the more knowledgeable the student is in a concept, the less improvement she will obtain by practicing the same concepts again and again. The greatest increase in the student's knowledge happens when the student does not know the skills that are provided in the quiz. If we expand and simplify Equation 3, we achieve Equation 4. Since $f(\cdot)$ is a monotonically increasing function, the estimated knowledge tensor ($\mathcal{T}$) and domain model ($\mathbf{Q}$) are both non-negative. This non-negativity is in accordance with assumptions in the educational domain: that the weight of each concept in each learning material cannot be negative and that the knowledge of students at any time and in any concept cannot be negative either.

Eventually, the matrix factorization includes solving Equations 1 and 4. Assuming that we have the values for $\mathcal{X}_t$ and $\mathbf{Q}$, Equation 4 can be considered as a static update and we can only optimize Equation 1 iteratively and update the knowledge values in each iteration using Equation 4. To achieve this goal, we try to optimize for the least regularized estimation error of our observed tensor ($\mathcal{Y}$) in Equation 5. Thus, our objective is to minimize the overall error, which is defined as:

$$\Sigma_{i=1}^{t} \parallel \mathcal{Y}_t - \mathcal{T}_t \mathbf{Q} \parallel^2 + \lambda(\Sigma_{i=1}^{t} \parallel \mathcal{T}_i \parallel^2 + \parallel \mathbf{Q} \parallel^2), \qquad (5)$$

where λ is a regularization parameter. The last two terms are added to the error equation to regularize the values in tensor $\mathcal{T}$ and matrix $\mathbf{Q}$. These two terms increase the sparsity of the knowledge and domain model by decreasing the values in these two factors, while preventing the factorization from being over-fit to the training data.

Since this method uses the iterative feedback loops and the two phases of prediction and learning, we name it Feedback-Driven Tensor Factorization (FDTF).

3. EXPERIMENTS

To asses the student performance prediction task, we compare the proposed FDTF model to a baseline tensor factorization algorithm that was introduced in previous recommender system literature. This tensor factorization algorithm is called the Bayesian Probabilistic Tensor Factorization (BPTF) and models the temporal change of user interests on items [10]. We choose this model as a baseline because of its consideration for time sequencing and the common use of recommender systems algorithms in the educational data mining literature [7]. As our second baseline, we run the Bayesian Knowledge Tracing (BKT) algorithm on the data [1]. Since BKT requires a pre-defined set of concepts, we use the manually-labeled concepts that have been discovered by experts in this case.

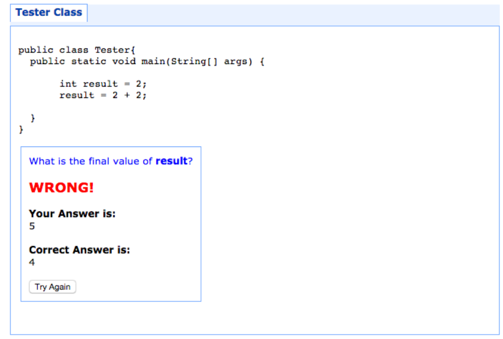

Figure 2: Screen-shot of QuizJet System

The FDTF algorithm has two parameters that need to be tuned: the number of concepts (c) and the learning rate of students (μ). We define these two parameters through cross-validation. Also, in our experiments, we set $\lambda = 0.0001$.

3.1 Dataset and Setup

We use student sequences of the QuizJet online self-assessment system to run our experiments [2]. This system produces parameterized Java quizzes based on a set of predefined templates. Hence, each student can repeat the same Java quiz, with different parameters, over and over again. The students submit their answer using a text box provided in the user interface and can receive immediate feedback. Figure 2 shows a screen-shot of this system in use.

The dataset was collected from the students who have taken a Java programming course from Fall 2010 to Spring 2013 (six semesters). The system was introduced in the class and students have voluntarily interacted with this system. The subject domain is organized by experts into 22 coherent topics. Each topic has several questions and each question is assigned to one topic. We use these sets of topics as the expert-labeled domain model in our experiments.

We experimented on $27,302$ records of 166 students on 103 questions. The average number of attempts on each question is equal to three. Our dataset is imbalanced: the total number of successful attempts in the data equals $18,848$ (69.04%) and the total number of failed attempts is 8454. We used a user-stratified 5-fold cross-validation to split the data so that the training set has 80% of the users (with all their records) randomly selected from the original dataset, while the remaining 20% of the users were retained for testing. In other words, 80% of students are in the training

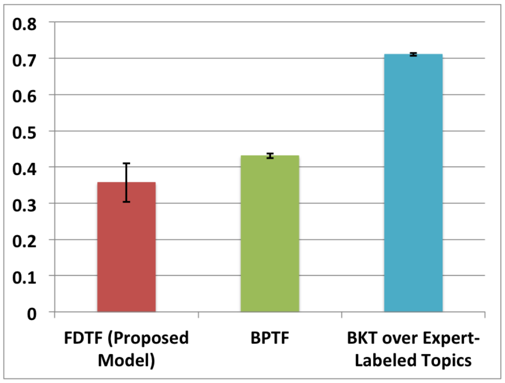

Figure 3: RMSE of Algorithms for Predicting Students Performance

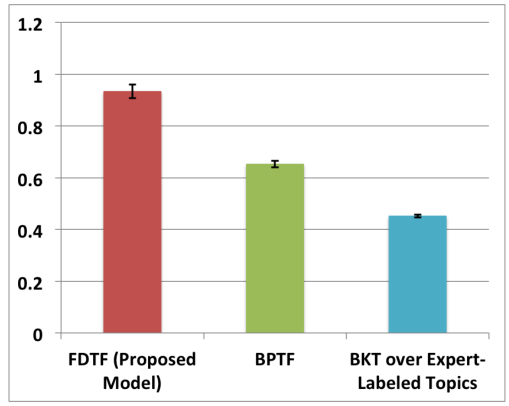

Figure 4: Accuracy of Algorithms for Predicting Students Performance

set and we have all of their sequences. For the remaining students (20%) we use 20% of their data to predict the rest 80% of it. Eventually, we include $80\% + 20\% * 20\% = 84\%$ of the whole dataset in the training set. We used the same set of data for all of the algorithms. We ran the experiments 3 times per stratification, and ended up with running each algorithm 15 times. The simple statistics of our dataset are shown in Table 1.

Table 1: Dataset Statistics

	Average	Min	Max
#attempts per sequence	3	1	50
#attempts per question	265	25	582
#attempts per student	165	2	772
#different students per question	87	7	142
#different questions per student	54	1	101

To find the best number of concepts (c) in each of the automatic PSP algorithms, we use cross-validation.

3.2 Experimental Results

As explained in Section 3, we examine the prediction performance of the proposed FDTF algorithm and the baseline models BPTF and BKT with expert-labeled topics. We then compare the accuracy of these three approaches. Since the dataset is imbalanced with approximately 70% positive labels and 30% negative labels, we define predicted values that are greater than 0.3 as positive-label predictions and predicted values that are less than or equal to 0.3 as negative-label predictions. Figure 4 shows the accuracy of the mentioned algorithms. The red, green, and cyan bars represent the accuracy of FTDF, BPTF, and BKT. As we can see in this figure, although the accuracy of the baseline tensor factorization model (BPTF) is better than Bayesian Knowledge Tracing, it is significantly less than the accuracy of the proposed approach (FDTF). Eventually, FDTF performs significantly better than both of the baseline algorithms.

Although the task of predicting student performance is a binary classification task in this setting (predicting either failure or success for students), the Root Mean Squared Error (RMSE) is traditionally used to evaluate this task in the literature. As a result, we compare the approaches based on the RMSE of approaches in addition to their accuracy. Figure 3 shows RMSE of these experiments for each of the approaches. Again, we can see that FDTF has a significantly better RMSE than both the BKT and BPTF algorithms.

These results show that, even though BKT adds the knowledge of topic-based domain model, the tensor factorization algorithms outperform it. Additionally, despite the facts that both BPTF and FDTF use the same data, model the student data as a tensor, and are temporal tensor factorization approaches, the proposed FDTF approach performs better than BPTF. These results show that explicitly modeling students' knowledge acquisition by considering their interactions with learning materials leads to better overall modeling of student knowledge, and thus provide a better overall prediction of student performance.

4. CONCLUSIONS AND FUTURE WORK

We proposed a novel tensor factorization model (FDTF) that can predict students' success or failure in future quizzes by explicitly modeling their knowledge acquisition during their interaction with learning materials. This approach does not require any expert or domain knowledge and can be automatically performed using students' historical attempt sequence. Our evaluations show that FDTF outperforms the predicting student performance approaches in the literature.

In future, we plan to explore the ability of the proposed approach in discovering the underlying domain model for the learning material, experiment on more diverse datasets, and compare our algorithm to other PSP and domain modeling approaches in the literature. We plan to improve our FDTF model to be able to model implicit feedback of students' activity, in addition to providing overall success and failure records.

The FDTF model has the potential to be used as a basis to recommend learning material to students. Also, it can help teachers discover domain models and edit or enhance learning materials, look up the concepts that students struggle to learn, and suggest appropriate learning activities.

5. ACKNOWLEDGMENTS

This work is partially supported by the Advanced Distributed Learning (ADL) Initiative (contract W911QY-13-C-0032).

6. REFERENCES

[1] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[2] I.-H. Hsiao, S. Sosnovsky, and P. Brusilovsky. Adaptive navigation support for parameterized questions in object-oriented programming. In *Learning in the Synergy of Multiple Disciplines*, pages 88–98. Springer, 2009.

[3] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review - Society for Industrial and Applied Mathematics*, 51(3):455–500, 2009.

[4] A. S. Lan, C. Studer, and R. G. Baraniuk. Quantized matrix completion for personalized learning. In *The 7th International Conference on Educational Data Mining (EDM)*, London, July 2014.

[5] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. In *14th International Conference on Artificial Intelligence in Education*, volume 2009, pages 531–538, 2009.

[6] S. Sahebi, Y. Huang, and P. Brusilovsky. Parameterized exercises in java programming: using knowledge structure for performance prediction. In *The second Workshop on AI-supported Education for Computer Science (AIEDCS)*, pages 61–70. University of Pittsburgh, 2014.

[7] S. Sahebi, Y. Huang, and P. Brusilovsky. Predicting student performance in solving parameterized exercises. In *Intelligent Tutoring Systems*, pages 496–503. Springer, 2014.

[8] N. Thai-Nghe, L. Drumond, T. Horváth, A. Nanopoulos, and L. Schmidt-Thieme. Matrix and tensor factorization for predicting student performance. In *the International Conference on Computer Supported Education*, pages 69–78. Citeseer, 2011.

[9] N. Thai-Nghe, T. Horvath, and L. Schmidt-Thieme. Context-aware factorization for personalized student's task recommendation. In *Proceedings of the International Workshop on Personalization Approaches in Learning Environments*, volume 732, pages 13–18, 2011.

[10] L. Xiong, X. Chen, T.-K. Huang, J. G. Schneider, and J. G. Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, volume 10, pages 211–222, 2010.