ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/372224157

Exploring the Individual Differences in Multidimensional Evolution of Knowledge States of Learners

Chapter · July 2023

DOI: 10.1007/978-3-031-34735-1_19

CITATIONS
0

READS
7

6 authors, including:

Liang Zhang
The University of Memphis
10 PUBLICATIONS 12 CITATIONS
SEE PROFILE

Phil Pavlik Jr
The University of Memphis
87 PUBLICATIONS 1,613 CITATIONS
SEE PROFILE

Xiangen Hu
The University of Memphis
155 PUBLICATIONS 2,612 CITATIONS
SEE PROFILE

Jody Cockroft
The University of Memphis
28 PUBLICATIONS 2,058 CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Operation ARA View project

Project Intelligent Tutoring Systems View project

All content following this page was uploaded by Liang Zhang on 15 July 2023.

The user has requested enhancement of the downloaded file.

Exploring the Individual Differences in Multidimensional Evolution of Knowledge States of Learners

Liang Zhang[1,2(✉)], Philip I. Pavlik Jr.[1,3], Xiangen Hu[1,2,3,4], Jody L. Cockroft[1,3], Lijia Wang[1,2], and Genghu Shi[5]

[1] Institute for Intelligent Systems, University of Memphis, Memphis, TN 38152, USA
{lzhang13,ppavlik,xhu,lwang3}@memphis.edu

[2] Department of Electrical and Computer Engineering, University of Memphis, Memphis, TN 38152, USA

[3] Department of Psychology, University of Memphis, Memphis, TN 38152, USA

[4] School of Psychology, Central China Normal University, Wuhan 430079, Hubei, China

[5] Department of Psychology, Guangxi Normal University, Guilin 541000, Guangxi, China

Abstract. The key to the effectiveness of Intelligent Tutoring Systems (ITSs) is to fit the uncertainty of each learner's performance in performing different learning tasks. Throughout the tutoring and learning process, the uncertainty of learners' performance can reflect their varying knowledge states, which can arise from individual differences in learning characteristics and capacities. In this investigation, we proposed a multidimensional representation of the evolution of knowledge states of learners to better understand individual differences among them. This assumption about this representation is verified using the Tensor Factorization (TF) based method, a modern state-of-the-art model for knowledge tracing. The accuracy of the Tensor-based method is evaluated by comparing it to other knowledge-tracing methods, to gain a deeper insight into individual differences among learners and their learning of diverse contents. The experimental data under focus in our investigation is derived from the AutoTutor lessons that were developed for the Center for the Study of Adult Literacy (CSAL), which employs a trialogue design comprising of a virtual tutor, a virtual companion and a human learner. A broader merit of our proposed approach lies in its capability to capture individual differences more accurately, without requiring any changes in the real-world implementation of ITSs.

Keywords: Intelligent tutoring systems · Knowledge tracing · Knowledge states of learners · Individual differences · Tensor-based method · Tutoring · Learning process

1 Introduction

Intelligent Tutoring System (ITSs) are deemed as intelligent due to their capabilities to capture various uncertainties of learners and ensure flexible and adaptive

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
R. A. Sottilare and J. Schwarz (Eds.): HCII 2023, LNCS 14044, pp. 265–284, 2023.
https://doi.org/10.1007/978-3-031-34735-1_19

interactions tailored to each individual needs [1–4]. The sources of uncertainties can arise both explicitly and implicitly from various dimensions, such as the learner's background, skills, diverse experience in the learning domain, learning characteristics, learning styles, and other psychological characteristics that cause individual differences in the learning process, learning ability or performance of learners [1,5–7]. Generally, modeling that relies on learning data can be a powerful and commonly used way to quantify sources of uncertainties in both academic and industry settings. The fluctuations of parameters obtained in such models to some extend reflect the physical sense of individual differences among learners, and can potentially serve as a reference for the real-world implementation of ITSs. However, such modeling work remains a challenging task due to the complexity of human learning nature (involves acquiring knowledge under the effects of task difficulties, memory, time, practice, sequence, etc.), the particularities for different domain knowledge, the existing uncertainties from the environment and learners, as well as instructional interventions [8–10].

Sensing and capturing the individual differences of each learner in modeling is crucial for achieving adaptiveness in ITSs. AI tools and techniques (based on the machine learning within our research scope) can contribute to the modeling work (may focus on certain or specific aspects of learning) by providing mathematical descriptions of the knowledge learning states of learners, as well as reasoning and problems-solving mechanisms along the tutoring/learning progress [5]. This can typically be achieved through the use of knowledge tracing approaches that allow the system to trace changes in the learner's knowledge state during learning, to implement probability estimates of the learner's performance, and guide pedagogical decisions based on mastery learning principles [11–14]. The knowledge tracing approaches basically assume that the learner' prior performance can help in predicting the performance of the learner on subsequent tasks. For example, the sequence of prior performance on practicing items for each learner is utilized for estimating the probability of answering each subsequent item correctly [15]. Usually, many researchers use elementary fragments of domain knowledge, like the concepts or concept-like elements, the Knowledge Units (KUs), Knowledge Components (KCs), or other question-related knowledge items if KCs are not specified, to generically constitute the implication relations or internal links among those items in domain knowledge [15–17]. They will serve as the knowledge of how to apply learning proficiency to estimate the probability of success when the learner responds to practice items in knowledge tracing. At the same time, the intelligent tutor can present adaptive instructions based on the probability prediction of the learner's performance, and aims to reach the goal of bolstering the learner's evolving knowledge state to converge to the wholly master states (or some pre-specified criterion, e.g. mastery criterion of probability of 0.95 [11,18]) in the targeted domain.

The state-of-the-art models for knowledge tracing can be generally categorized as factor analysis models and Bayesian Knowledge Tracing (BKT) [19]. The models within the family of factor analysis can be traced back to the development of Item Theory Response (IRT) [15,20]. The core idea of IRT is to estimate the probability of student success on the test items based on various factors of that item, typically using a logistic function. Usually, the skills, concepts, knowledge, or cognitive operations that are considered necessary for answering the item correctly are represented in the form of Q-matrices [15,21]. Some prototypes of the factor analysis models are the Additive Factor Model (AFM) [22] and Performance Factor Analysis (PFA) [23]. The BKT is theoretically supported by the hidden Markov model and Bayesian Belief Network. BKT estimates the learner's mastery of a skill and predicts the probability of the learner's success with a binary knowledge state (the learned state and unlearned state) [11]. Many BKT variants may include Individualized BKT [24], Dynamic BKT Model [25], and so on.

Moreover, some advanced methods for knowledge tracing are recently proposed, like the SPARse Factor Analysis (SPARFA) [26] and Tensor Factorization (TF) based method [27]. The SPARFA uses the quantized matrix completion to predict student performance in knowledge tracing. Specifically, the SPARFA represents the probability of answering a question successfully in terms of three factors: 1) the learner's knowledge of a set of latent or underlying concepts, 2) the connections between the question and concepts, and 3) the intrinsic difficulty of each question [26]. The TF based method refers to structurally represent student knowledge in three-dimensional space by taking into account the critical factors (e.g. learners, problems, and attempts) that significantly influence the learning progress. It implements probability estimates on student performance via tensor factorization mathematically. Some of TF variants in knowledge tracing are three-dimensional Bayesian Probability Tensor Factorization (3D-BPTF) [28], Feedback-Driven Tensor Factorization (FDTF) [27], Rank-Based Tensor Factorization (RBTF) [29].

The TF-based method is highlighted in this study. In an ITS learning environment, the learners' performance data can be captured, collected and reorganized in multiple ways to express their learning state and track their learning progress over time, which can be then utilized for modeling or other statistical analyses. In the TF-based method, one 3-dimensional tensor factorization framework can be built by involving the most critical factors of learners, attempts, and questions, for deeply exploring and exploiting their ensemble effects in the learning process. Some reasons for this choice include: 1) the capability of this method to handle the multiple dimensionalities of the factors (from learners, attempts or time, and questions) that critically affect the performance, 2) the scalability in the "elastic" size of these dimensions, 3) its generality in exploring and exploiting the learning model accurately (will be verified by the results) [30]. Additionally, the tensor-based construction is capable to avoid the "distortion" of effects from the sequence in modeling. Therefore, this proposed approach lay the foundation for us to explore the individual differences on the basis of accurate modeling, fidelity and multidimensionality.

Lastly, it is worth noting that the modeling work heavily relies on the experimental data from real scenarios as it can provide us lots of information about the learners' performance. The modeling work involves evaluating multiple candidate models, and the model that best fits the pattern of knowledge learning observed in existing experimental data is selected for further research.

2 A Brief Review of Study Case: CSAL AutoTutor

In this investigation, the Center for the Study of Adult Literacy (CSAL) AutoTutor, one example expectation-misconception-tailed (EMT) dialogue ITS, is under focus. The CSAl AutoTutor employs the trialogues design, which includes two computer agents (virtual tutor and virtual learner) and one human learner [31]. Figure 1(a) shows a screenshot of the interface for CSAL AutoTutor from the lesson on the topic of "Cause and Effect". The actions of learner while interacting with CSAL AutoTutor can involve a range of physical actions such as clicking, scrolling, dragging, and dropping on the exchange surface, and the system tracks the learner's performance based on the selected menu alternative at each turn in the conversation when a response is expected (this belongs to the categorical responses) [3,31–33].

The diagram for calling questions queue by hierarchical task difficulty levels (including intermediate, easy and hard, which are computed by the Coh-Metrix system [31,34,35]) in the system is shown in Fig. 3(b). After the human learner completes reading the entire text in the specific CSAL reading comprehension lesson, the questions will be displayed. The question at the intermediate difficulty level, denoted as $(Q_{Inter\_1}, Q_{Inter\_2}, Q_{Inter\_3}, ...)$, will be retrieved first from the question queue and presented the human learner. The human learner will then respond to the question within the context of the text, and provide categorical answers by clicking the relevant selection buttons provided. The human learner's performance on the questions can be assessed by comparing their answers with the expected ones, and their score will calculated sequentially. Once the intermediate queue is completed, there will be a switch to another one. If the match scores meet to or exceed the threshold of expectation (upgrade), the call is directed to the hard level queue $(Q_{Hard\_1}, Q_{Hard\_2}, Q_{Hard\_3}, ...)$. Otherwise, the call will be directed to the easy level question queue $(Q_{Easy\_1}, Q_{Easy\_2}, Q_{Easy\_3}, ...)$ for a downgrade. The response sequence of each individual human learner's performance (usually consisting of categorical responses such as correct or incorrect in CSAL AutuTutor) is generated sequentially as the question queue runs. This is the diagram commonly used to illustrate the process of calling different task difficulty levels of question queues for most CSAL lessons.

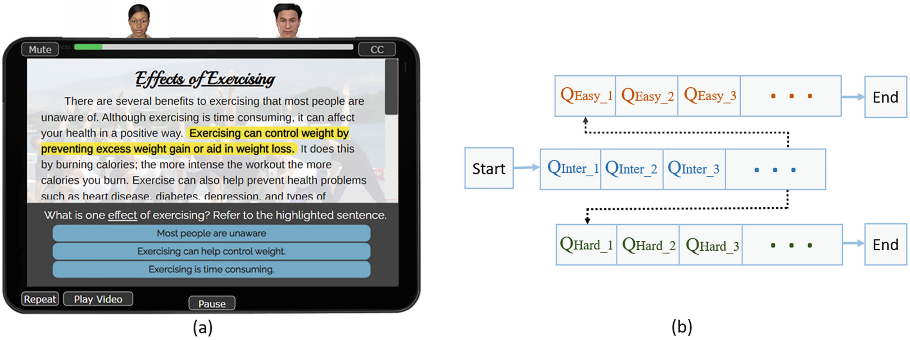

Fig. 1. CSAL AutoTutor: (a) screenshoot of interface, (b) diagram for calling different difficulty levels of questions queue

3 Multidimensional Evolution of Knowledge States of Learners for CSAL AutoTutor

As mentioned previously, the TF-based method introduces a multidimensional perspective to describe human learning. The similar perspective was originally from the book edited by Newell and Simon five decades ago [36]. It seeks to decode human learning as three-dimensional space, which includes the task dimension (different classes of task environments), performance-learning-development dimension (activities related to performing, learning, and developing are correlated with time scale), and the individual-difference dimension (various populations with a varying difference) in an evolutionary sense. In this study, we adopt this idea as a starting point and extract implementable concepts to instantiate a three-dimensional space based on the dimensions of learners, questions, and attempts in real learning scenarios of ITS. And we aim to present a structural and systematical framework that incorporates modern development of knowledge tracing methodologies to enable deeper and more extensive study of human learning (mainly focus on the individual differences in this paper).

As discussed earlier, in the CSAL AutoTutor system, the generation of a sequence of the learner's responses (performances in EMT-based ITSs) follows the running of the questions queue progressively. The elementary fragments of domain knowledge, like concepts or concept-like elements, KUs, KCs, or question-related items if the KCs are not specified, constitute the knowledge domain, which can be represented as [16,17]:

$$\Delta = \{\delta_1, \delta_2, \delta_3, ..., \delta_n\} \tag{3.1}$$

where the Δ specifies the knowledge domain and the δ refers to the KC or KU (if labeled, else question item). Note that KCs can be identified or refined by the "manual" approaches by the domain expert, some automated methods, or semi-automated methods; there are many examples of research on the identification of the KCs in different domains [37–39], but it is beyond the scope of this investigation to make these efforts.

The estimation of performance or outcome for one specific question or KC is definitely probabilistic by the individual difference of the learner's knowledge states. Stated differently, different knowledge states will contribute to different probabilities for enabling the learner to answer the question correctly or master the KC successfully (there is also the possibility of failure in performing because of slip or forget reasons). Accordingly, the estimation of mastery of the domain Δ for some learner (e.g. l_1) is predicted by probability as:

$$P(\Lambda_{l_1}^t) = \{P(\delta_1), P(\delta_2), P(\delta_3), ..., P(\delta_n)\} \tag{3.2}$$

For example, in the CSAL AutoTutor system, the learner's mastery of the domain can be estimated as:

$$\Lambda_{l_1}^t = \{0.52, 0.63, 0.85, 0.91, 0.92, ..., 0.93, 0.95\} \tag{3.3}$$

By encoding the actual performance of a learner sequentially as an array of binary values (1 for correct and 0 for incorrect), we can represent the learner's knowledge state as:

$$\Lambda_{l_1}^t = \{1, 1, 0, 1, 1, ..., 0, 1\} \tag{3.4}$$

The evolving knowledge state of each learner, encompassing all moments and questions, for each learner can be defined as a matrix frame:

$$\Lambda_{l_1} = [P_{i,j}]_{n \times m} = \begin{Bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ \vdots \\ Y_i \\ \vdots \\ Y_n \end{Bmatrix} = \begin{pmatrix} P_{11} & P_{12} & P_{13} & \cdots & P_{1m} \\ P_{21} & P_{22} & P_{23} & \cdots & P_{2m} \\ P_{31} & P_{32} & P_{33} & \cdots & P_{3m} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & P_{n3} & \cdots & P_{nm} \end{pmatrix} \tag{3.5}$$

where the P_{ij} is determined by predictions of the observed probability of the individual learner for mastering the i^{th} question (or associated KCs) at the j^{th} moment or attempt. The row vectors $\{Y_1, Y_2, Y_3, \cdots, Y_i, \cdots, Y_n\}^T$, also the observation vectors of Λ_{l_1}, demonstrate the learner's performance in the dimension of questions or KCs by some sequential order.

Taking the multiple learners into account, we can extend the two-dimensional matrix mentioned in (3.6) into three-dimensional tensor (including dimensions of learners, questions or KCs, and time or attempts).

If the domain of learners L (including total u learners) can be represented as:

$$L = \{l_1, l_2, l_3, \cdots, l_u\} \tag{3.6}$$

And accordingly, the tensor-based knowledge states of learners T for the learners' domain L is:

$$T = [\Lambda_{l_1}, \Lambda_{l_2}, \Lambda_{l_3}, \cdots, \Lambda_{l_u}] \tag{3.7}$$

4 Exploration of Models Based on Knowledge Tracing

To regularize the modeling work and improve the interpretability of all entries in modeling, two fundamental restricting assumptions are given:

- Non-negativity: Negative impact of the learner's knowledge on the probability of successfully answering present questions is not allowed. In other words, the entries about the quantized performance data (probability values) are non-negative.
- Sparsity: Data sparsity inevitably exists in the experimental dataset for modeling. For example, there may be some zero entries in modeling when students are incomplete or miss performance in answering questions.

4.1 About the Approaches

Four typical knowledge tracing models are to be explored in this study: 1) TF-based method, 2) PFA, 3) BKT, 4) SPARFA-Lite (one variant of the SPARFA).

(1) TF-based method: The-TF based method relies on the 3-dimensional tensor to represent students performance. Given a 3-dimensional tensor $T \in R^{I \times S \times J}$, the three dimensions donate the total number of learners (I), the total number of attempts (S), and the total number of questions (J) separately. Each cell τ_{isj} of T represents the performance variable of student l_i on question q_j at attempt a_s, e.g. $\tau_{isj} = \{0, 1\}$ with 1 representing a correct answer and 0 an incorrect answer in the CSAL AutoTutor case.

To obtain the third order $\hat{T}$, we define a vector U and a matrix V (as shown in Figure 7) [27],

$$\hat{T} \approx U \otimes V \tag{4.1}$$

where $U \in R^{I \times S \times C}$ and $V \in R^{C \times J}$. In this decomposition (Tucker Decomposition), the tensor U represents the knowledge of learners on the latent KCs at each attempt on questions, the matrix V represents the latent KCs required for solving each question, and the obtained $\hat{T}$ is the approximation of the T constructed by real data (its accuracy is evaluated by best-fit computing mathematically). Here we use vector $u_{is:}$ of U to represent the KCs required for the student l_i to answer questions at the attempt a_s, and the vector $v_{:j}$ to represent the latent KCs vector on the question q_j.

If we take into account bias from these factors of learner's ability, question difficulty, or student cohort strength, we add the learner, question, and attempt biases (b_l, b_a, b_q) in addition to an overall cohort bias (μ) to our above mentioned model, so the estimated $\hat{\tau}_{isj}$ is [29]:

$$\hat{\tau}_{isj} \approx u_{is:} \cdot v_{:j} + b_l + b_a + b_q + \mu \tag{4.2}$$

The objective function can be minimized as [29]:

$$\xi_1 = \sum_{isj} (\hat{\tau}_{isj} - \tau_{isj})^2 + \lambda(b_l^2 + b_a^2 + b_q^2) + \lambda_1 \parallel u_{is:} \parallel^2 + \lambda_2 \parallel v_{:j} \parallel^2 \tag{4.3}$$

We assign the cohort bias μ as $\mu = \frac{\sum_{isj} \tau_{isj}}{\sum_{isj} \Gamma(isj)}$, where $\Gamma(i, s, j)$ is an indicator function returning 1 if the tuple (i, s, j) is in our training set; otherwise 0 [29]. This is the regular type of TF-based method.

(2) PFA: PFA is a logistic regression model that predicts the probability of a learner's answer response on a question as a function of the learner's learning ability, KC-related features (e.g. difficulty), and previous success and failures [15,23,40–42]. One variant of PFA is given [15,42]:

$$logit(p_{ikj}) = \theta_i + \theta_j + \gamma_{ik} S_{ikj} + \rho_{ik} F_{ikj} \tag{4.4}$$

where i, j, and k represent the learner, attempts and KC separately. The θ_i is the coefficient for the learner i (about learning proficiency) and the θ_j the coefficient for KC k (about difficulty). The S_{ikj} refers to the number of prior successes the learner i has had on the KC k. The F_{ikj} refers to the number of prior failures the learner i has had on the KC k. The γ_{ik} the coefficient for the benefit of previous success on the KC k for the learner i. The ρ_{ik} becomes the coefficient for the benefit of previous failures on the KC k for the learner i. And the $logit(p_{ikj})$: is usually obtained by $ln(\frac{p_{ikj}}{1-p_{ikj}})$.

(3) BKT: BKT estimates the learner's master level on a skill and predicts the probability of a learner's binary response (correct or incorrect) in a binary state (the learned state or the unlearned state) [11]. It uses the Bayesian network to make the learner's performance up to one point linked by four parameters probabilistically [11,13,43–46]. Four important notations in BKT are given:

- $P(L_0)$: the initial or prior probability of mastering the skill for the learner.
- $P(T)$: the probability of acquiring or learning knowledge the learner by transforming from the unmastered state on one skill to the master state.
- $P(S)$: the probability of making an incorrect answer response by slipping in the mastered state on a skill for the learner.
- $P(G)$: the probability of making a correct answer response by guessing in an unmastered state on a skill for the learner.

Note that we define $P(L_l)$ as the probability of mastering the skill for a learner at attempt l. In CSAL AutoTutor case, we use $P(L_l \mid O_l)$ as the probability of learning a skill for the learner based on the learner's previous response, where $O_l \in \{0, 1\}$. The O_l is 1 if the answer response is correct and 0 if incorrect. So the $P(L_l \mid O_l)$ can be calculated through following equation [43–46]:

$$P(L_{l-1} = 1 \mid O_l = 0) = \frac{P(L_{l-1} = 1) * (1 - P(S))}{P(L_{l-1} = 1) * (1 - P(S)) + (1 - P(L_{l-1} = 1)) * P(G)} \tag{4.5}$$

$$P(L_{l-1} = 1 \mid O_l = 1) = \frac{P(L_{l-1} = 1) * P(S)}{P(L_{l-1} = 1) * P(S) + (1 - P(L_{l-1} = 1)) * (1 - P(G))} \tag{4.6}$$

$$P(L_{l-1} = 1 \mid O_l) = P(L_{l-1} = 1 \mid O_l) + (1 - P(L_{l-1} = 1 \mid O_l)) * P(T) \tag{4.7}$$

(4) SPARFA-Lite: The SPARFA-Lite is the variant of the SPARse Factor Analysis (SPARFA). The SPARFA-Lite leverages matrix completion to analyze the quantized graded learner responses, and automatically identify the required number of KCs [47]. The "Lite" means low computational complexity as compared to the conventional SPARFA [47]. The prediction of learners' performance is driven by the exploration of the number of KCs in SPARFA-Lite.

Suppose that the unknown and low-rank matrix Z represents the learners' responses to questions. Since it's a two-dimensional level, the entries of Z can represent overall quantized measurements by taking into account all responses through all attempts for the specified question.

Let $Y_{ij} \in \mathcal{O}$, where $\mathcal{O} = \{\ 1, \cdots, P\}$ is a set of P ordered labels. Inspired by [47,48], we use the following model for the observed response $Y_{i,j}$:

$$Y_{i,j} = \mathcal{Q}(Z_{ij} + \epsilon_{ij}),\ and\ \epsilon_{ij} \sim Logistic(0, 1) \tag{4.8}$$

where $\{\omega_0, \cdots, \omega_P\}$ is a set of quantization bin boundaries, with $\omega_0 \leq \omega_1 \leq \cdots \leq \omega_{P-1} \leq \omega_P$. The quantization bin boundaries $\{\omega_0, \cdots, \omega_P\}$ is assumed to known a priori.

In terms of the likelihood of the observed graded leaner response Y_{ij}, the model in (1) can be written equivalently as

$$p(Y_{ij} = p \mid Z_{ij}) = \Phi(w_p - Z_{ij}) - \Phi(w_{p-1} - Z_{ij}) \tag{4.9}$$

where $\Phi(x) = \frac{1}{1+e^{-x}}$ corresponds to the inverse link function.

In order to recover the low-rank matrix Z, we try to minimize the negative log-likelihood of the observed graded measurement of learner response $Y_{i,j} \in \Omega_{obs}$. The optimization problem can be described as following:

$$\begin{cases} minimize\ f(Z) = -\sum_{ij:(i,j)\in\Omega_{obs}} \log p(Y_{ij} \mid Z_{ij}) \\ subject\ to\ \|Z\| \leq \lambda \end{cases} \tag{4.10}$$

This optimization can be solved efficiently via the FISTA framework [49]. Here, the constant $\|Z\| \leq \lambda$ is used to promote the low-rank solution Z and the parameter $\lambda > 0$ is used to control its rank. In practice, the nuclear norm constant $\|Z\|_* \leq \lambda$ is applied here.

The gradient step is given by $\hat{Z}^{l+1} \leftarrow Z^l - s_l \nabla f$, where the s_l is the step-size at iteration l. For simplicity, s_l is the step-size $s_l = 1/L$, where the L is the Lipschitz constant, which is given by $L_{log} = 1/4$ for the inverse logit link.

The gradient of the objective function $f(Z)$ with respect to Z is given by:

$$[\nabla f]_{ij} = \begin{cases} \frac{\Phi'(L_{ij}-Z_{ij})-\Phi'(U_{ij}-Z_{ij})}{\Phi(U_{ij}-Z_{ij})-\Phi(L_{ij}-Z_{ij})},\ if\ (i,j) \in \Omega_{obs} \\ 0,\ otherwise \end{cases} \tag{4.11}$$

where the derivative of the inverse logit link function corresponds to $\Phi'(x) = \frac{1}{2+e^{-x}+e^{x}}$. The $Q \times N$ matrices U and L contain the upper and lower bin boundaries corresponding to the measurements Y_{ij}, i.e. we have $\omega_{Y_{ij}}$ and $L_{ij} = \omega_{Y_{ij}-1}$.

The projection step imposes low-rankness on Z. It's also the regularization preventing overfitting. The nuclear norm constant case $\|Z\|_* \leq \lambda$, this step requires a projection onto the nuclear norm ball with radius λ, which can be performed by first computing the SVD of Z followed by projecting the vector of singular values onto an l_1-norm ball with radius λ [50].

$$Z^{l+1} \leftarrow \tilde{U}diag(s)\tilde{V}^T, with\ s = P_\lambda(diag(S)) \tag{4.12}$$

where the $\tilde{U}diag(s)\tilde{V}^T$ denotes the SVD of the $\hat{Z}^{l+1}$. The operator $P_\lambda(\cdot)$ denotes the projection of a vector onto l_1-norm ball with radius λ, which can be computed at low complexity. In our study, the biased matrices that implicitly contain the intrinsic question difficulty and KCs are considered for Eq. 4.10.

If we define the $Q \times N$ matrix A with $A_{ij} = \Phi(Z_{ij}) \in [0, 1]$, which is the denoised and completed version of the (partially observed) graded leaner response matrix Y. The correctness is computed using the model's prediction (rounded toward 0 or 1 using 0.5 as the threshold) and the actual correctness of the student step in the data.

4.2 Evaluation Metrics

How can we measure what is "good" modeling? All reported results are the average 5-fold cross-validation (or run 5 times). Three measures of quality, Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Area Under the Curve (AUC) Score, are used to evaluate the performance of these models in modeling knowledge tracing for CSAL AutoTutor lessons case data.

$$MAE = \frac{1}{n}\sum \mid y_{pred} - y_{obs} \mid \tag{4.13}$$

$$RMSE = \sqrt{\frac{1}{n}(y_{pred} - y_{obs})^2} \tag{4.14}$$

where the y_{obs} refers to the observations while the y_{pred} represents the predictions by the corresponding model. The lower the value of these MAE and RMSE metrics, the better (the higher accuracy) the corresponding model is. And the AUC is computed by obtaining all y_{pred} values, and then using them to form the Receiver Operator Characteristic (ROC) curve in one orthogonal plane coordinate system (x-axis is true positives rate, and the y-axis is false negative rate). The AUC score refers to the area under that ROC curve. The model with a higher AUC score is better at modeling knowledge tracing.

5 About the Dataset

The prior dataset used for this study is collected from the CSAL AutoTutor lessons [51–53]. We select four lessons in the series of topics on stories and texts, which consist of Lesson 1 "Evaluating Information in Persuasive Text" (for finding the main arguments and support), Lesson 2 "Cause and Effect" (for finding causes and consequences in texts), Lesson 3 "Problems and Solutions" (for identifying problems and how to solve them), and Lesson 4 "Inferences from Texts" (for making inferences in informational texts). Each lesson has between 10 and 30 categorized-responses-based questions for assessing their performance about their literacy skill. The questions in each lesson are classified into three categories according to levels of task difficulties, which are intermediate (M), easy (E), and hard (H). Table 1 shows the simple statistics of this dataset.

Table 1. Statistics of dataset from CSAL AutoTutor lessons

Dataset	#learners	#questions	Max. attempts
Lesson 1	107	8(M)+8(E)+11(H)	9
Lesson 2	118	9(M)+10(E)+10(H)	9
Lesson 3	140	11(M)+8(H)	5
Lesson 4	46	10(M)+9(E)+10(H)	7

* Note: (1) Throughout the table, the number of questions is not evenly distributed among the different levels of task difficulties. For example, some easy-level questions in some lessons are rarely called in real experiments, e.g. only 2 learners in Lesson 1, and 0 learners in Lesson 3. (2) The number of learners differs in different levels of task difficulties for each lesson. The learners that finished the intermediate difficulty level of questions can be split into two parts (upgraded hard level and downgraded easy level) based on their performance at the intermediate level.

6 Results and Discussion

The results of evaluating four types of knowledge tracing models (BKT, PFA, SPARFA-Lite, and TF-based method) are shown in Table 2. As mentioned above, the M, E, and H refer to the intermediate, easy, and hard level of task difficulties of questions separately. And both PFA and BKT are evaluated under single-KC-fits-all mode (marked as "Single KC") and one-KC-one-question mode (marked as "Unique KC").

In order to get the best fit, the tuning parameters of these models need to be adjusted in optimization. For the current BKT used in our study, the initial values for the initial learning rate $P(L_0)$, slip parameter $P(S)$, and guess parameter $P(G)$ are randomly selected within the range of 0.05 to 0.95. The setting of these initial parameters' values in BKT are individualized for individual questions or skill items, and then they are used for the implementation of individualized knowledge tracing for each learner. During the BKT process, the values of the parameters are allowed to be adjusted by EM. This allows the model to converge towards the optimal values of these parameters and more accurately estimate the learning performance of the learner. The values of the tuning parameters obtained through BKT for different KCs are likely to be different depending on the specific KC. Some example BKT tuples that correspond to the KC items (or questions) from the intermediate level of Lesson 1 under the "Unique KC" mode are given in Eq. 6.1. Different values of all adjusted parameters $(P(L_0), (P(T), P(G), P(S))$ for different KCs $(KC1, KC2, KC3, \cdots, KC8)$ demonstrate unique characteristics or individual differences of each KC and the individualization for learner when learning each KC item.

$$BKT\ Tuples \begin{cases} [0.95, 0.45, 0.45, 0.10] \Rightarrow [P(L_0), (P(T), P(G), P(S)]_{KC1} \\ [0.95, 0.95, 0.45, 0.05] \Rightarrow [P(L_0), (P(T), P(G), P(S)]_{KC2} \\ [0.75, 0.80, 0.05, 0.45] \Rightarrow [P(L_0), (P(T), P(G), P(S)]_{KC3} \\ \vdots \\ [0.05, 0.05, 0.45, 0.45] \Rightarrow [P(L_0), (P(T), P(G), P(S)]_{KC8} \end{cases} \tag{6.1}$$

The estimation of PFA using the generalized linear mixed model with individual learners as random effects for each KC (under "Unique KC" mode) emphasizes the consideration of individual differences among different learners and their related skills for acquiring KCs in learning process [15,40]. Different estimates of intercepts and coefficients in a CSAL AutoTutor lesson indicate that it is possible to implement different learner models for the same lesson.

The function of averaging the overall performance across multiple attempts was used for getting the original matrix Z in the SPARFA-Lite. However, this definitely can mask individual variations in learning progress, and the factors that influence it, such as memory, time, and practice [10,54].

The RBTF type of TF method was applied in this study. The tuning parameters of RBTF can be different for different CSAL AutoTutor lessons, which mainly include λ, λ_1, λ_2, ω, the number of KCs C, and learning rate for this model lr.

As we can see from Table 2, the TF-based method beats the other three approaches based on the three metrics obtained by these four modeling, overall (although there may be some instances where the performance of RBTF is slightly worse than that of some other methods). It presents that across all these ten lessons, both the MAE values and RMSE values for seven lessons are lower than those of the other three models, and all AUC score values are larger than those of the other three models. It seems that for the instances where the performance is slightly worse, the values of MAE or RMSE are still close to the lowest value reported among the models. For example, the RMSE value of Lesson 1 (H) is 0.4019, which is only slightly higher than the RMSE value of 0.4011 for the BKT (Unique KC). And another example is the MAE value of Lesson 2 (E), which is 0.2758. This value is close to the lowest value of 0.2503 for the SPARFA-Lite model. The difference between these two values is relatively small. Therefore, our overall results suggested TF model performs better than the other three models.

Table 2. Comparisons of different knowledge tracing models on the CSAL AutoTutor dataset

Dataset	Metrics	BKT (Single KC)	BKT (Unique KC)	PFA (Single KC)	PFA (Unique KC)	SPARFA-Lite	TF-based method
Lesson 1 (M)	MAE	0.4623	0.3777	0.4500	0.4071	0.3954	**0.3706**
	RMSE	0.4790	0.4331	0.4733	0.4550	0.6287	**0.4301**
	AUC	0.5632	0.7574	0.5809	0.6703	0.5010	**0.7712**
Lesson 1 (H)	MAE	0.3535	0.3347	0.3340	0.3350	**0.2387**	0.2763
	RMSE	0.4178	**0.4011**	0.4315	0.4152	0.4850	0.4019
	AUC	0.5320	0.7067	0.5363	0.5929	0.4998	**0.7425**
Lesson 2 (M)	MAE	0.4252	0.3257	0.4150	0.3806	0.3245	**0.2982**
	RMSE	0.4575	**0.3957**	0.4557	0.4390	0.5691	0.3964
	AUC	0.5526	0.8169	0.5534	0.6744	0.5146	**0.8223**
Lesson 2 (H)	MAE	0.4184	0.3141	0.4003	0.3595	0.2979	**0.2871**
	RMSE	0.4512	**0.3861**	0.4490	0.4358	0.5449	0.3907
	AUC	0.5465	0.7986	0.5639	0.6584	0.5101	**0.8111**
Lesson 2 (E)	MAE	0.3731	0.3461	0.3666	0.3507	**0.2503**	0.2758
	RMSE	0.4261	0.4118	0.4415	0.4389	0.4969	**0.4072**
	AUC	0.5797	0.7166	0.4606	0.5269	0.5014	**0.7759**
Lesson 3 (M)	MAE	0.3330	0.3175	0.3293	0.3256	**0.2158**	0.2958
	RMSE	0.4077	0.3936	0.4227	0.4098	0.4644	**0.3884**
	AUC	0.4912	0.6970	0.5158	0.5308	0.4993	**0.7133**
Lesson 3 (H)	MAE	0.4588	0.3275	0.4446	0.4380	0.3400	**0.3015**
	RMSE	0.4768	0.4004	0.4782	0.4704	0.5826	**0.3950**
	AUC	0.4850	0.8276	0.5423	0.5432	0.5059	**0.8362**
Lesson 4 (M)	MAE	0.4598	0.4106	0.4437	0.4487	0.3625	**0.3424**
	RMSE	0.4766	0.4583	0.4775	0.4799	0.6008	**0.4544**
	AUC	0.5584	0.6893	0.5401	0.5308	0.5014	**0.7371**
Lesson 4 (H)	MAE	0.4945	0.3998	0.4914	0.4815	0.4287	**0.3838**
	RMSE	0.4968	0.4605	0.5154	0.5027	0.6544	**0.4567**
	AUC	0.4772	0.7295	0.5823	0.5312	0.5063	**0.7461**
Lesson 4 (E)	MAE	0.4819	0.4304	0.4789	0.4806	0.3583	**0.3490**
	RMSE	0.4929	0.4750	0.4989	0.4939	0.5789	**0.4674**
	AUC	0.5429	0.6545	0.4687	0.5000	0.5455	**0.8368**

* Because the easy-level questions in Lesson 1 and Lesson 3 are rarely called, their corresponding modeling results are missing in this table. The metric values may exhibit slight variations depending on the tuning parameters.

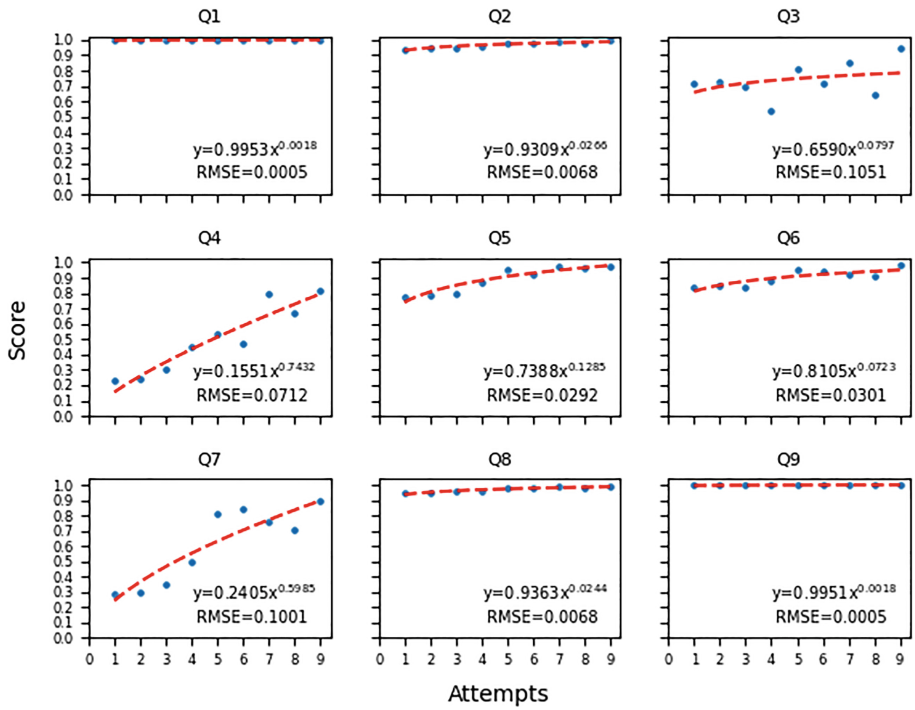

Fig. 2. The distribution of learner's performance for one learner in Lesson 2 (M). Note that the data is modeled by power-law functions and the RMSE is used as goodness-of-fit measure for the regression.

The tensor framework ($\hat{T}$) computed from TF can be regarded as the state space representation described in Session 3. In this sense, the ($\hat{T}$) provides intensive representation of the knowledge learning information that captures underlying structure and relationships between its elements, which overcomes the sparsity of original tensor by tensor factorization mathematically. This lays a foundation for helping us analyze, interpret and store the information more efficiently, more interpretable and easier.

Next, we investigated the individual differences derived from the tensor framework obtained from the TF method. The Lesson 2 (M) is taken as the example here.

Figure 2 shows the distribution of learner's performance for one learner from Lesson 2 (M). We make the following observations: 1) the score converges to 1 driven by attempts (a score of 1 represents the wholly master state), and 2) all distributions were well modeled by the same form of the power-law function:

$$Y = aX^b \tag{6.2}$$

where the Y represents the performance (probability for gaining correct answers), and X is the number of opportunities to practice a skill or attempts. a and b are the regression coefficients. In our research, a can represent a measurement of the student's starting level or ability in a given KC-related skill or knowledge. And b could be used to represent the learning rate of the learner, specifically in relation to knowledge acquired through practice. This finding confirms the well-known "power law learning curve" in educational or training contexts [55,56], which opens up another opportunity for further investigation when combined with the TF-based method.

In the CSAL AutoTutor lesson that includes the series of questions $[Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8, Q_9]$, the fluctuations of these two parameters can represent differences between learners in their learning of different skills (mainly in aspects of their learning styles, abilities, and progress). By exploring the boundaries of the model parameters, we can obtain the ranges of the values that the parameters can take.

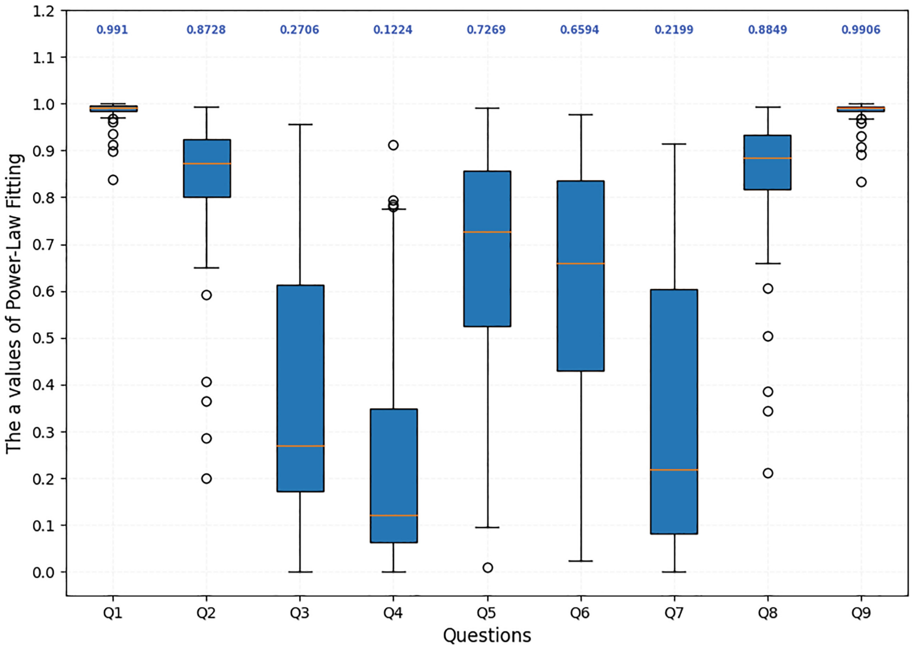

Fig. 3. The distribution of values of a parameter for all students in Lesson 2 (M)

Figure 3 shows the distribution of values of the a parameter. The interquartile range for the Q1 and Q9 are relatively shorter, which indicates that the initial learning features (starting level or ability) of the learners are very similar to each other. On the other hand, for the other questions, the difference is relatively larger, which suggests that there is more diversity in the initial abilities of the learners.

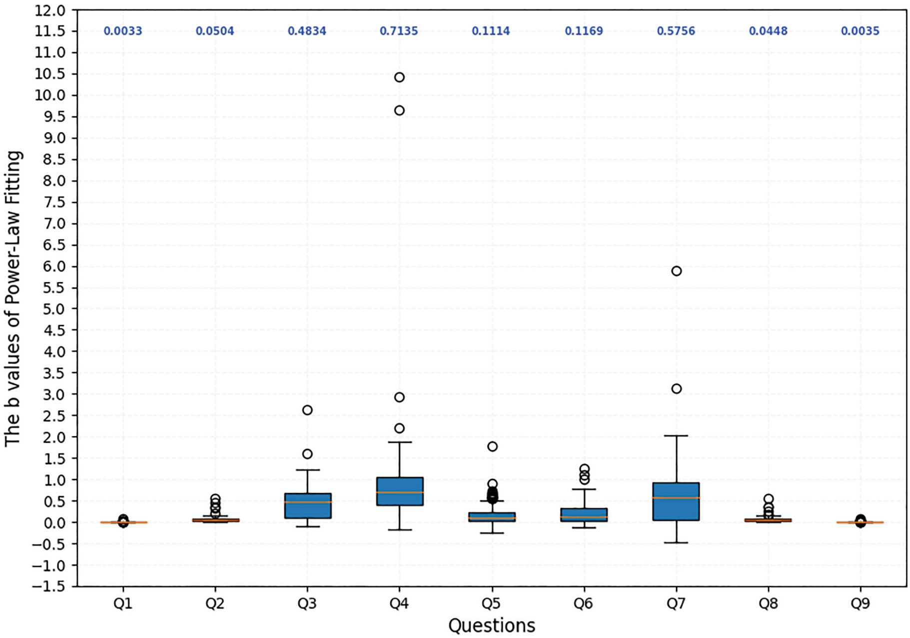

Fig. 4. The distribution of values of b parameter for all students in Lesson 2 (M)

Figure 4 shows the distribution of values of the b parameter. Similar observations are made for the b parameter. The relatively short interquartile range for the Q1 and Q9 suggests that the learning rate is very close for all learners, while the greater differences in learning other questions.

The two parameters a and b actually quantify the uncertainties generated by the initial states, evolving knowledge states, and boundary or extreme conditions that impact the learning process. The differences in the variations of two parameters a and b across different dimensions in the multidimensional framework, can be indicative of the individual learning patterns among learners and their learning of contents. Further research is still needed to understand the influence of these parameters on learning outcomes quantitatively.

7 Conclusion

To summarize, based on the results of the analysis, it appears that the tensor-based framework (TF-based) generates the best-fitting model for modeling the learning process (knowledge-level) for AutoTutor CSAL lessons. The findings suggest that the TF-based method can provide a relatively accurate way of capturing and predicting the learner's performance for real-world implementation of ITSs. This conclusion is based on the comparison of the model's performance measures (e.g., MAE, RMSE, and AUC) with those of other models. Observing that the distribution of parameters used in the model of an ITS fits a power-law function can provide valuable insights into the individual differences among learners, especially when considering multiple dimensions. By capturing the individual differences among learners, the ITS can provide more personalized and effective instruction, leading to improved learning outcomes.

The advantages of TF-based method (RBTF) over BKT, PFA, and SPARFA-Lite are: 1) the capability of this method to handle the multiple dimensionalities of the factors (from learners, attempts, or time, and questions) that critically affect the performance, 2) the scalability in the "elastic" size of these dimensions, 3) its generality in exploring and exploiting the learning model accurately, 4) its practical potential that more accurately capture the individual differences among learners without requiring any changes in the real-world implementation of ITSs.

Also, it is the potential to combine the use of TF-based method, BKT, and PFA (or just two of them) in order to model human learning more accurately in the future. By incorporating multiple methods, it may be possible to capture different aspects of the learning process and account for memory, time, practice, and sequence, as well as interactions between these factors.

This work is still progress, more deeper and extensive results will be presented in other paper.

Acknowledgements. This research was supported by the National Science Foundation Learner Data Institute (NSF #1934745).

References

1. Granić, A., Adams, R.: User sensitive research in e-learning: exploring the role of individual user characteristics. Univ. Access Inf. Soc. **10**(3), 307–318 (2011)
2. O'Shea, T., et al.: Tools for creating intelligent computer tutors. In: Proceedings of the International NATO Symposium on Artificial and Human Intelligence, pp. 181–199 (1984)
3. Graesser, A.C., et al.: Assessment with computer agents that engage in conversational dialogues and trialogues with learners. Comput. Hum. Behav. **76**, 607–616 (2017)
4. Mousavinasab, E., et al.: Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. Interact. Learn. Environ. **29**(1), 142–163 (2021)
5. Woolf, B.P.: Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing e-Learning. Morgan Kaufmann, Burlington (2010)
6. Egan, D.E.: Individual differences in human-computer interaction. In: Handbook of Human-Computer Interaction, pp. 543–568. Elsevier (1988)
7. Kuurstra, J.: Individual differences in human-computer interaction: a review of empirical studies (2015)
8. Landriscina, F.: Simulation and Learning. Springer, Heidelberg (2013)
9. MacLellan, C.J.: Computational models of human learning: applications for tutor development, behavior prediction, and theory testing. Ph.D. thesis. Carnegie Mellon University (2017)
10. Illeris, K: A comprehensive understanding of human learning. In: Contemporary Theories of Learning, pp. 1–14. Routledge (2018)

11. Corbett, A.T., Anderson, J.R.: Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User-adapted Interact. **4**(4), 253–278 (1994)
12. Block, J.H., Burns, R.B.: Mastery learning. Review Res. Educ. **4**, 3–49 (1976)
13. Pardos, Z.A., Heffernan, N.T.: Modeling individualization in a Bayesian networks implementation of knowledge tracing. In: De Bra, P., Kobsa, A., Chin, D. (eds.) UMAP 2010. LNCS, vol. 6075, pp. 255–266. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13470-8_24
14. Trifa, A., Hedhili, A., Chaari, W.L.: Knowledge tracing with an intelligent agent, in an e-learning platform. Educ. Inf. Technol. **24**(1), 711–741 (2019)
15. Pavlik, P.I., Eglington, L.G., Harrell-Williams, L.M.: Logistic knowledge tracing: a constrained framework for learner modeling. IEEE Trans. Learn. Technol. **14**(5), 624–639 (2021)
16. Essa, A.: A possible future for next generation adaptive learning systems. Smart Learn. Environ. **3**(1), 1–24 (2016)
17. Brusilovsky, P.: Adaptive hypermedia for education and training. Adapt. Technol. Training Educ. **46**, 46–68 (2012)
18. Corbett, A.T., Anderson, J.R.: Student modeling and mastery learning in a computer-based programming tutor. In: Frasson, C., Gauthier, G., McCalla, G.I. (eds.) ITS 1992. LNCS, vol. 608, pp. 413–420. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-55606-0_49
19. Abdelrahman, G., Wang, Q., Nunes, B.P.: Knowledge tracing: a survey. ACM Comput. Surv. **55**, 1–37 (2022)
20. Embretson, S.E., Reise, S.P.: Item Response Theory. Psychology Press (2013)
21. Barnes, T.: The Q-matrix method: mining student response data for knowledge. In: American Association for Artificial Intelligence 2005 Educational Data Mining Workshop, Pittsburgh, PA, USA, pp. 1–8. AAAI Press (2005)
22. Cen, H., Koedinger, K., Junker, B.: Comparing two IRT models for conjunctive skills. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 796–798. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_111
23. Pavlik, P.I., Jr., Cen, H., Koedinger, K.R.: Performance factors analysis–a new alternative to knowledge tracing. Online Submission (2009)
24. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized Bayesian knowledge tracing models. In: Lane, H.C., Yacef, K., Mostow, J., Pavlik, P. (eds.) AIED 2013. LNCS (LNAI), vol. 7926, pp. 171–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39112-5_18
25. Käser, T., et al.: Dynamic Bayesian networks for student modeling. IEEE Trans. Learn. Technol. **10**(4), 450–462 (2017)
26. Lan, A.S., et al.: Sparse factor analysis for learning and content analytics. arXiv preprint arXiv:1303.5685 (2013)
27. Sahebi, S., Lin, Y.-R., Brusilovsky, P.: Tensor factorization for student modeling and performance prediction in unstructured domain. In: International Educational Data Mining Society (2016)
28. Sahebi, S., Huang, Y., Brusilovsky, P.: Parameterized exercises in Java programming: using knowledge structure for performance prediction. In: The Second Workshop on AI-Supported Education for Computer Science (AIEDCS), University of Pittsburgh, pp. 61–70 (2014)
29. Doan, T.-N., Sahebi, S.: Rank-based tensor factorization for student performance prediction. In: 12th International Conference on Educational Data Mining (EDM) (2019)

30. Karatzoglou, A., et al.: Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 79–86 (2010)
31. Graesser, A.C., et al.: Reading comprehension lessons in AutoTutor for the Center for the Study of Adult Literacy. In: Adaptive Educational Technologies for Literacy Instruction, pp. 288–293. Routledge (2016)
32. Graesser, A.C., Li, H., Forsyth, C.: Learning by communicating in natural language with conversational agents. Curr. Dir. Psychol. Sci. **23**(5), 374–380 (2014)
33. Fang, Y., Lippert, A., Cai, Z., Hu, X., Graesser, A.C.: A conversation-based intelligent tutoring system benefits adult readers with low literacy skills. In: Sottilare, R.A., Schwarz, J. (eds.) HCII 2019. LNCS, vol. 11597, pp. 604–614. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22341-0_47
34. Graesser, A.C., et al.: Coh-Metrix measures text characteristics at multiple levels of language and discourse. Elementary Sch. J. **115**(2), 210–229 (2014)
35. McNamara, D.S., et al.: Automated Evaluation of Text and Discourse with Coh-Metrix. Cambridge University Press, Cambridge (2014)
36. Newell, A., Simon, H.A., et al.: Human Problem Solving, vol. 104. 9. Prentice-Hall, Englewood Cliffs (1972)
37. Cen, H., Koedinger, K., Junker, B.: Learning factors analysis–a general method for cognitive model evaluation and improvement. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 164–175. Springer, Heidelberg (2006). https://doi.org/10.1007/11774303_17
38. Koedinger, K.R., Corbett, A.T., Perfetti, C.: The knowledge-learning-instruction framework: bridging the science-practice chasm to enhance robust student learning. Cogn. Sci. **36**(5), 757–798 (2012)
39. Aleven, V., Koedinger, K.R.: Knowledge component (KC) approaches to learner modeling. Des. Recommendations Intell. Tutoring Syst. **1**, 165–182 (2013)
40. Yudelson, M., Pavlik, P.I., Koedinger, K.R.: User modeling–a notoriously black art. In: Konstan, J.A., Conejo, R., Marzo, J.L., Oliver, N. (eds.) UMAP 2011. LNCS, vol. 6787, pp. 317–328. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22362-4_27
41. Chi, M., et al.: Instructional factors analysis: a cognitive model for multiple instructional interventions. In: EDM 2011, pp. 61–70 (2011)
42. Eglington, L.G., Pavlik, P.I., Jr.: How to optimize student learning using student models that adapt rapidly to individual differences. Int. J. Artif. Intell. Educ. 1–22 (2022)
43. Baker, R.S.J., Corbett, A.T., Aleven, V.: More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In: Woolf, B.P., Aïmeur, E., Nkambou, R., Lajoie, S. (eds.) ITS 2008. LNCS, vol. 5091, pp. 406–415. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-69132-7_44
44. Baker, R.S.J., Corbett, A.T., Aleven, V.: Improving contextual models of guessing and slipping with a truncated training set (2008)
45. van De Sande, B.: Properties of the Bayesian knowledge tracing model. J. Educ. Data Min. **5**(2), 1–10 (2013)
46. Wang, D., et al.: Traditional knowledge tracing models for clustered students (2021)
47. Lan, A.S., Studer, C., Baraniuk, R.G.: Quantized matrix completion for personalized learning. arXiv preprint arXiv:1412.5968 (2014)
48. Lan, A.S., Studer, C., Baraniuk, R.G.: Matrix recovery from quantized and corrupted measurements. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4973–4977. IEEE (2014)

49. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
50. Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
51. Shi, G., Pavlik, P., Jr., Graesser, A.: Using an additive factor model and performance factor analysis to assess learning gains in a tutoring system to help adults with reading difficulties. In: Grantee Submission (2017)
52. Shi, G., et al.: Diagnostic assessment of adults' reading deficiencies in an intelligent tutoring system. In: ITS Workshops, pp. 105–112 (2018)
53. Shi, G., et al.: Exploring an intelligent tutoring system as a conversation-based assessment tool for reading comprehension. Behaviormetrika **45**(2), 615–633 (2018). https://doi.org/10.1007/s41237-018-0065-9
54. Wickelgren, W.A.: Human learning and memory. Ann. Rev. Psychol. **32**(1), 21–52 (1981)
55. Newell, A., Rosenbloom, P.S.: Mechanisms of skill acquisition and the law of practice. Technical report. Carnegie-Mellon University, Pittsburgh, PA, Department of Computer Science (1980)
56. DeKeyser, R.: Skill acquisition theory. In: Theories in Second Language Acquisition, pp. 83–104. Routledge (2020)

View publication stats