LogCF: Deep Collaborative Filtering with Process Data for Enhanced Learning Outcome Modeling

Fu Chen
University of Alberta
fu4@ualberta.ca

Ying Cui
University of Alberta
yc@ualberta.ca

Effective learning outcome modeling is crucial to the success of learning evaluation in education. In the digital age, the movement towards online learning and computerized assessments has resulted in an explosion of structured and unstructured educational data (e.g., learners' problem-solving process data), which offers new opportunities for large-scale learning outcome modeling. Traditional psychometric models are of limited scalability and cannot adequately model item and learner features with incomplete and unstructured learner performance data. Existing advances in machine learning typically don't account for learners' problem-solving processes for learning outcome modeling. Leveraging the collaborative filtering approach used in recommender systems, we develop a general framework of deep learning-based collaborative filtering with process data for enhanced learning outcome modeling, which is named LogCF. LogCF is capable of modeling learner- and item-skill associations as well as predicting learners' item responses. In our experiments on two datasets of distinctive features, we demonstrate the superior predictive capacity of LogCF compared with other educational data mining and psychometric measurement models under different conditions of training/test partition ratios. In addition, we derive three variants of LogCF to examine whether item-skill associations learned or refined by LogCF are superior to the expert-specified ones. In addition, we also demonstrate the interpretability of learner- and item-skill associations learned by LogCF.

Keywords: learning outcome modeling, process data, log data, collaborative filtering, deep learning, Q-matrix

1. Introduction

The dawning age of big data has witnessed extensive developments and changes in education in the 21st century. Particularly, the rapid evolution in information and communication technologies enables learners all over the world to have affordable access to personally tailored educational experiences. Notably, the area of personalized learning is nourished by the availability of and access to a tremendous amount of data on learning behaviors. By analyzing learning behavior data, learners' learning progress can be continuously monitored and assessed by analyzing their interactions with learning resources, and learners can be provided with timely enhancement and remediation adaptively based on their learning progress and outcomes.

Effective personalized learning systems rely on accurate and timely monitoring and evaluation of learning outcomes. In this regard, many educational systems such as intelligent tutoring systems (Psotka et al., 1988) and computerized adaptive testing (van der Linden and Glas, 2000) are developed based on a series of learning outcome modeling techniques in both domains of educational data mining and educational measurement, e.g., Bayesian knowledge tracing (BKT; Corbett and Anderson 1994), item response theory (IRT; Embretson and Reise 2000), and cognitive diagnosis (Tatsuoka, 2009). Earlier approaches for learning monitoring and evaluation, psychometric measurement models in particular, however, are not advantageous in this era of educational big data because they are not scalable and efficient and also require a large amount of human effort for pre-defined rules. Fortunately, machine learning techniques have demonstrated superior prediction performance and high scalability for learning outcome modeling in large-scale personalized learning settings, according to some pioneering studies (Baker and Yacef, 2009; Bergner et al., 2012; Cheng et al., 2019; Lan et al., 2014). These machine learning-based developments or applications mainly serve two purposes: estimating learners' mastery level of latent skills, such as learning outcome modeling, and organizing learning content (e.g., learning materials, assessment tasks, questions), such as skill modeling, which can be conceptualized as learning analytics and content analytics respectively (Lan et al., 2014).

Particularly, in recent years, both communities of educational data mining and psychometric measurement have devoted efforts to data-driven skill modeling, or knowledge component modeling, because of its pivotal role in effective learning evaluation. For example, in psychometrics, successful applications of cognitive diagnostic models often require an accurate specification on item-skill associations, i.e., the Q-matrix, and a misspecified Q-matrix would lead to poor model-data fit (Hansen et al., 2016; Liu et al., 2016), which in turn undermines model classification accuracy. As such, some efforts have been made to refine the Q-matrix for improved learner ability estimation in the community of cognitive diagnosis (Liu et al., 2012; Chiu, 2013). Despite that, skill modeling by domain experts might be infeasible for large-scale assessments given its high cost of time and money. Fortunately, a variety of methods have been developed from the community of educational data mining to automatically extract or refine the mapping of assessment items to latent skills (Chaplot et al., 2018; Desmarais, 2012; Desmarais and Naceur, 2013; Lindsey et al., 2014; Sun et al., 2014).

Collaborative filtering (CF), an approach widely used in recommender systems (Linden et al., 2003; Sarwar et al., 2001) is good fit to machine learning-based learning analytics and content analytics. CF aims to recommend new items to users based on ratings or clicks of the new items as well as users' prior interactions with the other items (Su and Khoshgoftaar, 2009). More specifically, model-based CF utilizes partial interaction information to estimate a set of item- and user-related parameters, which are then used to make probabilistic predictions on unknown interactions. Like recommending new items to users in recommender systems, in education, predicting learners' responses on unassigned items helps in determining the best-fitting assessment tasks and associated learning materials. For example, it is more beneficial for learners if they can be recommended those learning materials and assessment problems within the zone of their proximal development. Problems that are either not likely or very likely to be correctly solved are of limited benefits for learners. In this sense, the CF approaches should be able to predict learners' future item responses. Among various model-based CF algorithms, matrix factorization is widely used and popularized by the Netflix Prize. The basic idea of matrix factorization is that a user-item interaction matrix with missing information can be decomposed into the product of two or more lower dimensionality matrices, which represent user-factor and item-factor associations. In this sense, CF, especially matrix factorization, aligns with the purpose of learning analytics and content analytics. Using matrix factorization with educational assessment data enables modeling learner- and item-skill associations, which represent the degree to which learners master the latent skills and the degree to which items measure latent skills. However, despite the interpretability of lower dimensionality matrices with latent skills, the linear combination, represented by the inner-products, of learner-skill and item-skill associations in matrix factorization is still far from comprehensive for explaining the complexity of learner-item interactions. As such, some recent advances propose to incorporate deep learning into CF for capturing a higher degree of complexity of learner-item interactions by improving the model intricacy or leveraging auxiliary information (Elkahky et al., 2015; He et al., 2017; van den Oord et al., 2013; Wang et al., 2015; Zhang et al., 2016).

Recently, in addition to learners' explicit performance data, using process data, or log file data, for profiling and facilitating learning is an emerging area in both educational data mining and psychometric measurement. Process data is a valuable source of information revealing more details on the learning process through learners' interactions with evaluation tasks. Unfortunately, conventional educational data mining techniques (e.g., BKT) and psychometric measurement models (e.g., IRT and cognitive diagnostic models) mostly exploit learning performance data and cannot deal with process data explicitly. In the literature, learners' process data has been used to identify learners' strategies for solving problems (Greiff et al., 2015), evaluate learners' latent skills (Liu et al., 2018), and predict learners' learning outcomes (Chen et al., 2019). These studies have shed light on the potential of process data for modeling and interpreting learners' learning outcomes.

Given the aforementioned disadvantages of conventional methods for learning outcome and skill modeling, the potential of learner process data, and the advances in deep learning-based CF, this paper proposes a general framework of deep learning-based collaborative filtering with process data (LogCF) to enhance learning outcome modeling. Specifically, the inputs for our framework include learner and item identifications (IDs) and learners' problem-solving process data (i.e., raw actions and time sequences for solving each assessment task), which are then fed into two deep learning architectures. In the deep learning architecture for learner and item IDs, learner and item IDs are embedded as learner and item latent factors, which are then concatenated and fed into multiple neural network layers for learning non-linear associations between items and learners (i.e., the collaborative filtering module of our framework). In the deep learning architecture for the process data, the action embeddings and the time sequences are fed into multiple long short-term memory (LSTM; Hochreiter and Schmidhuber 1997) layers for learning the temporal dependencies between actions and time durations in solving assessment tasks. The LSTM outputs of actions and time are then concatenated and fed into multiple neural network layers for learning the non-linear association between actions and time durations. Finally, the final collaborative filtering output and the process data output are concatenated and fed into a neural network layer with sigmoid activation to learn the probability of successfully solving a problem.

LogCF can be used in different scenarios. For example, in intelligent tutoring systems, learners' learning outcome data are very sparse given that learners are assigned with different sets of questions, and questions are responded to by different numbers of learners. However, in educational assessments, it is often the case that a group of learners is evaluated by the same set of test or questionnaire items. We believe that LogCF might benefit both areas. Therefore, in our study, the predictive capacity of LogCF is validated by experiments with one dataset in educational data mining and one dataset in educational assessment, which are two distinct perspectives of model performance evaluation.

In addition to the prediction performance of LogCF, we also evaluate the capacity of LogCF to retrieve from scratch or refine the expert-specified item-skill associations (i.e., the Q-matrix). Specifically, we derive three variants of LogCF, which impose different constraints on item-skill associations and compare their prediction performance under the conditions of different latent skill dimensions and training/test partition ratios. In the three variants of LogCF, item-skill associations are fixed as or initialized by the weights defined by experts, or randomly initialized, or fined tuned by a two-stage process. The comparison between LogCF variants reveals whether LogCF is capable of learning or refining a mapping of items to skills better than the expert-specified one.

This work contributes to the literature in the following ways:

1. We utilize a deep learning architecture to model learner and item representations along with learners' problem-solving actions and time durations and develop a generic framework for making probabilistic predictions based on deep CF and deep neural networks. We show how to incorporate process data into learning outcome modeling in a general way, which reveals the potential of process data in predicting learning outcomes.

2. We show the superior prediction performance of LogCF in comparison with other educational data mining or psychometric measurement approaches by extensive experiments with a dataset from the community of educational data mining and an international large-scale assessment dataset used widely by the community of psychometric measurement.

3. We demonstrate the capacity of LogCF to learn from scratch or refine the expert-specified item-skill associations and how they are affected by latent skill dimensions.

4. We demonstrate the interpretability of LogCF in explaining the degree to which learners understand and items measure the latent skill, in comparison with the parameters estimated by psychometric measurement models.

2. Related work

2.1. CF for skill and learning outcome modeling

Extensive work in educational data mining has been conducted to learn or refine item-skill associations from learner data. The item-skill association is a parallel concept to the concept of a Q-matrix in psychometrics, which is a binary matrix depicting the mapping of assessment items to targeted skills (i.e., which items measure which skills). The weights of item-skill associations can also be real-valued, which indicate the strength of each item measuring each latent skill. Despite the fact that CF developments and applications specific to education are scarce, a variety of educational data mining techniques have been developed based on the idea of CF, especially matrix factorization, to estimate or refine Q-matrices from learner data. For example, CF has been used to evaluate an expert-generated Q-matrix (Durand et al., 2015) and to generate a data-driven Q-matrix (Desmarais, 2012; Desmarais and Naceur, 2013; Sun et al., 2014). Findings suggested that the data-driven Q-matrix by CF could be successfully used to model learning outcomes and the CF approach also resulted in better prediction performance than experts' knowledge (Desmarais and Naceur, 2013; Matsuda et al., 2015). The idea of matrix factorization was also used to learn item-skill associations from scratch. For example, in the sparse factor analysis algorithm proposed by Lan et al. (2014), item- and learner-skill associations and item difficulties can be directly learned from learners' binary-valued item response data without any auxiliary information. Matrix factorization approaches can also be used for learning outcome modeling by incorporating contextual information. For example, the time when a learner was graded was found to be helpful for improving the prediction performance (Almutairi et al., 2017). In addition to matrix factorization, tensor factorization approaches have been used for learning outcome modeling by modeling learners' attempt sequences with selected course quizzes as feedback (Sahebi et al., 2016) or by accounting for temporal effects (Thai-Nghe et al., 2012). However, it should be noted that matrix factorization-related approaches (e.g., alternating least square, non-negative matrix factorization, and Boolean matrix factorization) were found to have limited capacities to retrieve expert-specified Q-matrices from scratch (Desmarais, 2011; Desmarais and Naceur, 2013), and they have received more interest by the community of educational data mining for their capacities to refine Q-matrices.

Furthermore, a wide range of approaches irrelevant to CF has also been effective for skill and learning outcome modeling. For example, Lindsey et al. (2014) proposed a generative probabilistic model based on BKT with experts' knowledge as a prior to discover the mapping of items to latent skills and found that expert-specified item-skill associations were of limited value for their approach. In addition, learning outcome modeling can be enhanced by integrating learners' log data with multi-modal data streams (e.g., audio and video; Liu et al. 2016; Liu et al. 2019), and unifying multiple educational data mining methods (e.g., knowledge tracing machines; Vie and Kashima 2019).

2.2. Deep learning for skill and learning outcome modeling

A variety of deep learning-based approaches have been developed for skill and learning outcome modeling in the literature. Deep knowledge tracing (DKT; Piech et al. 2015) is a representative approach based on recurrent neural networks, which model the temporal dependencies between learners' history item responses to predict their future item responses. DKT can also be used for open-ended item responses such as programming exercises (Wang et al., 2017). The interpretability of deep learning-based knowledge tracing models can be improved by integrating with psychometric models such as IRT (Yeung, 2019).

Deep learning-based approaches can also be used for skill modeling. For example, the dynamic and deep variant of the additive factors model (Pardos and Dadu, 2018) was developed to refine or learn from scratch the expert-specified item-skill associations. Moreover, some recent advances have focused on how to derive Q-matrices without learner performance data. For example, in the cognitive representation learner proposed by Chaplot et al. (2018), the Q-matrix for each item can be automatically extracted from the item content through representation learning based on convolutional or recurrent neural networks. Their framework does not require any learner performance data and works well for items in ill-structured domains. Furthermore, a variety of information in addition to performance data can be incorporated by deep learning for learning outcome modeling. For example, in the deep learning-enhanced item response theory framework proposed by Cheng et al. (2019), item difficulties, item discriminations, and learner abilities are extracted from item texts and item-associated latent skills through a deep learning module to predict learners' item responses. Despite relying on an IRT framework for prediction, their approach largely outperforms IRT models given that item text information is exploited for training. In addition, deep learning approaches such as recurrent neural networks were also used to detect learners' affect from their interactions with learning systems for learning outcome modeling (Botelho et al., 2017).

Despite not being designed specifically for educational problems, some deep learning-based CF approaches have also been very promising in learning outcome modeling. These approaches mainly incorporate multiple neural network layers in CF for learning more complex learner and item representations or non-linear associations between learners and items for prediction. For example, Nguyen et al. (2018) designed a neural network architecture for matrix completion by stacking neural network layers onto the item and learner vectors to learn item and learner representations, Li et al. (2015) devised a deep learning CF framework for learning more effective item and user representations via item and user side information by combining probabilistic matrix factorization with marginalized denoising stacked auto-encoders, and He et al. (2017) developed a neural CF framework without side information by leveraging a multi-layer perceptron to learner the non-linearities of user-item interaction along with a generalized matrix factorization. These deep learning-based CF approaches generally showed higher prediction accuracy on missing information. However, it has been found that some deep learning-based advances in this area might not be capable of outperforming well-tuned conventional baselines (Dacrema et al., 2019).

2.3. Psychometric measurement models

Apart from machine learning approaches, learning outcome modeling is also a long-standing topic in the community of educational measurement or psychometrics. Most psychometric measurement models are latent variable models. Nowadays, the most popular psychometric model families are IRT models (Embretson and Reise, 2000) and cognitive diagnosis models (Rupp et al., 2010). IRT estimates learners' probabilities of correct responses by modeling a set of item and learner parameters. Generally, item parameters involve item difficulty, item discrimination, and item guessing, and learner parameters indicate learners' latent ability levels. IRT models have different assumptions on item parameters. For example, the three parameter logistic (3PL) model assumes that item difficulty, item discrimination, and item guessing work as a whole to affect the probabilities of correct responses, the two parameter logistic (2PL) model adopts both item difficulty and item discrimination, and the Rasch model (Rasch, 1980) only considers item difficulty for modeling with a fixed item discrimination. In IRT models, the item-learner interaction is modeled by the linear combination of item difficulty, item discrimination, and learner ability, which is in turn converted to a predicted probability of a correct response ranging from 0 to 1 by a sigmoid transformation. IRT models assume unidimensionality, which means only one latent skill can be measured by the assessment items. More recently, the multidimensional item response modeling (MIRT) approach was developed to deal with multidimensional data (Yao and Boughton, 2007), which allows different item discriminations and learner abilities for multiple latent skills. Similar to MIRT, cognitive diagnosis models also account for multiple latent skills. Unlike MIRT, cognitive diagnosis requires a prespecified Q-matrix to map each item to each skill and estimates a skill mastery profile for each learner. The major challenge for effective cognitive diagnosis is how to accurately prespecify the Q-matrix, which is typically decided by domain experts.

In recent years, given the importance of process data for learning outcome modeling, psychometric measurement models have been also adapted to model learners' problem-solving processes. For example, in their study analyzing a complex problem-solving item of an international large-scale assessment dataset, Liu et al. (2018) developed a modified multilevel mixture IRT model to uncover learners' problem-solving strategies and estimate learners' abilities at both the process and item levels. Similarly, Shu et al. (2017) developed a Markov-IRT model for analyzing learners' problem-solving processes to derive features for psychometric measurement. Their studies demonstrate the potential of conventional psychometric measurement models for analyzing process data. However, the capacity of psychometric models to model process data is still under-investigated, especially for the case of multiple items and multiple latent skills.

3. Preliminary

Prior to the introduction of our approach, we first review the basic ideas of matrix factorization. Matrix factorization is an exceptionally effective model-based CF approach. It allows the discovery of how learners and items are associated with the latent factors by factorizing a complete or incomplete learner-item interaction matrix into two or more lower dimensionality matrices of learner and item features. Given an item response matrix $R \in \mathbb{R}^{m \times n}$ consisting of item responses of m learners to n items, matrix factorization decomposes R into two low-rank matrices $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$, which represent the learner-skill and the item-skill association matrices of rank k respectively:

$$R \approx UV^T. \tag{1}$$

The ith row $\overline{u_i}$ of U represents a learner factor of k entries, which stands for the association of user i with the k latent skills. Likewise, the jth row $\overline{v_j}$ of V represents an item factor of k entries, which stands for the association of item j with the k latent skills. As such, each matrix R entry r_{ij} can be approximately modelled as a dot product of the ith learner factor and the jth item factor:

$$r_{ij} \approx \overline{u_i} \cdot \overline{v_j}. \tag{2}$$

Like other machine learning algorithms, regularization can be used to discourage larger latent factor values to avoid overfitting. For a matrix factorization problem with L_2 regularization on both U and V, the algorithm learns the latent learner and item factors U and V by minimizing the following objective function:

$$\arg\min_{U,V} J = \frac{1}{2} \left\| R - UV^T \right\|^2 + \frac{\lambda}{2} \left\| U \right\|_{\mathrm{F}}^2 + \frac{\lambda}{2} \left\| V \right\|_{\mathrm{F}}^2, \tag{3}$$

where λ indicates the regularization weight and $\|\cdot\|_{\mathrm{F}}^2$ indicates the Frobenius norm. Regular optimization methods such as stochastic gradient descent can be used to solve the objective function.

Specific to different application contexts, many constraints can be applied to matrix factorization to improve prediction performance or enhance interpretability such as non-negativity, orthogonality, and sparseness (Ding et al., 2006; Hoyer, 2004; Lee and Seung, 2001). For example, L_1 regularization can be used to encourage sparse latent factor matrices.

4. OUR APPROACH: DEEP CF WITH PROCESS DATA

As we mentioned earlier, matrix factorization is an effective approach for learning item and learner representations for the purposes of learning and content analytics. In other words, we hope to identify the links between each learner and each item to each latent skill. However, conventional matrix factorization approaches cannot adequately capture the complexity of learner-item interactions given that a learning outcome (e.g., item responses) might be influenced by many individual and contextual factors. Recently, deep learning-based CF has shown to be very promising in modeling the user-item interaction by incorporating more complex learning structures. However, none of these approaches accounts for the information of how learners solve assessment tasks. Inspired by a few pioneering studies in education (Greiff et al., 2015; Liu et al., 2018; Chen et al., 2019), we consider that the complexity of learner-item interactions could be reflected by learners' actions and time durations in solving assessment tasks. Our approach attempts to provide a general framework integrating deep learning-based CF with process data learning for enhanced prediction accuracy of responses to a future task along with the interpretability of latent factor matrices for learners and items. In the following sections, we first formularize our problem, then present the general framework of our approach, followed by the introduction to deep CF and deep learning of process data.

4.1. PROBLEM FORMULATION

Suppose there are m learners, n items, and a total of k latent skills measured by the assessment items. The item response matrix can be represented by $\mathbf{R} = \{R_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$, where $R_{ij} = \langle \mathbf{m}_i, \mathbf{n}_j, r_{ij} \rangle$ indicates that learner $\mathbf{m}_i$ has a score of r_{ij} on item $\mathbf{n}_j$. In addition, each item response has an associated problem solving process, and the process data matrix can be represented by $\mathbf{L} = \{L_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$, where $L_{ij} = \langle \mathbf{m}_i, \mathbf{n}_j, l_{ij} \rangle$ indicates that learner $\mathbf{m}_i$ has a problem solving process of l_{ij} on item $\mathbf{n}_j$. It should be noted that $\mathbf{m}_i$ and $\mathbf{n}_j$ can be used to represent various learner and item features. Given that our framework is based on a CF approach, $\mathbf{m}_i$ and $\mathbf{n}_j$ are used to represent learner and item IDs. Given the item response records $\mathbf{R}$ of learners on items , and their problem solving process records $\mathbf{L}$, our goal is to construct a model $\mathcal{M}$ to identify the learner-skill associations $\mathbf{U} = [\mathbf{u}_1, ..., \mathbf{u}_m]$ and the item-skill associations $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_n]$, which are in turn used to predict learners' responses on items. Values of $\mathbf{U}$ and $\mathbf{V}$ can be naturally interpreted as learners' understanding of the k latent skills with larger values indicating higher skill mastery proficiency, and values of $\mathbf{V}$ can be naturally interpreted as the connections of items to the latent skills with larger values indicating stronger affinity. It should be noted that there are no ground truths for $\mathbf{U}$ and $\mathbf{V}$, so the learned $\mathbf{U}$ and $\mathbf{V}$ can be evaluated by the prediction accuracy of model $\mathcal{M}$ in predicting learners' item responses. As we mentioned earlier, in many educational application contexts such as personalized learning, not all of the learning or assessment tasks in the item pool are assigned to each learner, which therefore leads to many missing values in the response matrix $\mathbf{R}$ and the process data matrix $\mathbf{L}$. This is a major challenge for most psychometric measurement models. However, our framework is able to handle the case of missing values.

Given the problem definition, we use the following model for the binary-valued item response $R_{ij} \in \{0,1\}$ of learner $\mathbf{m}_i$ on item $\mathbf{n}_j$, with 1 indicating a correct response and 0 an incorrect response:

$$\begin{aligned} Z_{ij} &= \mathbf{h}^T \begin{bmatrix} \phi^{\text{CF}} \\ \phi^{\text{Log}} \end{bmatrix} \\ R_{ij} &\sim Ber(\sigma(Z_{ij})). \end{aligned} \tag{4}$$

In the model, $Ber(z)$ indicates that learners' correct responses follow a Bernoulli distribution with success probability z, and $\sigma(z)$ denotes a sigmoid function which transforms a real value z to a success probability as follows:

$$\sigma(z) = \frac{1}{1+e^{-z}}. \tag{5}$$

As such, the probability of learner $\mathbf{m}_i$ correctly answering item $\mathbf{n}_j$ $\sigma(Z_{ij}) \in [0,1]$. Z_{ij} indicates the output of a neural network layer with the learned representations of learner-item interactions by deep CF, ϕ^{CF}, and the learned representations of the problem-solving process, ϕ^{Log}, as the inputs. $\mathbf{h}$ denotes the edge weights for outputting Z_{ij} in the neural network layer. In our framework (see Figure 1), ϕ^{CF} is learned by multiple neural network layers with the concatenation of $\mathbf{U}$ and $\mathbf{V}$ as the input, and $\mathbf{U}$ and $\mathbf{V}$ are represented by the learner and item embeddings of learner and item IDs; ϕ^{Log} is learned by multiple neural network layers with the concatenation of action and time representations as the input, and the action and time representations are learned by multiple LSTM network layers with the raw action and time sequences as the inputs. More elaboration on how to learn ϕ^{CF} and ϕ^{Log} by our framework is given in the following sections.

Given the above definition, the likelihood of the observed response matrix R_{ij} can be represented as

$$p(R_{ij}|\mathbf{u}_i, \mathbf{v}_j) = \sigma(Z_{ij})^{R_{ij}}(1-\sigma(Z_{ij}))^{1-R_{ij}} \tag{6}$$

and the optimization problem can be formularized as

$$\underset{\mathbf{U},\mathbf{V}}{\text{maximize}} \sum_{i,j} \log p(R_{ij}|\mathbf{u}_i, \mathbf{v}_j). \tag{7}$$

4.2. General Framework

The general framework of our approach is presented in Figure 1, which is a fusion of deep CF and deep learning of process data. In equation (4), ϕ^{CF} and ϕ^{Log} are learned by deep CF (on the left of the framework) and deep learning of process data (on the right of the framework) respectively.

4.2.1. Deep CF

A multi-layer neural network architecture adopted from the neural CF (He et al., 2017) is built for learning learner-item interactions. Despite the item response matrix shown in the framework, the raw inputs of the deep CF architecture are actually learner and item IDs, which represent learner and item features. Given the categorical nature of learner and item IDs, they are converted to sparse binary vectors by one-hot encoding for learning. For example, for a total of

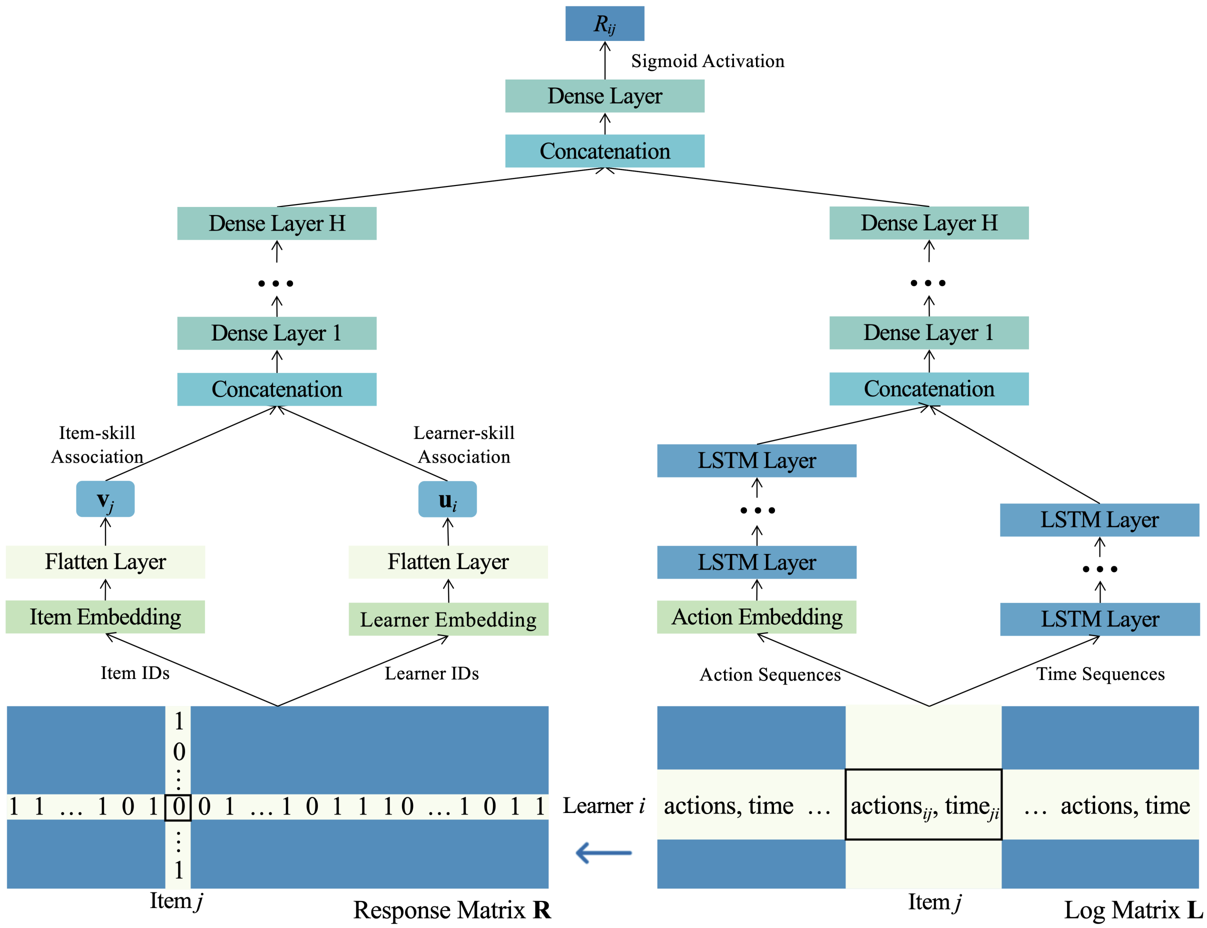

Figure 1: General framework of deep CF with process data.

100 unique learners in the data, each learner is represented as a 100-dimensional vector with a value of 1 for one dimension and all 0s for the other dimensions, which is very sparse for learning, As such, to better represent learners and items, we use the embedding technique to convert sparse vectors of items and learners to dense vectors with a dimension that is the number of latent skills. For instance, suppose there are three latent skills measured by the items, then each item can be represented as a three-dimensional vector with each dimension indicating one latent skill. Likewise, each learner can also be represented as a three-dimensional vector rather than a 100-dimensional vector. Therefore, to learn the k latent factors of learners and items, an embedding layer is then stacked on the input layer of learner and item IDs to project them onto a dense vector of k dimensions. The learner embedding and the item embedding, $\mathbf{u}_i$ and $\mathbf{v}_j$, represent the learner-skill association and the item-skill association respectively. On top of the item and learner embedding layers, a flattening layer is used to reshape the outputs of the embedding layers as the inputs for a following multi-layer neural architecture. The multiple neural network layers are used to discover the complexity of learner-item interactions within the CF framework. In contrast to conventional matrix factorization approaches representing the learner-item interaction as the product of $\mathbf{u}_i$ and $\mathbf{v}_j$, the neural network is able to learn non-linear interactions from the concatenation of learner and item factors. More specifically, the concatenation of learner and item factors is simply to bind the learner and item factors together. For example, given a

three-dimensional item embedding and a three-dimensional learner embedding, the concatenation of them leads to a six-dimensional factor. The output of the last neural network layer ϕ^{CF} is the representation of learner-item interactions learned by the deep learning structure. The above deep CF learning process can be formulated as

$$\phi_{ij}^{\text{CF}} = f_H(\mathbf{W}_H^T f_{H-1}(...f_2(\mathbf{W}_2^T f_1(\mathbf{W}_1^T \begin{bmatrix} \mathbf{U}^T\mathbf{m}_i \\ \mathbf{V}^T\mathbf{n}_j \end{bmatrix}))...)). \tag{8}$$

In the formula, $\mathbf{U}$ and $\mathbf{V}$ denote the latent factor matrices (i.e., learner- and item-skill associations), $\mathbf{W}_1$ to $\mathbf{W}_H$ denote the edge weights for the H neural network layers, f_1 to f_H denote the activation function for each layer, and Θ_f denotes the model parameters of the architecture. It should be noted that in the deep CF architecture, the number of neural network layers and the number of nodes in each layer are hyper-parameters to be tuned in training. In our study, the number of neural network layers for obtaining ϕ_{ij}^{CF} is set as $H = 4$ with node sizes of 64, 32, 16, and 8 for the educational data mining dataset and $H = 2$ with node sizes of 16 and 8 for the educational assessment dataset.

4.2.2. Deep learning of problem-solving process

The deep learning architecture for the process data is similar to the deep CF as we introduced above. However, given that the data of learners' actions and time durations in problem solving are of a sequential nature, we adopt LSTM networks to discover the time-series dependencies between problem solving steps. Specifically, the bottom of the architecture shows the inputs for the framework, which are learners' raw action and time sequences. Given a learner $\mathbf{m}_i$'s action sequence $a_{ij} = \{e_1, e_2, ..., e_Q\}$ in solving problem $\mathbf{n}_j$ with Q steps, since the categorical actions cannot be directly used for training, we first transform each action e_q in a_{ij} to a sparse binary vector by one-hot encoding, which are then embedded as a dense vector of d_0 dimensions. Subsequently, the embedded action vectors are fed into a LSTM layer to update the hidden state $h_q \in \mathbb{R}^{d_h}$ of each action e_q at the q-th problem-solving step with the previous hidden state h_{q-1} in a recurrent way as:

$$\begin{aligned}
f_q &= \sigma(W_f[h_{q-1}, e_q] + b_f), \\
i_q &= \sigma(W_i[h_{q-1}, e_q] + b_i), \\
C_q &= f_q \cdot C_{q-1} + i_q \cdot \text{tanh}(W_C[h_{q-1}, e_q] + b_C), \\
o_q &= \sigma(W_o[h_{q-1}, e_q] + b_o), \\
h_q &= o_q \cdot \text{tanh}(C_q),
\end{aligned} \tag{9}$$

where f_q, i_q and o_q denote the forget gate, the input gate, and the output gate of an LSTM cell respectively. C_q denotes the cell state, or the current history, at the q-th step. σ is the sigmoid activation as we mentioned earlier and tanh is the hyperbolic tangent activation, which is another type of non-linear transformation. Moreover, W_f and b_f, W_i and b_i, and W_o and b_o indicate the weights and bias of the forget gate, the input gate, and the output gate respectively. The weight matrix of each gate is of a shape $d_h \times d_0$, and the bias matrix of each gate is of a shape d_h. It should be noted that the flexibility of LSTM networks for modeling time-series dependencies is attributable to the three gates controlling the information of inputs, the information to be remembered or forgotten in the internal cell state, and the information of outputs. For LSTM layers, the dimensionality of the output space (i.e., the number of LSTM units d_h) is a hyper-parameter to be tuned in training. In addition, we can stack multiple LSTM layers to capture a higher complexity of action dependencies. The same structure of LSTM layers also applies to the time sequences in our deep learning architecture.

Subsequently, the LSTM outputs of action and time sequences A_{ij} and T_{ij} are concatenated as the input fed into a multi-layer neural network architecture which is similar to the one in deep CF. Specifically, the output of the last neural network layer ϕ^{Log} is the representation of the problem solving process learned by the deep learning structure, which can be formulated as:

$$\phi_{ij}^{\text{Log}} = f_H(\mathbf{W}_H^T f_{H-1}(...f_2(\mathbf{W}_2^T f_1(\mathbf{W}_1^T \begin{bmatrix} A_{ij} \\ T_{ij} \end{bmatrix}))...)). \tag{10}$$

In our study, the number of neural network layers for obtaining ϕ_{ij}^{Log} is set as $H = 4$ with node sizes of 64, 32, 16, and 8 for the educational data mining dataset and $H = 2$ with node sizes of 16 and 8 for the educational assessment dataset.

4.2.3. Prediction

The topmost layer in our general framework of LogCF depicts the combination of the outputs of deep CF and process data for the final prediction of learners' probabilities of correctly solving assessment tasks. Specifically, we concatenate ϕ_{ij}^{CF} and ϕ_{ij}^{Log} as the input of a neural network layer with one-dimensional output. Finally, we use a sigmoid activation to transform the output values as success probabilities which are $\in [0, 1]$.

4.2.4. Variants of LogCF

To evaluate the capacity of LogCF to learn or refine the expert-specified item-skill associations, we derived three variants of LogCF with the same topology shown in Figure 1. These variants mainly differ in how the weights of the deep CF architecture in the model are initialized, which can be categorized as variants with or without expert-specified item-skill associations. Variants with the expert-based information are:

- **expert-Q:** In this model, the item-skill associations $\mathbf{V}$ are fixed as the ones pre-defined by experts, and all the other weights of the deep CF architecture are adjustable. This variant predicts unseen learning outcomes totally relying on expert-specified item-skill associations.

- **expert-Q-init:** In this model, the item-skill associations $\mathbf{V}$ are initialized with the ones pre-defined by experts, and they can be adjusted in training.

The variant without the expert-based information is:

- **random-init:** A uniform distribution ranging in $(\sqrt{-6/(N_i + N_o)}, \sqrt{6/(N_i + N_o)})$ is used to initialize all weights of the deep CF architecture including item-skill associations, where N_i and N_o denote the input size and the output size respectively (Glorot and Bengio, 2010).

It should be noted that all three variants of LogCF are learned with the same pre-training of the process data architecture in the model. In other words, the process data architecture of LogCF is first trained as an independent model for predicting item responses (i.e., the Log method shown in the section on baselines). In the three variants, the weights associated with actions and time durations are then initialized with the weights learned by the pre-trained process data learning architecture.

4.2.5. LogCF learning

In LogCF, the model parameters to be updated include the item, learner, and action embedding weights, the weights of the multi-layer neural network architectures for learning the learner-item interaction and the problem-solving process, and the weights of the last neural network layer for the final prediction. Taking the negative logarithm of the likelihood shown in equation (6), we can obtain the objective function for our problem, which is the binary cross-entropy loss:

$$J = -\sum_{i,j} R_{ij}\log\hat{R}_{ij} + (1 - R_{ij})\log(1 - \hat{R}_{ij}), \tag{11}$$

where $\hat{R}_{ij}$ indicates the probability of correct responses predicted by the model. As such, the parameters of LogCF can be directly learned by minimizing the objective function. Specifically, we adopt the optimization method of Adaptive Moment Estimation (Adam; Kingma and Ba 2014) for training the model, which is widely-used optimizer in deep learning, capable of adapting individual learning rates for each parameter.

4.2.6. A hypothetical example

To further illustrate how LogCF works for item response modeling, we build a hypothetical example shown in Figure 2. In this example, we assume a computer-based assessment of problem-solving competency which involves five learners and three assessment tasks. Learners' true item responses are shown at the top right corner of Figure 2. It can be seen that learners 1 and 2 get one item correct, learners 3 gets two items correct, learner 4 gets all the three items correct, and learner 5 gets all the items incorrect. With LogCF, our goal is to estimate learners' predicted probabilities of correctly answering each item, which is the output of LogCF (see the top matrix of predicted probabilities). In addition, we also hope to estimate the item- and learner-skill associations which are shown at the bottom left corner of Figure 2 (item-skill associations are fixed in the variant of expert-Q). It can be seen that learners 3 and 4 have relatively stronger associations with the latent skill than the other learners, indicating their higher problem-solving proficiency. Among the three items, item 2 has the strongest association with the latent skill. The item- and learner-skill associations are embedded from the five learner IDs and the three item IDs, and they are finally represented as some vectors ϕ_{ij}^{CF} learned by a multi-layer neural network architecture. Moreover, for each task, learners' problem-solving action steps and the time spent for each step are logged by the system; these are presented at the bottom right corner of Figure 2. These action and time sequences are finally represented as some vectors ϕ_{ij}^{Log} learned by a deep learning architecture with multiple LSTM and neural network layers. The two representations ϕ_{ij}^{CF} and ϕ_{ij}^{Log} (shown in the middle of Figure 2) are used to produce the predicted probabilities of correctly answering each item.

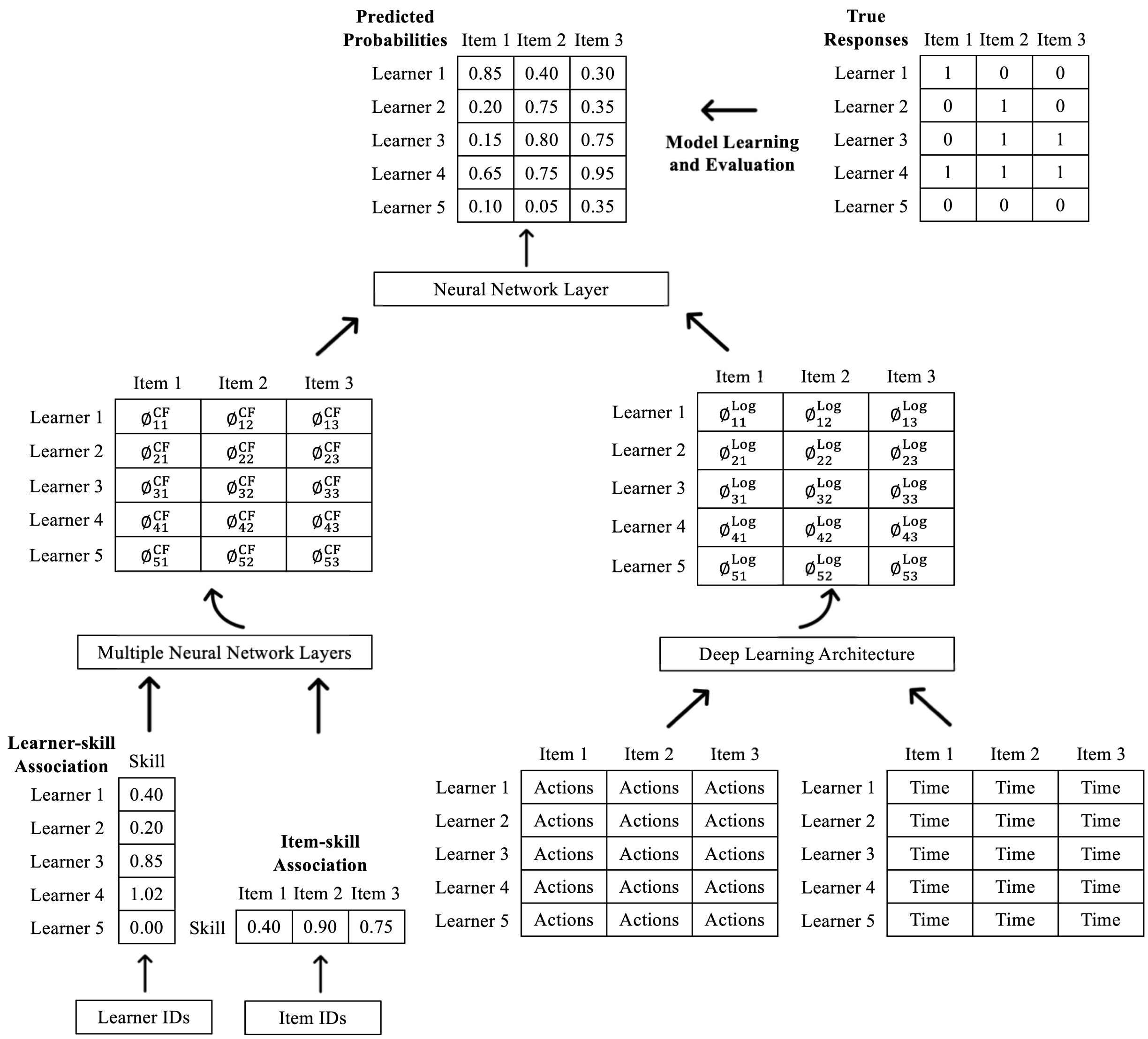

Figure 2: A hypothetical example of LogCF.

5. EXPERIMENTS

In this section, we conduct extensive experiments with two datasets of distinctive characteristics to demonstrate the effectiveness of LogCF with the following five research questions:

1. Does LogCF outperform other recent educational data mining approaches and psychometric measurement models?

2. Do LogCF variants for learning item-skill associations from scratch have higher prediction performance than LogCF variants incorporating expert-specified item-skill associations?

3. How does the number of latent skills affect the predictive power of LogCF?

4. Does LogCF perform well at different levels of missing responses or at different percentages of learners' first item responses used for training?

5. Are learner- and item-skill associations estimated by LogCF interpretable in the context of educational assessment?

Specifically, we compare the performance of LogCF with educational data mining and psychometric measurement models in predicting missing responses, under the condition of different training/test partition rates. In addition, for the educational data mining dataset, we demonstrate the capacity of LogCF to retrieve or refine expert-specified item-skill associations and how its prediction performance is affected by the number of latent skills. For the educational assessment dataset, we demonstrate the interpretability of learner- and item-skill associations in comparison with the item and learner parameters estimated by IRT models, as well as the theoretical guidance of the assessment. In the following sections, we first present the experimental setup, followed by presenting the results to answer the above five research questions.

5.1. Dataset description

The educational data mining dataset is a web-based tutoring dataset named "Lab study 2012 (cleanedLogs)" under the project "Fractions Lab Experiment 2012" led by Vincent Aleven, accessed via the PSLC DataShop (Koedinger et al., 2010), available at `https://pslcdatashop.web.cmu.edu/`. The dataset involves 74 learners, 19 knowledge components (i.e., latent skills), a total of 14,959 problem-solving steps, and a total of 37,889 transactions. Learners attempted to solve a set of mathematical problems on fractions, and they sometimes accessed different problem sets in terms of both test length and problem content. Solving each problem required learners to take a set of problem-solving steps, which can be considered as items. For each step, a number of transactions are involved. Transactions refer to learners' interactions with the tutoring system and each transaction represents one action. Learners might take several actions to successfully attempting a step, which are indicators of learners' problem-solving processes. In the dataset, each action is associated with a time duration. The dataset is preprocessed for our experiments. Specifically, we remove all transactions performed by the tutor and all transactions without time durations. In addition, steps with an outcome of "correct" or "incorrect" are considered as items, and steps with an outcome of "hint" are considered as intermediate actions for solving a step. To improve the differentiability of actions, we redefine actions as learners' actions combined with their selections. In the final dataset, approximately 73% of item responses are correct.

The educational assessment dataset is from the Programme for International Student Assessment (PISA; `https://www.oecd.org/pisa`) in 2012. PISA is an international large-scale assessment across different nations and economies measuring 15-year-old students' literacy in mathematics, reading, and mathematics. It is conducted by the Organization for Economic Cooperation and Development (OECD) every three years with a particular emphasis on one of the three competence fields at each cycle. In 2012, PISA measured mathematics literacy in approximately 470,000 students from 65 countries and economies (OECD, 2014b). In addition, PISA 2012 also evaluated students' creative problem-solving competency. The assessment items of creative problem solving were contextualized, real-life tasks. Students needed to engage in different strategies to generate a fully correct solution. An example item about how to use an air conditioner can be accessed at `http://www.oecd.org/pisa/test-2012/testquestions/question3/`. In this item, students were asked to figure out which air conditioner controls affect temperature and humidity by adjusting different control values. The corresponding effect of each adjustment is reflected by the changes in temperature and humidity. As such,

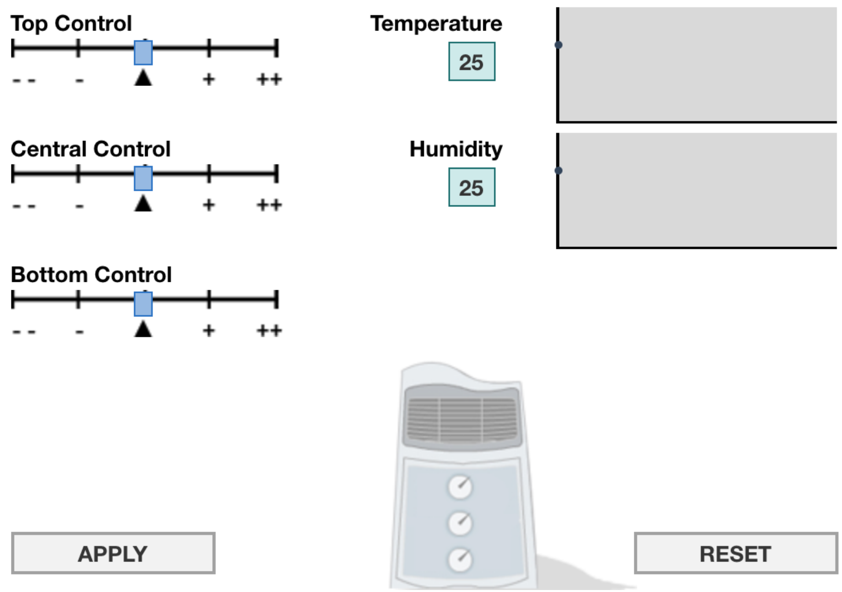

Figure 3: Screenshot of the interface of a problem solving item about climate control.

in this item, students' actions include each attempt of control adjustment, their resetting actions, and their attempts to draw lines between controls and temperature and humidity for answering the question. The interface of this item is presented in Figure 3.

In addition to students' item scores on these tasks, PISA also published students' process data in solving some assessment problems. The process data can be accessed at https://www.dropbox.com/s/b8kb4jmqnha6jom/CPRO_logdata_released.zip?dl=0. Given that students only took a subset of all the assessment items, to ensure a large sample size for deep learning and demonstrate how LogCF can be used in the psychometric measurement settings, we subset the data to four items from the publicly available datasets. In the problem-solving assessment, although there were a few different items, some items were not suitable for process data learning because their item scoring was solely based on students' problem-solving action sequences. That said, the process data involved correct item responses. Therefore, we only subset a limited number of items for the experiment. The experimental dataset in our study includes 10,070 students' item scores and process data on each of the four problem-solving items. The four items measure the same latent competency of complex problem solving. Students were not given explicit hints when solving the problems and were not able to attempt the same problem multiple times. In addition, according to the PISA scoring rubric, students would be given a credit of 2 when they correctly solved a problem, a partial credit of 1 when they showed some correct actions but did not fully solve the problem, and a credit of 0 when they did not show any correct actions and incorrectly solve the problem. In our study, we recode a full credit as 1 and a partial or incorrect credit as 0. For the process data, we define an action as an event (e.g., draw a line between a control and humidity or temperature) along with its corresponding event type (e.g., the type of "Diagram") in the log file, and we calculate the time duration for each action by taking the difference between two consecutive timestamps. Moreover, we hide the actions showing the details on students' final solutions on solving an item because they were associated with item scores.

It should be noted that the two datasets used in our study differ in a range of characteristics. First, the "Lab study 2012" dataset is an unstructured web-based tutoring dataset while the PISA 2012 dataset is a structured standardized assessment dataset. Second, the "Lab study 2012" dataset involves substantially fewer learners but more items for each learner than the PISA 2012 dataset. In other words, in the "Lab study 2012" dataset, each learner is associated with many items and problem-solving actions, and in the PISA 2012 dataset, each item is associated with many learners and problem-solving actions. Third, the two datasets assess largely different competencies. Assessment items on fractions from the "Lab study 2012" dataset are much more straightforward than items on complex problem solving from the PISA 2012 dataset. It is reasonable to posit that actions and time durations for the "Lab study 2012" dataset might be less useful for predicting learners' problem-solving successes than those for the PISA 2012 dataset. Finally, the "Lab study 2012" dataset involves much more latent skills than the PISA 2012 dataset. The PISA 2012 dataset only measures one major competency which is however very common in achievement tests in education.

5.2. Experimental setup

5.2.1. LogCF training setting

The following hyperparameters of LogCF were tuned in our experiments. First, we conducted a hyperparameter search on three candidate regularization weights for item and learner embedding layers (i.e., item- and learner-skill associations), 0, 0.001, and 0.1, and selected 0.001 for the "Lab study 2012" dataset and 0 for the PISA 2012 dataset. In our experiments, if large regularization weights were imposed on item and learner embeddings, the estimated item- and learner-skill associations would be too concentrated around 0. This is especially not desired for the interpretability of learner-skill associations because learners could not be differentiated very well in terms of latent ability levels. Therefore, we chose small regularization weights. Second, dropout layers (Srivastava et al., 2014) were applied prior to each neural network layer to prevent overfitting. Among candidate values of 0, 0.2, and 0.5, the dropout rate was finalized as 0.2 for the "Lab study 2012" dataset and 0 for the PISA 2012 dataset. The output dimension for the embedding layer for actions was searched across 8, 16, 32, and 50, and 16 and 50 were used for the "Lab study 2012" and PISA 2012 datasets respectively. Regarding the depth and nodes of the neural network and LSTM layers in LogCF, for the "Lab study 2012" dataset, we applied the same four-layer architecture (i.e., $H = 4$) to both the deep CF and process data learning parts, with node sizes of 64, 32, 16, and 8; for the PISA 2012 dataset, we applied the same two-layer architecture (i.e., $H = 2$) to both the deep CF and process data learning parts, with node sizes of 16 and 8. Moreover, we conducted a hyperparameter search on three candidate learning rates, 0.001, 0.01, and 0.1, and used 0.001 for LogCF. In addition, the batch size is set as 64 and the number of epochs is set as 60 for training. Early stopping was applied to prevent overfitting.

With respect to preprocessing the actions and time durations, the maximum action and time sequence lengths for each item were set as 54 and 161 for the "Lab study 2012" and PISA 2012 datasets respectively, given that the items with the most actions have lengths of 54 and 161 for the two datasets. Items with fewer actions were padded with zeros. In addition, time durations were scaled with min-max normalization such that they were in a range between zero and one.

Regarding the number of latent skills, it should be noted that the three variants of LogCF based on the expert-specified item-skill associations have a fixed number of latent skills and the latent skill dimensions can be tuned for the two variants of LogCF with random initialization.

To reveal how the latent skill dimension affects the predictive power of LogCF, we compared the model performance with the following number of later skills: $k/4$, $k/2$, $k - k/10$, k, $k + k/10$, and $k \times 2$ (Pardos and Dadu, 2018), where k is the original expert-specified number of latent skills. As such, the candidate values of latent skill dimensions were set as 5, 10, 17, 19, 21, and 38 for the "Lab study 2012" dataset. The number of latent skills was set as 1 for the PISA 2012 dataset given that the four items were designed to measure a single latent skill according to the PISA 2012 assessment framework.

The proposed framework was implemented using Keras (Chollet et al., 2015).

5.2.2. Baselines

We compared LogCF with the following approaches in terms of predictive power:

- **NeuralCF.** The NeuralCF method is the deep learning-based CF architecture shown in Figure 1. The output of the last neural network layer is directly fed into the prediction module without concatenating with the output of the process data learning architecture. Given that NeuralCF is a sub-architecture of LogCF, the three variants of LogCF also apply to NeuralCF. Thus, the same three variants of NeuralCF were used for comparison in our experiments. It should be noted that the comparison between LogCF and NeuralCF is similar to an ablation study because removing the process data learning architecture of LogCF results in NeuralCF. As such, it is expected that LogCF should outperform NeuralCF because learners' actions and time durations for solving problems are considered important features for prediction in our study.

- **Log.** The Log method is the process data learning architecture shown in Figure 1. The output of the last neural network layer is directly fed into the prediction module without concatenating with the output of the deep learning-based CF architecture. It should be noted that removing the deep learning-based CF architecture of LogCF results in Log.

- **Additive Factors Model (AFM).** AFM (Cen et al., 2005; Cen et al., 2006) is a statistical model for modeling learners' probabilities of successfully solving items. Given the model, learner i's probability of correctly answering item j can be formulated as

$$P(R_{ij} = 1|\theta_i) = \frac{1}{1 + e^{-(\theta_i + \sum_{k=1}^{K} \beta_k q_{jk} + \sum_{k=1}^{K} \gamma_k q_{jk} t_{ik})}} \tag{12}$$

  where θ_i denotes learner i's latent ability, β_k denotes the easiness of skill $k \in \{1, ..., K\}$, γ_k indicates the learning rate of skill $k \in \{1, ..., K\}$, q_{jk} indicates the mapping between item j and skill k, t_{ik} is the total number of opportunities learner i has previously practiced skill k, and K is the total number of latent skills. It should be noted that an expert-specified Q-matrix representing item-skill associations q_{jk} is required by AFM.

- **dAFM.** dAFM is a dynamic and deep variant of AFM (Pardos and Dadu, 2018). dAFM is derived based on two major changes to AFM: the weights of item-skill associations q_{jk} in dAFM are adjustable rather than fixed, and the learning opportunity counts for each latent skill t_{ik} are not fixed inputs but dynamically calculated as the item-skill associations change in training. dAFM is developed based on a deep learning framework with a recurrent neural network (RNN) layer as a learning opportunity counter. According to their experimental results (Pardos and Dadu, 2018), we used two variants of dAFM, fine-tuned and QkDense, as baselines for comparison.

- **Deep Knowledge Tracing (DKT).** DKT (Piech et al., 2015) is a popular RNN-based learning outcome modeling method used in educational data mining. In DKT, learners' sequential item responses are used as inputs, which indicate what items were answered and if they were answered correctly by learners. The inputs are represented by a one-hot encoding of item responses given a small number of unique items or a low-dimensional representation of item responses given a large number of unique items. The outputs of DKT are vectors indicating learners' probabilities of getting each unique item correct. As such, DKT predicts a learner's next item response simply by retrieving his or her success probability on the item from the probability vector predicted by learning his or her history item responses. Given the RNN framework, the temporal dependencies between learners' history item responses can be exploited to enhance the prediction of future item responses. It should be noted that items in DKT are represented by their associated latent skills rather than the questions accessed by learners. As such, each learning opportunity is associated with a unique latent skill in DKT.

The above models were implemented with the "Lab study 2012" dataset as baselines for evaluating the predictive power of LogCF. For the PISA 2012 dataset, given that PISA achievement scores and questionnaire construct scores were derived based on IRT models, we analyzed the data with the following three IRT models for both the purposes of evaluating the predictive power and interpretability of LogCF.

- **Rasch.** The Rasch model (Rasch, 1980) is the most parsimonious IRT model only with an item difficulty parameter. By Rasch model, learner i's probability of correctly answering item j can be formulated as

$$P(R_{ij} = 1|\theta_i) = \frac{1}{1 + e^{-(\theta_i - b_j)}} \tag{13}$$

  where θ_i denotes learner i's latent ability and b_j denotes item j's difficulty.

- **2PL.** The 2PL model (Embretson and Reise, 2000) is another IRT model that parameterizes both item difficulty and item discrimination. By 2PL model, learner i's probability of correctly answering item j can be formulated as

$$P(R_{ij} = 1|\theta_i) = \frac{1}{1 + e^{-a_j(\theta_i - b_j)}} \tag{14}$$

  where θ_i, a_j, and b_j denote learner i's latent ability, item j's discrimination, and item j's difficulty respectively.

- **3PL.** Similar to the 2PL model, the 3PL model (Embretson and Reise, 2000) also parameterizes item difficulty and item discrimination, but it additionally parameterizes item guessing. By 3PL model, learner i's probability of correctly answering item j can be formulated as

$$P(R_{ij}=1|\theta_i)=c_j+\frac{1-c_j}{1+e^{-a_j(\theta_i-b_j)}} \tag{15}$$

where c_j in the model denotes item j's guessing.

5.2.3. Evaluation

We evaluated the model performance for different training/test partition ratios. Specifically, each model was evaluated with 40%, 30%, 20%, and 10% of all item responses going for test and the remaining going for training. In addition, 20% of the training samples were used as a validation set in training. It should be noted that the training/test partition was conducted at the entry level such that each item response and its associated actions and time durations were considered as one independent sample for training. However, given that item responses in the "Lab study 2012" dataset were of a temporal nature, we also split the "Lab study 2012" dataset by taking learners' first item responses for training and the remaining item responses for test. Unlike the entry-level training/test partition, a set of consecutive item responses for each learner was used for training or test in the sequential training/test partition. To reveal how early LogCF is capable of predicting learners' future item responses, we took learners' first 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% item responses as training samples and the remaining item responses as test samples.

We evaluated the performance of each approach from both the regression and classification perspectives. For classification, we used accuracy (ACC) and the area under the receiver operating characteristic curve (AUC; Ling et al. 2003). ACC is simply the ratio of the number of correct predictions to the total number of predictions. For a predicted probability $P \in [0,1]$, a cut-off value of 0.5 is typically used to predict a sample as positive or negative for calculating ACC. However, AUC does not assume any cut-off values for evaluation. It is the area under the plot of sensitivity against the false positive rate (1 minus specificity) for different cut-off values, which is insensitive to the issue of class imbalance. For regression, we use the mean absolute error (MAE) and the root mean square error (RMSE) as the evaluation metrics (Willmott and Matsuura, 2005). In our case, given a predicted probability P_{ij} of learner i correctly answering item j with a ground truth R_{ij} and a total of N predictions, MAE is given by

$$\text{MAE}=\frac{\sum_{i,j}|P_{ij}-R_{ij}|}{N}, \tag{16}$$

and RMSE is given by

$$\text{RMSE}=\sqrt{\frac{\sum_{i,j}(P_{ij}-R_{ij})^2}{N}}. \tag{17}$$

5.3. Experimental results

5.3.1. Main prediction results

Table 1 shows the testing performance of each model for the "Lab study 2012" dataset in terms of each evaluation metric given different training/test partition ratios (at the entry level). Generally, regardless of training/test partition ratios, variants of LogCF show slightly higher ACC and AUC rates and slightly lower MAE and RMSE rates than the majority of baselines, indicating that

Table 1: Model performance across training/test partition ratios for the "Lab study 2012" dataset.

Training	Model	Variant	ACC	AUC	MAE	RMSE
90%	LogCF	expert-Q	**0.7645**	**0.7706**	0.3458	**0.3995**
		expert-Q-init	0.7552	0.7628	0.3376	0.4012
		random-init	0.7614	0.7680	0.3445	0.4007
	Neural CF	expert-Q	0.7485	0.7353	0.3536	0.4088
		expert-Q-init	0.7497	0.7278	0.3488	0.4093
		random-init	0.7485	0.7281	0.3474	0.4088
	Log	N/A	0.7580	0.7060	0.3710	0.4150
	DKT	N/A	0.7389	0.7595	**0.3355**	0.4083
	AFM	N/A	0.7262	0.6548	0.3670	0.4249
	dAFM	Fine-tuned	0.7274	0.7201	0.3445	0.4149
		QkDense	0.7303	0.6676	0.3683	0.4233
80%	LogCF	expert-Q	**0.7596**	0.7575	0.3355	**0.4014**
		expert-Q-init	0.7562	0.7525	0.3299	0.4024
		random-init	**0.7596**	0.7525	0.3365	0.4035
	Neural CF	expert-Q	0.7414	0.7094	0.3546	0.4148
		expert-Q-init	0.7401	0.7099	0.3495	0.4150
		random-init	0.7303	0.7074	0.3540	0.4164
	Log	N/A	0.7540	0.7148	0.3620	0.4143
	DKT	N/A	0.7537	**0.7599**	**0.3290**	0.4016
	AFM	N/A	0.7247	0.6610	0.3713	0.4253
	dAFM	Fine-tuned	0.7432	0.6858	0.3484	0.4133
		QkDense	0.7397	0.6664	0.3636	0.4196
70%	LogCF	expert-Q	**0.7483**	0.7523	0.3375	0.4072
		expert-Q-init	0.7468	0.7517	**0.3322**	0.4075
		random-init	0.7462	0.7561	0.3336	**0.4063**
	Neural CF	expert-Q	0.7337	0.7036	0.3611	0.4207
		expert-Q-init	0.7322	0.7038	0.3532	0.4207
		random-init	0.7322	0.7000	0.3539	0.4218
	Log	N/A	0.7437	0.7147	0.3638	0.4183
	DKT	N/A	0.7410	**0.7614**	0.3352	0.4073
	AFM	N/A	0.7235	0.6518	0.3660	0.4263
	dAFM	Fine-tuned	0.7319	0.6865	0.3581	0.4178
		QkDense	0.7309	0.6543	0.3636	0.4240
60%	LogCF	expert-Q	0.7494	0.7479	0.3349	0.4084
		expert-Q-init	0.7449	0.7498	0.3302	**0.4081**
		random-init	**0.7515**	0.7470	**0.3244**	0.4093
	Neural CF	expert-Q	0.7337	0.7016	0.3585	0.4214
		expert-Q-init	0.7277	0.6987	0.3517	0.4223
		random-init	0.7301	0.7057	0.3496	0.4206
	Log	N/A	0.7423	0.7039	0.3590	0.4207
	DKT	N/A	0.7383	**0.7522**	0.3374	0.4111
	AFM	N/A	0.6872	0.5817	0.3936	0.4516
	dAFM	Fine-tuned	0.7307	0.6749	0.3561	0.4203
		QkDense	0.7203	0.6532	0.3722	0.4272

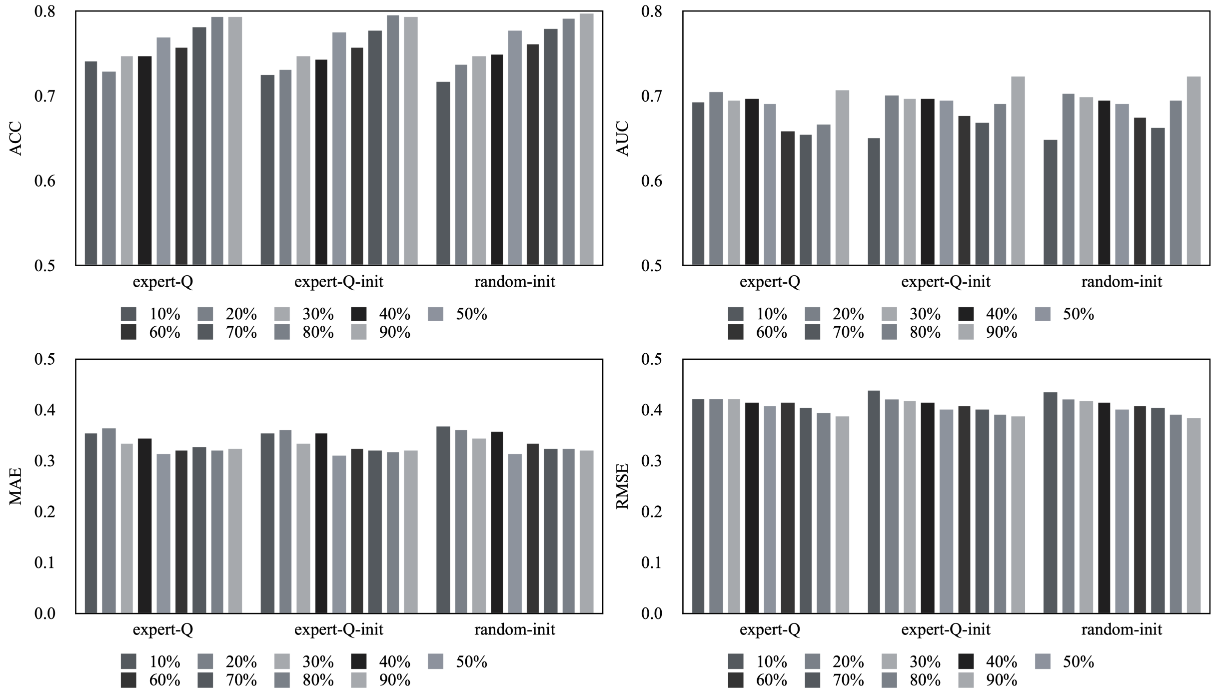

Figure 4: Model performance of LogCF at different percentages of learners' first items going for training.

LogCF slightly outperforms other models in terms of predictive power. More concretely, it can be seen that using more samples for training slightly improves the prediction performance of LogCF since ACC and AUC rates are slightly higher and MAE and RMSE rates are slightly lower when 90% and 80% item responses go for training. However, the differences between training/test partition ratios are trivial for LogCF. The negative influences of low training/test partition ratios on the performance of AFM and dAFM are more explicit. When fewer samples go for training, the prediction accuracy of AFM and dAFM is greatly reduced.

Moreover, to answer research question 4, we also evaluated the model performance given different levels of learners' first item responses going for training. In Figure 4, it can be seen that for all variants of LogCF, more first item responses going for training results in higher ACC and AUC rates and lower MAE and RMSE rates. This indicates that LogCF is more likely to successfully predict learners' future item responses when more learning history is available for training. However, even if very few history item responses are available (e.g., 10%), LogCF variants still show acceptable predictive capacity, which might be due to the contribution of learners' problem-solving processes. Compared with the training/test partition at the entry level, the prediction performance of LogCF, especially for AUC, is deteriorated when data is split in a sequential way. This is however an unsurprising finding given that the class weights for training and test datasets might be largely different in this case.

Figure 5 shows the model performance for the PISA 2012 dataset with respect to each evaluation metric at different levels of missing responses. It can be seen that compared with the IRT models, LogCF performs the best under all the experimental conditions. Concretely, in terms of ACC and AUC, LogCF has an approximate ACC rate ranging from 0.83 to 0.85 and an approximate AUC rate ranging from 0.91 to 0.93 across different levels of missing responses. Generally,

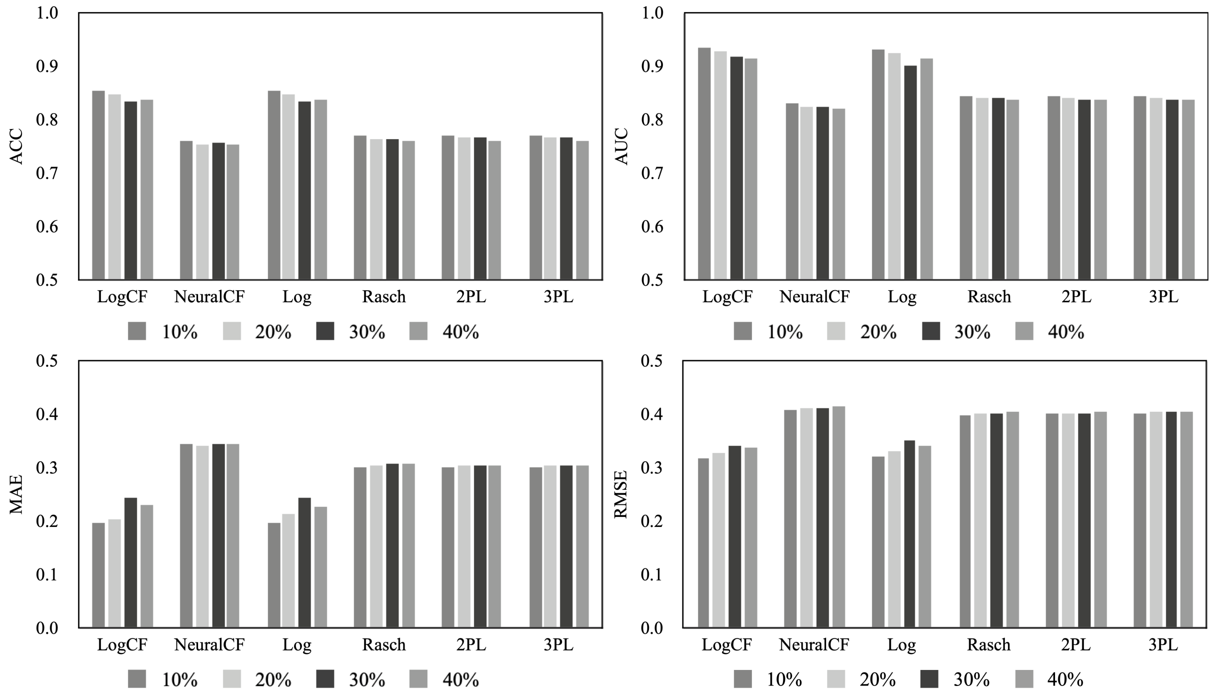

Figure 5: Model performance at different levels of missing responses for the PISA 2012 dataset.

the IRT models have medium but much lower ACC and AUC rates than LogCF. Moreover, the three IRT models do not show significant differences in prediction. Regarding the regression metrics, compared with the IRT models, LogCF shows much lower MAE and RMSE rates at each level of missing responses.

With respect to the comparison between LogCF and NeuralCF, as expected, given both datasets, LogCF variants outperform NeuralCF under various experimental conditions. This indicates that the process data learning architecture of LogCF significantly contributes to prediction. Particularly, the difference in prediction performance between LogCF and NeuralCF is larger for the PISA 2012 dataset than that for the “Lab study 2012” dataset. This implies that learners’ actions and time durations contribute more to their problem-solving successes for the PISA 2012 dataset than is the case for the “Lab study 2012” dataset. As we mentioned earlier, the PISA 2012 study evaluated learners’ competencies on complex problem solving, while “Lab study 2012” tested learners’ knowledge of fractions. This explains why the process data is more influential for the PISA 2012 study than that for “Lab study 2012”.

Notably, the comparison between LogCF and Log indicates that the prediction performance of Log was as good as that of LogCF for the PISA 2012 dataset (the higher performance of LogCF was negligible). However, for the Lab study 2012 dataset, the performance of Log, especially AUC, was much worse than that of LogCF. Therefore, we considered that despite the fact that Log learns a lot from data, adding the deep learning-based CF architecture is still of value, because it is capable of improving the prediction performance and estimating item- and learner-skill associations.

In general, according to the experimental results of the two datasets, from both the classification and regression perspectives, our approach LogCF demonstrates a substantially higher prediction performance than the baselines. Moreover, its prediction performance would not be

Table 2: Model performance of LogCF (random-init) given different numbers of latent skills.

Training	Metric	5 skills	10 skills	17 skills	19 skills	21 skills	38 skills
90%	ACC	0.7608	0.7565	0.7398	**0.7614**	0.7515	0.7602
	AUC	**0.7772**	0.7712	0.7605	0.7680	0.7593	0.7641
	MAE	0.3413	0.3361	0.3437	0.3445	0.3414	**0.3339**
	RMSE	**0.3981**	0.3987	0.4033	0.4007	0.4016	0.3992
70%	ACC	0.7398	0.7411	0.7396	**0.7462**	0.7390	0.7261
	AUC	0.7551	0.7554	0.7451	**0.7561**	0.7417	0.7015
	MAE	0.3517	0.3380	0.3369	**0.3336**	0.3383	0.3513
	RMSE	0.4095	0.4067	0.4101	**0.4063**	0.4108	0.4224

greatly affected by the missing response rates, indicating the robustness of LogCF.

5.3.2. Performance of learning or refining item-skill associations

With regard to research question 2, in general, variants incorporating expert-specified item-skill associations show negligibly better prediction performance than the variant without expert information given that expert-Q and expert-Q-init show the highest prediction performance more frequently. This implies that unfortunately, item-skill associations learned by LogCF from scratch are not superior to the original Q-matrix defined by experts. However, given their similar prediction results, it is safe to conclude that item-skill associations learned by LogCF are not worse than the original expert-specified ones. Particularly, it can be seen that when fewer item responses go for training, the variant of expert-Q is slightly less competitive than other variants of LogCF.

5.3.3. Effects of the number of latent skills on LogCF

To answer research question 3, the LogCF variants without expert information were further evaluated with different latent skill dimensions. Table 2 presents the testing performance of random-init and FT-random for the "Lab study 2012" dataset given different numbers of latent skills at the missing response levels of 70% and 90%. Overall, the effect of the number of latent skills on the predictive power of LogCF is not significant, given that the model performance remains stable with the increase of latent skill dimensions.

5.3.4. Interpretability of LogCF

The interpretability of a model is especially beneficial to educational practitioners. In our study, LogCF is developed based on CF which estimates latent factors of learners and items. Specifically, in LogCF, the learner-skill association can be interpreted as learners' mastery levels of the targeted skills, which can be used to diagnose learners' learning outcomes; the item-skill association can be interpreted as the degree to which items measure the targeted skills, which can be used to organize learning and evaluation materials. In psychometric measurement models, a parallel concept of the learner-skill association is learners' latent ability levels, and parallel concepts of the item-skill association are item difficulty and item discrimination. Item difficulty corresponds to the point of the learner ability scale at which a learner of the same ability has

Table 3: Item-skill associations or item parameters estimated by LogCF and baselines (missing rate = 30%).

	Item 1	**Item 2**	**Item 3**	**Item 4**
LogCF	-0.07	-1.06	1.42	0.59
NeuralCF	-0.02	-1.20	1.51	0.65
Rasch (intercept d_j)	-0.17	-2.23	2.23	1.40
2PL (discrimination a_j)	1.53	1.84	1.59	1.10
2PL (intercept d_j)	-0.18	-2.50	2.32	1.26
3PL (discrimination a_j)	2.49	1.67	1.72	1.11
3PL (intercept d_j)	-0.87	-2.38	2.41	1.26

a 50% chance of correctly answering the item, and item discrimination corresponds to the capability of an item to differentiate learners by their abilities. As shown in equations 13 to 15, the linear combination of learner ability, item difficulty, and item discrimination with a sigmoid transformation models a learner's probability of correctly answering an item. Given that learner ability and item difficulty are on the same scale in IRT models, whether a learner is able to get an item correct is affected by both item discrimination and the difference between their ability and the item difficulty (i.e., the product of two). LogCF, however, models a single item parameter, which therefore can be considered as the similar item parameter as the product of item difficulty and item discrimination. A high item-skill association indicates that the item is strongly related to the targeted skill and a strong mastery of the targeted skill is required to get it correct.

ITEM-SKILL ASSOCIATION. Given the above theoretical clarification, we further compare the item-skill associations estimated by LogCF with the item parameters estimated by the baselines under the condition of 30% missing responses as an example for illustrating the interpretability of LogCF. As shown in Table 3, LogCF suggests that items 1 and 2 have negative item-skill associations and items 2 and 3 have positive item-skill associations. For the IRT models, we present both the item discrimination a_j and the item intercept $d_j = -a_j \times b_j$ in Table 3. Item intercept can be considered as a combination of item discrimination and item difficulty. Generally, the IRT models suggest that items 1, 2, and 3 have higher item discriminations than item 4. In terms of the item intercept, items 1 and 2 have negative values, and items 3 and 4 have positive values. The pattern of IRT item intercepts is similar to that of item-skill associations estimated by LogCF. To further visualize how item-skill associations resemble item intercepts, we present a line chart showing the pattern of item parameters of each method in Figure 6. It can be seen that the line of item-skill associations of LogCF follows a similar shape to the lines of item intercepts of the three IRT models. In addition, the item-skill associations of LogCF are highly correlated with the item intercepts of 2PL/3PL with a correlation coefficient of 0.96, indicating that they share almost the same interpretation.

In addition to the comparison of item parameters across methods, we also referred to the PISA 2012 assessment framework and results (OECD, 2014a) for more evidence validating our results. According to the PISA 2012 assessment framework, the four items involved in our analyses map onto different proficiency levels of complex problem solving at a scale ranging from below level 1 to level 6. Concretely, item 2 maps onto the second-highest level, level 5, which requires a high proficiency level of complex problem solving; item 1 maps onto level 3,

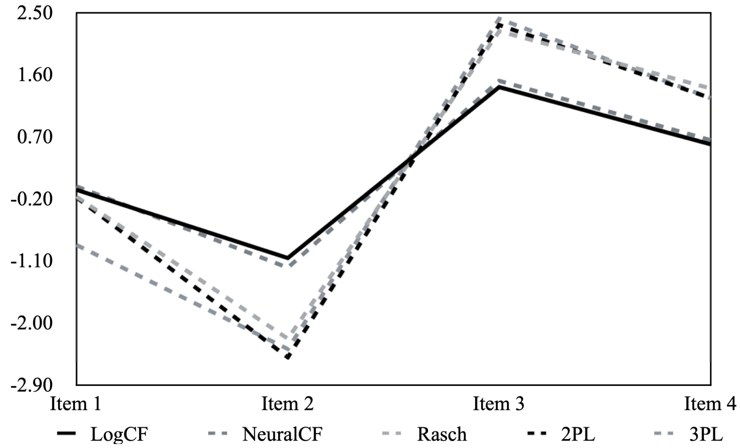

Figure 6: Plot of item-skill associations or item intercepts for LogCF and baselines. Lower values indicate higher proficiency levels required by items.

which requires a medium proficiency level of complex problem solving; and items 4 and 3 map onto level 1 and below level 1 respectively, which require a relatively lower proficiency level of complex problem solving. However, the above mapping of items to the theoretical proficiency levels does not necessarily suggest that items 1 and 2 measure the latent construct more strongly than items 3 and 4 do (i.e., items 1 and 2 have higher item discriminations than items 3 and 4). Instead, the mapping aligns with the item difficulty or the item easiness rather than the item discrimination. According to the theoretical framework, high-level items (items 1 and 2) are supposed to be more difficult than low-level items (items 3 and 4). This theoretical reasoning is validated by the IRT results given that the item difficulties of items 1 and 2 are much higher than those of items 3 and 4 (although we did not present item difficulties in the Table, they can be calculated as the negative values of item intercepts divided by item discriminations). However, we emphasize that the item-skill association estimated by LogCF is more of a parallel concept to item intercept, which is the negative product of item discrimination and item difficulty. In other words, item-skill associations estimated by LogCF incorporate both the information of how strongly an item measures the latent construct and how difficult the item is. According to Table 3, item difficulties have higher variance across the four items than item discriminations, which means that item difficulties are more influential for determining item-skill associations for the PISA 2012 items used in our study. In this sense, the mapping of these four items to the proficiency levels aligns with the magnitudes of item-skill associations estimated by LogCF, which further validates the interpretability of LogCF.

Admittedly, item-skill associations estimated by LogCF cannot be interpreted in the completely same way as item difficulties and item discriminations because the process data learning of LogCF also brings information to the estimation of item-skill associations. In this sense, item-skill associations might also incorporate additional information on learners' problem-solving processes. However, we consider the process data learning of LogCF as a regularization technique for training. As Figure 6 suggests, item-skill associations estimated by NeuralCF (without process data learning) are much less interpretable because they cannot be solved with a unique solution by NeuralCF. Therefore, estimating item-skill associations with NeuralCF is more of an ill-posed problem. As such, adding process data learning in LogCF regularizes the weights of item skill-associations as they provide more information in training.

Table 4: Correlation coefficients of learner-skill associations or latent ability estimates between LogCF and baselines (missing rate = 30%).

	LogCF	**NeuralCF**	**Rasch**	**2PL**
NeuralCF	0.99			
Rasch	0.74	0.73		
2PL	0.73	0.72	0.99	
3PL	0.73	0.72	0.99	1.00

LEARNER-SKILL ASSOCIATION. Learner-skill associations can be interpreted as learners' proficiency levels on the targeted skill. Given that the IRT models also provide learners' latent trait levels, we calculate the correlation coefficients of learner parameters between LogCF and the baselines, which are presented in Table 4. It can be seen that the learner-skill associations estimated by LogCF are highly correlated with those estimated by the CF-based methods and IRT models, which implies that the ranking of learners by LogCF is not very different from the ranking by the other methods.

6. LIMITATIONS

Although LogCF does not see any action sequences from the test sample in training, in the test stage, however, actions and durations on new items are needed for predicting unseen responses, which could be a limit in practice. However, we consider that the model could be of practical implications in the following circumstances. First, in psychometric measurement analysis of educational assessment data, typically we are interested in examining item quality and estimating learner abilities instead of predicting unseen item responses. In that case, LogCF could be used to evaluate items and learners as a "psychometric measurement model" which additionally exploits process data for modeling. Second, in the circumstance that predictions of future item responses are desirable, the model could be used with some modifications. For example, in the setting of massive open online courses, if we consider a course as an item, the process data would involve actions and associated durations over a long period (i.e., from registering the course to finishing the course). In that case, with partial process data, the model could detect at-risk students who would drop or fail the course at a very early stage, which is beneficial for early intervention. Moreover, even in conventional web tutoring settings, the model could still predict unseen and future item responses only based on the learned weights of the deep learning-based CF architecture, which were regularized by process data learning.

Despite the focus of LogCF on item- and learner-skill associations and predictive capacity, the process data learning of LogCF is not very informative for deciphering how learners attempt problems. We did not conduct the process data analytics mainly because LogCF does not model actions and time durations from the training learner-item interactions at the item level, but models them through embedding simultaneously. Given the current architecture of process data learning, analyzing action embeddings might be of limited information for uncovering learners' problem-solving processes. In addition, some items might share the same actions (e.g., for the PISA 2012 dataset, the actions of starting or ending an item were the same across some items), which confounds the unique contribution of actions to item responses. Nevertheless, we noticed that there exist some studies analyzing action sequences to inform how learners attempt problems. For example, a recent study proposed to use sequence-to-sequence autoencoders to extract informative latent variables from learners' action sequences in solving a problem (Tang et al., 2019). Their study demonstrates the possibility of using process data to decipher how learners attempt a problem. Instead, as we mentioned earlier, the process data learning of LogCF is more of a regularization technique for estimating item- and learner-skill associations in our study.

Although we tried to demonstrate the effectiveness of LogCF for learning or refining the expert-specified item-skill associations, the variant of LogCF without expert information was not superior to the variants with expert information. However, given that random-int achieved almost as good prediction performance as expert-Q and expert-Q-int, LogCF could be used to reduce the cost of pre-specifying a Q-matrix by domain experts. In addition, the interpretability of estimated item-skill associations by LogCF is limited. Although we used regularization to improve the interpretability, the estimated item-skill associations are not in a binary or categorical scale which might not be well accepted by educational practitioners. In our study, the interpretability is elucidated in reference to IRT item parameters, which is not intuitive enough for those who are not familiar with psychometric measurement. Moreover, refinement and estimation of item-skill associations are validated by predictive generalization in our study, which should be further examined by other means.

Furthermore, LogCF is developed for binary item responses. In many educational settings, non-binary scoring is used more often. How LogCF is capable of modeling non-binary scores should be considered in future investigations.

Lastly, like some recommendation system algorithms, after training the model, LogCF works well only for trained learners and items, which bears a cold-start problem for new learners and users. This issue, however, can be addressed by embedding learners and items with auxiliary information. For example, learners' learning styles and demographic information can be incorporated in learner embedding, and item texts can be incorporated in item embedding. As such, features of new learners and new items can be directly learned and evaluated by the model, which helps in mitigating the cold-start problem.

7. Contributions

In this work, to develop a system for modeling and predicting learners' learning outcomes as well as providing information on learner and item latent factors, we devised a general framework based on deep CF with process data to model learner-item interactions. This framework involves two main components which are a deep CF architecture and a deep learning-based architecture exploiting learners' actions and time in problem solving. To demonstrate the usefulness of our approach, we compared the effectiveness in missing response prediction between our approach and conventional matrix factorization and psychometric measurement model with the data of an international large-scale complex problem-solving assessment. The experimental results with the real-world dataset validated the effectiveness and interpretability of LogCF.

Based on the experimental results, we consider our framework as a contribution to practice in the following ways. First, as we mentioned earlier, despite the existing machine learning-based approaches for learning outcome modeling, we argue the importance of incorporating learner process data in learning outcome modeling. As our results suggest, the variants of LogCF outperform NeuralCF (i.e., the model without process data learning). This indicates that process data learning helps in refining training and prediction. For example, by modeling learners' problem-solving processes, our framework might be more capable of differentiating a correct response based on a full understanding of the latent skills, partial understanding of the latent skills, or guessing, which improves the accuracy of the learner-skill and item-skill association estimation. This feature of LogCF is especially beneficial for personalized learning. For example, in intelligent tutoring systems, it is often the case that a feedback or hint is given when learners give incorrect responses while solving a problem (Psotka et al., 1988). However, the same incorrect responses might have very different underlying problem-solving processes, which affect the diagnosis of learners' mastery of latent skills. In this sense, our system however is more efficient and accurate for cognitive diagnosis by exploiting the process data. Second, LogCF is capable of learning item-skill associations from scratch which are not worse than the expert-specified ones. Given the results on the comparison between the three variants of LogCF, LogCF performs well regardless of whether the expert-specified item-skill associations are available or not. In some scenarios where numerous items are automatically generated by machine for computer-based assessments, experts' efforts in specifying how strongly each item measures each latent skill might not be required given that item-skill associations can be automatically learned by the model, which benefits the development of large-scale assessments.

Finally, given our implementation of LogCF with two datasets from both perspectives of educational data mining and psychometric measurement, it is evident that LogCF is a generic and flexible framework that can be applied in various educational or even non-educational settings. For example, in the area of digital game-based assessments, evidence modeling is an ongoing research topic and a variety of approaches have been proposed to connect performance indicators to targeted skills in previous studies (de Klerk et al., 2015). However, the majority of previous studies mainly focused on learners' explicit performance indicators, and using learners' process data from digital game-based assessments in evidence modeling is an emerging trend (Min et al., 2020). Our approach could be used for evidence modeling with learners' process data in digital game-based assessments. Moreover, in the context of online education (e.g., massive open online courses), it is crucial to recommend tailored learning tasks or learning content to learners. Our approach is capable of modeling learners' performance in past learning opportunities to predict their performance in future ones, which facilitates individualized learning path recommendations with their estimated mastery levels of latent skills and learning outcomes.

Effective learning is shaped by numerous contextual factors such as learners' task and cognitive conditions (e.g., learning resources, interest, and motivation), cognitive processes and products (e.g., behaviors and performance in learning tasks), and internal or external feedback and standards (Winne, 2005). As such, in terms of future work, more explorations on exploiting learner and item features and learning contextual information for learning outcome modeling is needed. For example, our framework can be extended by incorporating learners' motivation, learning styles and other cognitive and emotional measures, and the text, audio, and video information of items, to improve the accuracy of learning outcome prediction.

References

ALMUTAIRI, F. M., SIDIROPOULOS, N. D., AND KARYPIS, G. 2017. Context-aware recommendation-based learning analytics using tensor and coupled matrix factorization. *IEEE Journal of Selected Topics in Signal Processing 11,* 5, 729–741.

BAKER, R. S. AND YACEF, K. 2009. The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining 1,* 1, 3–17.

BERGNER, Y., DRÖSCHLER, S., KORTEMEYER, G., RAYYAN, S., SEATON, D. T., AND PRITCHARD,

D. E. 2012. Model-based collaborative filtering analysis of student response data: Machine-learning item response theory. In *Proceedings of the 5th International Conference on Educational Data Mining*, K. Yacef, O. R. Zaïane, A. Hershkovitz, M. Yudelson, and J. C. Stamper, Eds. International Educational Data Mining Society, 95–102.

BOTELHO, A. F., BAKER, R. S., AND HEFFERNAN, N. T. 2017. Improving sensor-free affect detection using deep learning. In *Proceedings of the 18th International Conference on Artificial Intelligence in Education*, E. André, R. Baker, X. Hu, M. M. T. Rodrigo, and B. du Boulay, Eds. Springer International Publishing, Cham, 40–51.

CEN, H., KOEDINGER, K., AND JUNKER, B. 2005. Automating cognitive model improvement by A* search and logistic regression. In *Proceedings of the Workshop on Educational Data Mining at the Twentieth National Conference on Artificial Intelligence*. 47–53.

CEN, H., KOEDINGER, K., AND JUNKER, B. 2006. Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, M. Ikeda, K. D. Ashley, and T.-W. Chan, Eds. Springer, Berlin, Heidelberg, 164–175.

CHAPLOT, D. S., MACLELLAN, C., SALAKHUTDINOV, R., AND KOEDINGER, K. 2018. Learning cognitive models using neural networks. In *Proceedings of the 19th International Conference on Artificial Intelligence in Education*, C. Penstein Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, and B. du Boulay, Eds. Springer International Publishing, Cham, 43–56.

CHEN, Y., LI, X., LIU, J., AND YING, Z. 2019. Statistical analysis of complex problem-solving process data: An event history analysis approach. *Frontiers in Psychology 10*, 486.

CHENG, S., LIU, Q., CHEN, E., HUANG, Z., HUANG, Z., CHEN, Y., MA, H., AND HU, G. 2019. DIRT: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, New York, 2397–2400.

CHIU, C.-Y. 2013. Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement 37,* 8, 598–618.

CHOLLET, F. ET AL. 2015. Keras. https://keras.io.

CORBETT, A. T. AND ANDERSON, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4,* 4, 253–278.

DACREMA, M. F., CREMONESI, P., AND JANNACH, D. 2019. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 101–109.

DE KLERK, S., VELDKAMP, B. P., AND EGGEN, T. J. 2015. Psychometric analysis of the performance data of simulation-based assessment: A systematic review and a bayesian network example. *Computers & Education 85*, 23–34.

DESMARAIS, M. C. 2011. Conditions for effectively deriving a Q-matrix from data with non-negative matrix factorization. In *Proceedings of the 4th International Conference on Educational Data Mining*, M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, and J. C. Stamper, Eds. International Educational Data Mining Society, 41–50.

DESMARAIS, M. C. 2012. Mapping question items to skills with non-negative matrix factorization. *SIGKDD Explorations Newsletter 13,* 2 (May), 30–36.

DESMARAIS, M. C. AND NACEUR, R. 2013. A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In *Proceedings of the 16th International Conference on*

Artificial Intelligence in Education, H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, Eds. Springer, Berlin, Heidelberg, 441–450.

DING, C., LI, T., PENG, W., AND PARK, H. 2006. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 126–135.

DURAND, G., BELACEL, N., AND GOUTTE, C. 2015. Evaluation of expert-based q-matrices predictive quality in matrix factorization models. In *Proceedings of the 10th European Conference on Technology Enhanced Learning*, G. Conole, T. Klobučar, C. Rensing, J. Konert, and E. Lavoué, Eds. Springer International Publishing, Cham, 56–69.

ELKAHKY, A. M., SONG, Y., AND HE, X. 2015. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 278–288.

EMBRETSON, S. E. AND REISE, S. P. 2000. *Item Response Theory for Psychologists*. Lawrence Erlbaum Associates, Mahwah, New Jersey.

GLOROT, X. AND BENGIO, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Y. W. Teh and M. Titterington, Eds. Proceedings of Machine Learning Research, vol. 9. 249–256.

GREIFF, S., WÜSTENBERG, S., AND AVVISATI, F. 2015. Computer-generated log-file analyses as a window into students' minds? A showcase study based on the PISA 2012 assessment of problem solving. *Computers & Education 91*, 92–105.

HANSEN, M., CAI, L., MONROE, S., AND LI, Z. 2016. Limited-information goodness-of-fit testing of diagnostic classification item response models. *British Journal of Mathematical and Statistical Psychology 69,* 3, 225–252.

HE, X., LIAO, L., ZHANG, H., NIE, L., HU, X., AND CHUA, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 173–182.

HOCHREITER, S. AND SCHMIDHUBER, J. 1997. Long short-term memory. *Neural Computation 9,* 8, 1735–1780.

HOYER, P. O. 2004. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research 5,* Nov, 1457–1469.

KINGMA, D. P. AND BA, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

KOEDINGER, K. R., BAKER, R. S., CUNNINGHAM, K., SKOGSHOLM, A., LEBER, B., AND STAMPER, J. 2010. A data repository for the EDM community: The PSLC DataShop. In *Handbook of Educational Data Mining*, C. Romero, S. Ventura, M. Pechenizkiy, and R. Baker, Eds. CRC Press, Boca Raton, FL.

LAN, A. S., WATERS, A. E., STUDER, C., AND BARANIUK, R. G. 2014. Sparse factor analysis for learning and content analytics. *The Journal of Machine Learning Research 15,* 1, 1959–2008.

LEE, D. D. AND SEUNG, H. S. 2001. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*. 556–562.

LI, S., KAWALE, J., AND FU, Y. 2015. Deep collaborative filtering via marginalized denoising auto-encoder. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 811–820.

LINDEN, G., SMITH, B., AND YORK, J. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 1, 76–80.

LINDSEY, R. V., KHAJAH, M., AND MOZER, M. C. 2014. Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in Neural Information Processing Systems*. 1386–1394.

LING, C. X., HUANG, J., AND ZHANG, H. 2003. AUC: A better measure than accuracy in comparing learning algorithms. In *Conference of the Canadian Society for Computational Studies of Intelligence*. Springer, 329–341.

LIU, H., LIU, Y., AND LI, M. 2018. Analysis of process data of PISA 2012 computer-based problem solving: Application of the modified multilevel mixture irt model. *Frontiers in Psychology 9*, 1372.

LIU, J., XU, G., AND YING, Z. 2012. Data-driven learning of Q-matrix. *Applied Psychological Measurement 36,* 7, 548–564.

LIU, R., DAVENPORT, J. L., AND STAMPER, J. C. 2016. Beyond log files: Using multi-modal data streams towards data-driven KC model improvement. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. International Educational Data Mining Society, 436–441.

LIU, R., STAMPER, J., DAVENPORT, J., CROSSLEY, S., MCNAMARA, D., NZINGA, K., AND SHERIN, B. 2019. Learning linkages: Integrating data streams of multiple modalities and timescales. *Journal of Computer Assisted Learning 35,* 1, 99–109.

LIU, Y., TIAN, W., AND XIN, T. 2016. An application of M2 statistic to evaluate the fit of cognitive diagnostic models. *Journal of Educational and Behavioral Statistics 41,* 1, 3–26.

MATSUDA, N., FURUKAWA, T., BIER, N., AND FALOUTSOS, C. 2015. Machine beats experts: Automatic discovery of skill models for data-driven online courseware refinement. In *Proceedings of the 8th International Conference on Educational Data Mining*, O. C. Santos, J. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. M. Luna, M. C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, and M. C. Desmarais, Eds. International Educational Data Mining Society, 101–108.

MIN, W., FRANKOSKY, M. H., MOTT, B. W., ROWE, J. P., SMITH, A., WIEBE, E., BOYER, K. E., AND LESTER, J. C. 2020. DeepStealth: Game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies 13,* 2, 312–325.

NGUYEN, D. M., TSILIGIANNI, E., AND DELIGIANNIS, N. 2018. Extendable neural matrix completion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6328–6332.

OECD. 2014a. *PISA 2012 Results: Creative Problem Solving: Students' Skills in Tackling Real-Life Problems (Volume V)*. OECD Publishing.

OECD. 2014b. *PISA 2012 Technical Report*. OECD Publishing.

PARDOS, Z. A. AND DADU, A. 2018. dAFM: Fusing psychometric and connectionist modeling for Q-matrix refinement. *Journal of Educational Data Mining 10,* 2, 1–27.

PIECH, C., BASSEN, J., HUANG, J., GANGULI, S., SAHAMI, M., GUIBAS, L. J., AND SOHL-DICKSTEIN, J. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*. 505–513.

PSOTKA, J., MASSEY, L. D., AND MUTTER, S. A. 1988. *Intelligent tutoring systems: Lessons learned.* Psychology Press.

RASCH, G. 1980. *Probabilistic models for some intelligence and attainment tests*. University of Chicago Press, Chicago.

RUPP, A. A., TEMPLIN, J. L., AND HENSON, R. A. 2010. *Diagnostic Measurement: Theory, Methods, and Applications*. Guilford Press, New York.

SAHEBI, S., LIN, Y., AND BRUSILOVSKY, P. 2016. Tensor factorization for student modeling and performance prediction in unstructured domain. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. International Educational Data Mining Society, 502–506.

SARWAR, B., KARYPIS, G., KONSTAN, J., AND RIEDL, J. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. ACM, New York, 285–295.

SHU, Z., BERGNER, Y., ZHU, M., HAO, J., AND VON DAVIER, A. A. 2017. An item response theory analysis of problem-solving processes in scenario-based tasks. *Psychological Test and Assessment Modeling 59,* 1, 109–131.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., AND SALAKHUTDINOV, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15,* 1, 1929–1958.

SU, X. AND KHOSHGOFTAAR, T. M. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence 2009*, 1–19.

SUN, Y., YE, S., INOUE, S., AND SUN, Y. 2014. Alternating recursive method for q-matrix learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, J. C. Stamper, Z. A. Pardos, M. Mavrikis, and B. M. McLaren, Eds. International Educational Data Mining Society, 14–20.

TANG, X., WANG, Z., LIU, J., AND YING, Z. 2019. An exploratory analysis of the latent structure of process data via action sequence autoencoder. *arXiv preprint arXiv:1908.06075*.

TATSUOKA, K. K. 2009. *Cognitive Assessment: An Introduction to the Rule Space Method*. Multivariate Applications Series. Routledge, New York.

THAI-NGHE, N., DRUMOND, L., HORVÁTH, T., KROHN-GRIMBERGHE, A., NANOPOULOS, A., AND SCHMIDT-THIEME, L. 2012. Factorization techniques for predicting student performance. In *Educational Recommender Systems and Technologies: Practices and Challenges*, O. C. Santos and J. G. Boticario, Eds. IGI Global, 129–153.

VAN DEN OORD, A., DIELEMAN, S., AND SCHRAUWEN, B. 2013. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*. 2643–2651.

VAN DER LINDEN, W. J. AND GLAS, C. A. 2000. *Computerized Adaptive Testing: Theory and Practice*. Kluwer, New York.

VIE, J.-J. AND KASHIMA, H. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 750–757.

WANG, H., WANG, N., AND YEUNG, D.-Y. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 1235–1244.

WANG, L., SY, A., LIU, L., AND PIECH, C. 2017. Deep knowledge tracing on programming exercises. In *Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale*. 201–204.

WILLMOTT, C. J. AND MATSUURA, K. 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research 30,* 1, 79–82.

WINNE, P. H. 2005. A perspective on state-of-the-art research on self-regulated learning. *Instructional Science 33,* 5/6, 559–565.

YAO, L. AND BOUGHTON, K. A. 2007. A multidimensional item response modeling approach for improving subscale proficiency estimation and classification. *Applied Psychological Measurement 31,* 2, 83–105.

YEUNG, C.-K. 2019. Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. *arXiv preprint arXiv:1904.11738*.

ZHANG, F., YUAN, N. J., LIAN, D., XIE, X., AND MA, W.-Y. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 353–362.