Deep Collaborative Filtering for Enhanced Learning Outcome Modeling

by

Fu Chen

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Measurement, Evaluation, and Data Science

Department of Educational Psychology
University of Alberta

Abstract

The digital learning and assessment movement has contributed to an explosion of structured and unstructured learner data (e.g., learner problem-solving product and process data). This calls for new developments in large-scale learning outcome modeling to optimally address the variety, volume, uncertainty, and velocity of big data in education. Existing learning outcome modeling techniques, such as psychometric measurement models and Bayesian models, typically require structured product data and fail to account for process data. Moreover, most of them are incapable of learning associations between items and latent skills. Leveraging the advantages of collaborative filtering (CF) used in recommender systems, this study proposes three novel deep learning-based CF approaches — SDCF, LogCF, and LogSDCF — to model both product and process data for enhanced learning outcome modeling. The three models are also capable of discovering item-skill associations from the data without expert information. Specifically, SDCF is developed to model product data sequentially by predicting learners' next item responses based on their history of item responses; LogCF is proposed to model both product and process data to predict learners' missing or future item responses when item responses are not in a sequential form; LogSDCF is devised to model both product and process data to predict learners' future item responses based on their response history when items are presented in a sequential form. To evaluate the effectiveness of the proposed approaches, the three models were compared with conventional learning outcome modeling approaches using both simulated and real-world datasets. Results showed that all three approaches achieved higher prediction accuracy of learners' future or missing item responses than their baselines. Moreover, the proposed approaches were found to be promising in discovering item-skill associations without expert input.

Acknowledgement

Too many people helped me in my Ph.D. journey. First of all, I would like to express my heartfelt gratitude to my supervisor, Dr. Ying Cui, for her consistent support, encouragement, and patience during my Ph.D. studies. My dissertation would not have been successful without her guidance. In addition, I also want to thank my supervisory committee members, Drs. Maria Cutumisu and Okan Bulut, for their valuable insights and feedback on my work. They are always helpful whenever I have questions. I am fortunate to be a CRAMEr as all professors were so professional, friendly, and supportive.

I also need to thank all my buddies, friends, and coworkers for their support during the COVID-19 pandemic. Thank you, Zac, Toffee, Twinkle, and Sparkle, for always being with me. My life has become much easier and more enjoyable because of you. Also, I'd like to thank Chang and Jiaying. We studied together in many courses and supported each other through difficult times. I also appreciate Qi's generous help when I had difficulties in life and academics.

Finally, I cannot forget to thank my family for their unconditional love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

The rising tide of big data in the 21st century catalyzes extensive developments and changes in a wide range of fields (e.g., marketing, business, and health). Despite the effective use of big data in other sectors, explorations of big data in education are still relatively rare and mostly rudimentary (Macfadyen et al., 2014). To date, advanced by the rapid evolution in information and communication technologies, the integration of big data and adaptive learning systems has given rise to a growing movement of personalized learning in a variety of education sectors (e.g., K-12 education, Roberts-Mahoney et al., 2016; higher education, Kong & Song, 2015; online courses, Daniel et al., 2015). Notably, the advances in digital learning (e.g., mobilization of learning technologies) hold the promise that learners all over the world will have easy and affordable access to personalized learning experience in the near future.

In a personalized learning environment, customized learning plans are typically created for learners based on what they know, what they lack, and how they learn best. Obviously, the success of personalized learning relies heavily on the availability of and access to a big amount of data on individuals' learning behaviors. Specifically, through mining learner data and modeling learning behaviors, personalized learning systems are capable of continuously monitoring and assessing individuals' learning progress based on their interactions with learning resources (e.g., learning materials and assessment questions). Consequently, learners can be informed of their strengths and weaknesses and provided with timely feedback and remediation. Therefore, it is desirable that novel approaches can be developed to better elicit, evaluate, and collect individuals' learning behaviors. Nowadays, computer-based assessment (CBA) for learning has been increasingly used to evaluate learners' learning outcomes and provide customized feedback across a wide range of learning contexts (Shute & Rahimi, 2017). The popularity and effectiveness of CBAs in personalized

learning are attributable to their capacities to evaluate higher-level learner competencies and their flexibility in assessment administration. Moreover, from the data perspective, compared with standardized paper-pencil assessments, CBAs can elicit and collect much more information about how learners perform on and solve each learning task. This enables education practitioners to better evaluate and validate an assessment, and to provide learners with finer-grained feedback. Therefore, in this dissertation, the methodological developments are situated in the context of CBA for learning.

Learning Outcome Modeling for Digital Learning

Accurate evaluation of learning outcomes plays a pivotal role in an effective CBA. In this regard, several learning outcome modeling techniques were developed for earlier CBAs (e.g., intelligent tutoring systems, Psotka et al., 1988; computerized adaptive testing, van der Linden & Glas, 2000) in both domains of educational data mining and psychometrics. Some representative approaches include Bayesian Knowledge Tracing (BKT; Corbett & Anderson, 1994), Item Response Theory (IRT; Lord, 1952), and Cognitive Diagnosis Models (CDM; Tatsuoka, 1990). Earlier learning outcome modeling approaches, however, are not designed to effectively utilize educational big data in the digital era. For example, BKT and some psychometric models such as cognitive diagnosis typically require extensive human efforts to pre-define learning rules (e.g., which items/learning opportunities measure which skills), and they are of limited scalability and efficacy in handling great amounts of learners and items, let alone a larger number of missing responses.

In recent years, leveraging machine learning advances, a growing body of research in educational data mining developed numerous novel approaches with high scalability and predictive capacity for learning outcome modeling, which greatly benefit the large-scale application of CBAs (e.g., Baker & Yacef, 2009; Bergner et al., 2012; Cheng et al., 2019; Lan et al., 2014). Most of these approaches were proposed for two purposes: learning

analytics and content analytics (Lan et al., 2014). Learning analytics refers to "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" (Long et al., 2011). Content analytics, however, focuses more on methodologies. It was defined as "automated methods for examining, evaluating, indexing, filtering, recommending, and visualizing different forms of digital learning content" (Kovanović et al., 2017). In terms of learning analytics, the learning outcome modeling approaches aim to estimate or infer the degree to which learners master the knowledge or latent skills, which is called learner modeling in the literature; in terms of content analytics, a growing number of learning outcome modeling approaches focus on optimizing the organization of learning content, such as learning materials, assessment tasks and hints, for improved learning outcomes. Notably, the two purposes can be approached within a single modeling framework (e.g., Lan et al., 2014).

For content analytics, an imperative issue to address in both communities of educational data mining and psychometrics is domain modeling (i.e., discovering item-skill associations). Extensive efforts in the literature were devoted to developing models that learn from scratch or that refine item-skill associations in comparison with expert-specified ones. For example, in psychometrics, CDMs require careful consideration of item-skill associations, or the Q-matrix, by domain experts, given that a mis-specified Q-matrix often results in poor model-data fit and, consequently, in undermined model classification accuracy (Hansen et al., 2016; Liu et al., 2016). Therefore, an increasing number of approaches were developed to refine or estimate Q-matrices without strong involvement of human knowledge for cognitive diagnosis in psychometrics (e.g., Chiu, 2013; Liu et al., 2012). In educational data mining, estimating or refining item-skill associations is particularly important in recent years. Approaches for discovering item-skill associations in educational data mining typically

rely on advances of machine learning techniques (e.g., Chaplot et al., 2018; Desmarais, 2012; Desmarais & Naceur, 2013; Lindsey et al., 2014; Sun et al., 2014). Compared with psychometric approaches, machine learning-based approaches are more capable of handling unstructured and incomplete item response data and modeling great amounts of items and learners in large-scale settings.

Collaborative Filtering for Learning Outcome Modeling

Among numerous machine learning advances, the approach of collaborative filtering (CF) is suitable to both learner modeling and domain modeling under the condition of incomplete item response matrices. CF is a pivotal method for recommender systems (Linden et al., 2003; Sarwar et al., 2001) that is used to recommend new items (e.g., movies, books, shopping items) to users based on their histories of item clicks or item ratings (Su & Khoshgoftaar, 2009).

Concretely, CF approaches often assume that item-user interactions are affected by a set of latent factors. They use present item-user interactions to estimate the latent factors, which are in turn used to make probabilistic predictions on future interactions. This idea of CF for recommender systems applies to learning outcome modeling. In digital learning, it is desirable to recommend learners tailored learning materials and associated assessment items within the zone of their proximal development (Campione et al., 1984). Typically, recommendations of learning materials and assessment items are based on learners' learning profiles. Learners' past learning performance measured by the CBA is analogous to users' past preferences or taste information in CF, and the unassigned questions or assessment tasks for learners are analogous to the new items to be recommended in recommender systems. In this sense, CF approaches should be applicable to learning outcome modeling in terms of predicting future or unseen item responses.

A commonly used type of model-based CF is matrix factorization, which is popularized by the Netflix Prize[1]. For an incomplete user-item interaction matrix with missing information, matrix factorization decomposes it into the product of two or more lower-dimensionality matrices, with a rank as the number of latent factors. As such, with some model constraints, entries of the lower dimensionality matrices might resemble user- and item-factor associations, which are analogous to the learner- and item-skill associations in learning analytics and content analytics. In this sense, how CF, especially matrix factorization, works for prediction conforms with the purpose of learner modeling and domain modeling. With learner data, learner- and item-skill associations estimated by CF quantify the degree to which learners acquire and items measure the latent skills. However, conventional CF approaches such as matrix factorization are limited in capturing a high degree of complexity of learner-item interactions because probabilistic predictions by CF are often based on linear combinations (e.g., inner-products) of learner- and item-skill associations. Therefore, in recent years, to capture the complexity of learner-item interactions, some work proposed to incorporate deep learning architectures into CF to improve the model intricacy or exploit the auxiliary information for modeling (e.g., Elkahky et al., 2015; He et al., 2017; van den Oord et al., 2013; Wang et al., 2015; Zhang et al., 2016).

Potential of Process Data Modeling

In the digital learning context, learner data can take the forms of both product data and process data. Product data refers to learners' final work products in learning such as responses to assessment questions. Process data, or log file data, records the information regarding how learners produce the final work products, which are typically stored as log file entries (Rupp et al., 2012). In recent years, how to successfully use process data to profile and facilitate learning is an emerging research topic in both the domains of educational data

[1] https://www.netflixprize.com/

mining and psychometric measurement. Process data, unlike explicit product data, reveal a wealth of information regarding learners' interactions with the system, which can be used to uncover their learning or problem-solving processes. However, learning outcome modeling with process data has not been well addressed by conventional models in educational data mining (e.g., BKT) or psychometric measurement (e.g., IRT and cognitive diagnostic models). Nevertheless, some pioneering studies demonstrated that learner process data can be successfully used to reveal learners' problem-solving strategies (Greiff et al., 2015), evaluate learners' latent skills (Liu et al., 2018), and predict learners' learning outcomes (Chen et al., 2019). Although these approaches were mostly not generic and their applications were often of a small scale, they have cast light on the potential of process data modeling for interpreting and predicting learning outcomes.

Terminology

Prior to outlining the research objective, the key terms used in the dissertation are defined.

Learner. A learner refers to a person who is involved in and interacts with the digital learning system, such as students who use an intelligent tutoring system or a learning system, examinees who participate in educational assessments, or respondents who take a psychological scale or questionnaire.

Item. An item refers to the object learners interact with in the digital learning system. In should be noted that items can be defined at different levels. For example, for a mathematics test of multiple-choice questions, each question constitutes an item. However, for a complex simulated problem, learners might take a set of problem-solving steps to provide a final solution. If each of the problem-solving steps can be scored, each step should be considered as an item. Moreover, another related term used in educational data mining is "transaction", which refers to learners' interactions with the system in each problem-solving

step. In this case, transactions can be considered as actions for attempting a step, indicative of their problem-solving processes.

Skill. The skill refers to what the assessment measures, which is typically unobservable. In the educational data mining community, a skill is often called a "knowledge component", whereas in the psychometrics community, it might refer to dimensions or attributes of a higher-order concept.

Learner modeling. Learner modeling is a critical feature of digital learning systems. It refers to "inferring, from the learner's problem-solving actions and answers, what is likely well understood or mastered, and what is not, from only a few observations, and to move on in the curriculum at the right pace for that specific learner" (Desmarais & Baker, 2012). In the literature, learner modelling often has the same meaning as student modeling, skill modeling, or knowledge modeling.

Domain modeling. Domain modeling refers to "the assignment of individual items to skills and the modeling of relations among skills" (Pelánek, 2017). The dissertation focuses on the first component of domain modeling, which models item-skill associations that can be represented as both categorical and continuous values. For example, item-skill associations can be of two classes, "yes" or "no", which indicate whether a skill is measured by an item; they can also be real-valued, which indicates the degree to which an item measures a skill. In the former case, item-skill associations have the same meaning as those in the Q-matrix in psychometrics.

Learning outcome modeling. In the dissertation, learning outcome modeling denotes a general term encompassing both learner modeling and domain modeling. More specifically, it refers to making probabilistic predictions of learners' item responses as well as discovering item-skill associations based on the learner performance data. The "learning outcome"

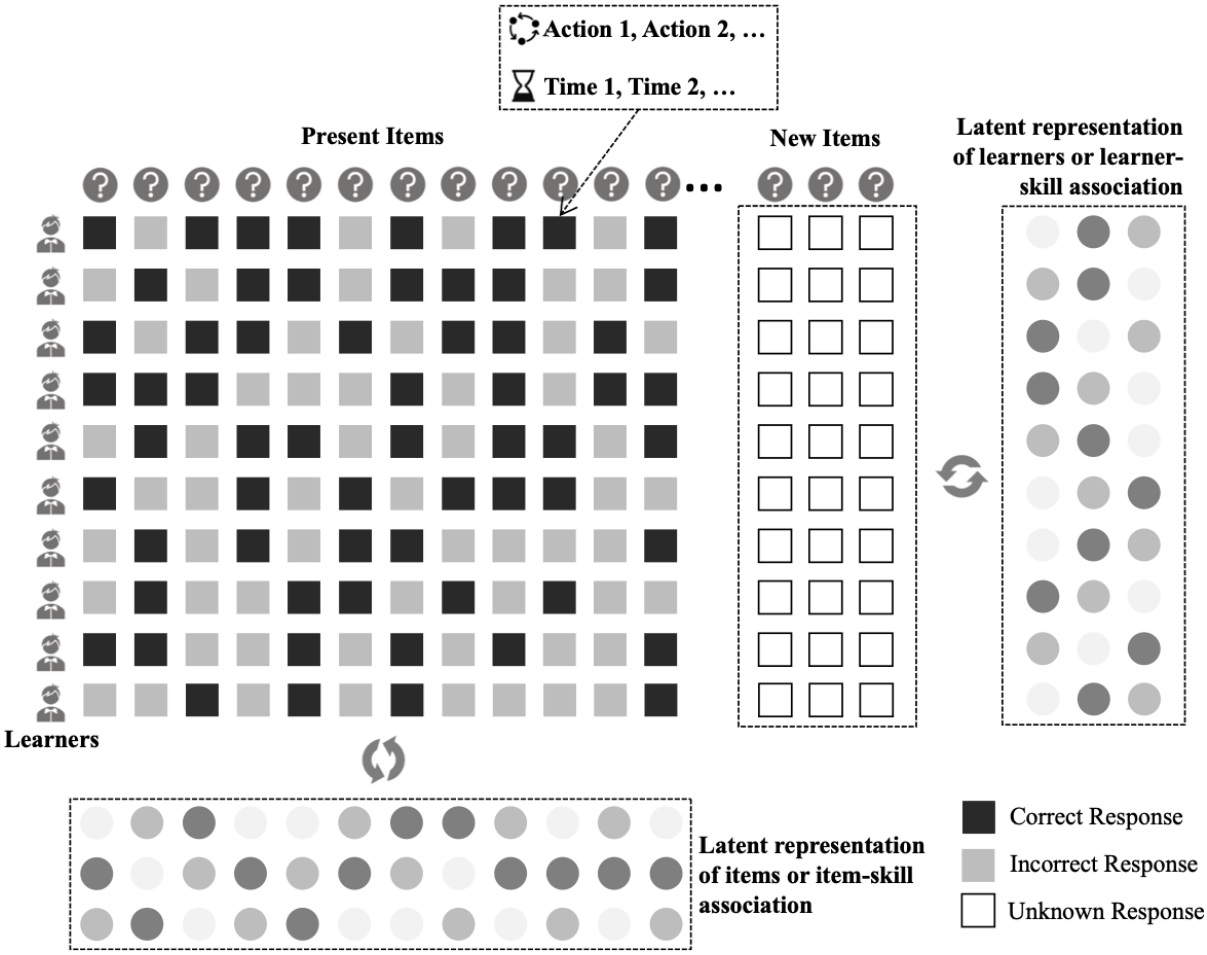
includes both learners' explicit work products such as item responses and their in-process problem-solving actions and time information logged by the system.

Research Objective

As summarized in the previous sections, there are two types of learner performance data associated with digital learning: product data and process data. A vast majority of existing methods in educational data mining and psychometrics were developed to exploit product data for learning outcome modeling, and much fewer methods account for process data. The overarching objective of this dissertation is to investigate how to enhance learning outcome modeling with the presence of process data based on the deep CF framework. A graphical intuition is presented in Figure 1 to illustrate the problem of learning outcome modeling in the dissertation. The graph shows that several learners have correct or incorrect responses on a number of items, which constitutes an item response matrix (i.e., the product data). Moreover, for each observed item response, there are a set of problem-solving actions and associated time durations, which constitute the process data. The product and the process data serve as the input for learning outcome modeling. For each learner, learning outcome modeling estimates his or her probabilities of getting new items (in)correct (the part of new items shown by the dotted rectangle), which are the output of the model.

Figure 1

Schematic Illustration of Learning Outcome Modeling



Note. Black squares indicate correct responses, grey squares indicate incorrect responses, and empty boxes indicate unknown responses to be estimated. The latent factors estimated by CF are indicated by circles of different colors. All content within dotted rectangles is not directly observable through the item response matrix.

In addition to predicting new item responses, the model also estimates latent representations of learners and items. With appropriate model regularization, they can be used to indicate the associations between learners and skills, and items and skills, which represent the degree to which learners master the skills and items measure the skills, respectively. Despite the graph depicting the input and output of the model, a well-designed model architecture is needed to make the model learnable and predictable.

Moreover, the item response matrix includes temporal information regarding the sequence of item responses. Addressing the temporal dependencies between item responses in learning outcome modeling takes advantage of more information on learning progress, and it might contribute to a more predictive model. Given the above considerations, the dissertation proposes three approaches to address the aforementioned research objective: 1) a CF-based sequential modeling framework solely based on learners' product data, 2) a CF-based framework based on both product and process data, and 3) a CF-based framework integrating the first two approaches. More details regarding each approach will be provided in the following chapters.

Dissertation Organization

The rest of the dissertation is organized as follows.

Chapter 2 first reviews the context of learning outcome modeling in terms of computer-based assessment for learning. Subsequently, it reviews a wide range of existing models and approaches for learning outcome modeling. Moreover, the pioneering work on process data analysis in education is also covered in Chapter 2. Based on the identified research gaps, Chapter 2 ends with research questions and corresponding approaches.

Chapter 3 focuses on the first research problem and elaborates the technical details of the proposed approach, SDCF. It first introduces the preliminaries on sequential modeling, especially for BKT and DKT. Next, it formulates the research problem followed by the general framework of SDCF with details on model architecture, prediction, and model learning. Then it presents experiments based on both simulated and real-world datasets to validate the prediction capacity of SDCF. The section on experiments covers dataset description, training set-up, hyperparameter tuning, and baseline models. Finally, the results of SDCF are presented in Chapter 3.

Chapter 4 addresses the second research problem and demonstrates the development of LogCF. It starts with an introduction to the preliminary of matrix factorization, followed by problem formulation and details on the general framework (e.g., model architecture, prediction, and model learning). Next, the chapter presents experiments with two different types of real-world datasets. Dataset description, training set-up, hyperparameter tuning, and baseline models are presented in detail. Finally, the chapter presents the experimental results of the model.

Chapter 5 focuses on the third research problem and demonstrates the development of LogSDCF. Similar to the previous two chapters, it presents details on problem formulation, the general framework, experiments with a real-world dataset, and results.

In Chapter 6, a general discussion on limitations, contributions, implications, and future directions of the proposed approaches is presented.

Chapter 2 Related Work

This chapter starts with a brief introduction to the context of learning outcome modeling in the digital era, computer-based learning and assessments, with an emphasis on the importance of learning outcome modeling for personalized learning and formative assessments and the potential of process data analysis. Subsequently, the chapter presents a comprehensive survey of existing mainstream approaches for learning outcome modeling in a variety of fields, followed by an overview section summarizing these approaches. Then, a brief review of pioneering work on process data analysis is presented to reveal the potential of process data for learning outcome modeling. Finally, the chapter ends with research problems based on the gaps in previous work and proposes corresponding approaches.

Computer-Based Assessment for Learning

Benefits of Computer-Based Assessments for Formative Evaluation

Assessment is a fundamental element of teaching and learning. It is an important way to collect learner information and measure learners' performance and understanding with respect to learning goals. Generally, we distinguish two types of assessments: summative assessments (i.e., assessments *of* learning) and formative assessments (i.e., assessments *for* learning; Black & Wiliam, 2009). The former refers to evaluating individuals' learning outcomes at the end of a teaching and learning unit in contrast to an established standard or benchmark. The latter refers to ongoing evaluations of learners' progress with prescriptive feedback for improving teaching and learning. Summative and formative assessments are developed to address different educational purposes. For example, traditional school learning is often evaluated with standards-based summative assessments for accountability purposes (e.g., final or midterm exams, standardized tests for admissions, and standardized assessments for informing educational policy). The results of summative assessments can be used to inform learners if they have achieved educational goals, to make comparison between

different populations, to promote the accountability at different educational levels, and to inform educational policy (Shute & Rahimi, 2017). However, in learner-centered scenarios, formative assessments are used frequently as a supportive approach. With formative assessments, instructors can evaluate individuals' learning progress in a timely manner while learners receive individualized instruction and feedback to improve their learning outcomes. Compared with summative assessments, formative assessments demonstrate greater potential in supporting learning and they are successfully used for different audiences across various content domains and educational sectors (e.g., Davies & Ecclestone, 2008; Gikandi et al., 2011; Meek et al., 2017; Shute et al., 2008; Tsai et al., 2015). Moreover, it was found that learners, especially struggling learners, instructed with formative assessments are more likely to increase their academic performance than those instructed with standard pedagogical approaches (e.g., Carrillo-de-la-Pena et al., 2009; Kleitman & Costa, 2014; López-Pastor et al., 2013; Pastor, 2011). In summary, formative assessment, or assessment *for* learning, is playing an increasingly important role in education given its advantages in supporting learning.

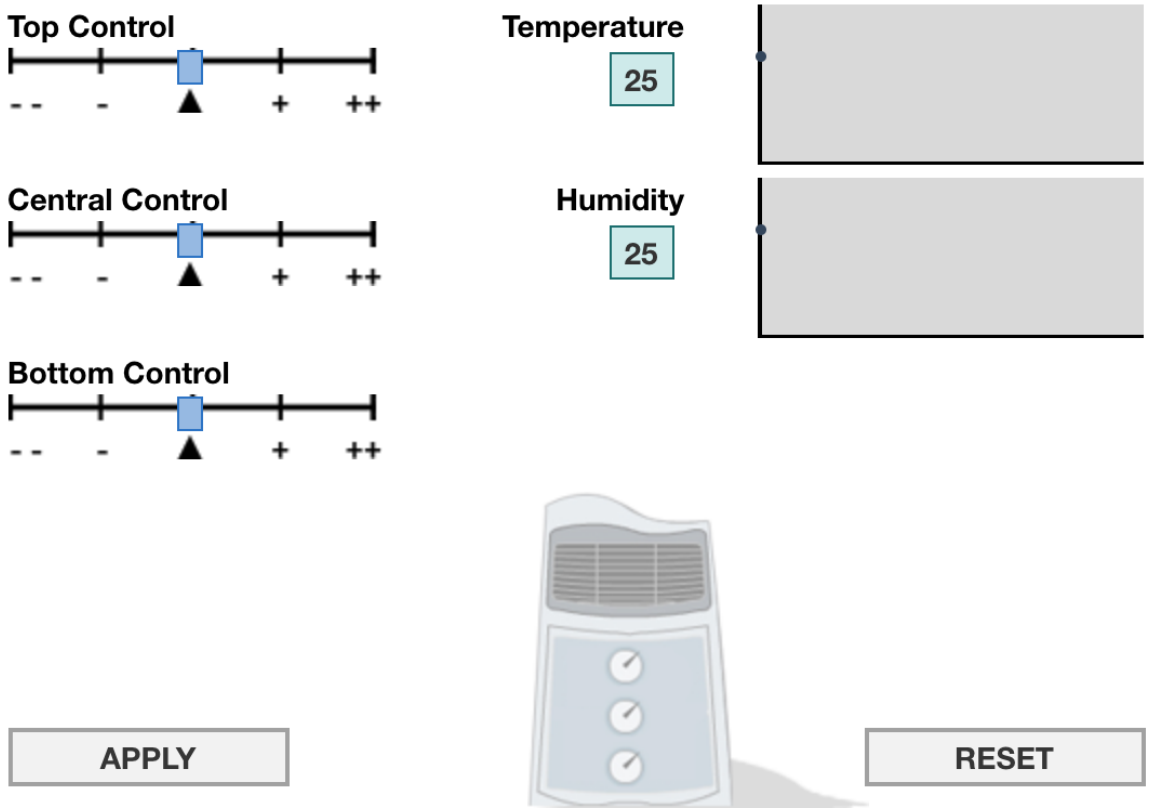
Among different applications of formative assessments, CBA for learning is extensively used nowadays. CBA for learning can be traced back to the early 1960s, when computers started to play a role in a variety of sectors. According to the review by Shute and Rahimi (2017), in the past (from the early 1960s to the late 1990s), CBA for learning only played a supplementary role in assisting instruction in classrooms. For example, computerized testing was used as a supplementary learning tool (Cartwright & Derevensky, 1975). More recently, CBA for learning started to be used to address more complicated competencies, such as problem-solving skills (e.g., Baker & Mayer, 1999). Nowadays, the explosive ICT developments in recent years are changing learning from instructor-centered to learner-centered and have given rise to the emerging use of new pedagogical approaches,

such as project-based learning (Bell, 2010), game-based learning (Kiili, 2005) and more recently, personalized learning (Shute et al., 2016). Inevitably, benefiting from ICT advances, CBA for learning is also transforming a variety of educational processes (e.g., Chatzopoulou & Economides, 2010; Joosten-ten Brinke et al., 2007; Peat & Franklin, 2002; Terzis & Economides, 2011). Incorporated with new technologies, CBAs can be designed to approximate real-life problem-solving environments with more integrative and interactive tasks (e.g., Azevedo et al., 2010; Blanchard et al., 2012), a desired feature of assessment for the 21st century (Shute & Becker 2010). Consequently, more complex and multidimensional learner competencies such as creativity, critical thinking, and problem-solving skills can be evaluated with CBAs in the digital age (e.g., Greiff et al., 2014; Pásztor et al., 2015; Rosen & Tager, 2014). In addition to the capacities of CBAs to measure high-level and complex skills, compared to conventional fixed-form assessments, CBAs can be scheduled and delivered to students in a more flexible and adaptive way. For example, in the recent work by Bulut et al. (2020), an intelligent recommender system was developed to determine the optimal time when a student need to be tested, which is capable of reducing redundant test administrations without impeding upon accurate evaluation of learning progress. To sum up, from both angles of skill evaluation and test administration, CBA for learning bears great potential for effective formative assessments.

An Example CBA Task on Problem Solving

Figure 2

Screenshot of a Problem-Solving Item on Climate Control in PISA 2012



A good example of CBA task is the problem-solving question of the Programme for International Student Assessment (PISA) in 2012, which is a large-scale international assessment program evaluating 15-year-old students' literacy in mathematics, reading and mathematics as well as their problem-solving competencies (Organisation for Economic Co-operation and Development, 2014). Figure 2 presents the interface of a sample question on climate control. The question asked students to figure out what each air conditioner control is used for (control temperature or control humidity). When solving the question, students adjusted different control values and checked the changes in temperature and humidity so that they were able to map each control to temperature or humidity. Students could play around with the controls multiple times before they gave the final answer. Contrast to standardized test questions, this question was designed to approximate real-life scenarios and students

could interact with the task. Moreover, students' actions and time durations for solving the problem are logged by the system, which can be used to uncover their problem-solving processes. For example, a good strategy for solving this problem is changing values of a control while keeping other controls constant. Whether students used this strategy or not cannot be directly observed through the product data (i.e., correctly solving the problem or not) but can be informed by their problem-solving action sequences.

Digital Game-Based Assessments and Evidence-Centered Design

Digital game-based assessment (DGBA) is a dominant family of CBA for learning, which attracts increasing interest from researchers and practitioners in education in recent years (Hwang & Wu, 2012). DGBAs are considered as stealth assessments given that learners' skills are evaluated unobtrusively and non-disruptively through their interactions with the game engine (Shute et al., 2009). The evidence-centered design (ECD) model is typically used as the conceptual underpinning for building a DGBA (e.g., DiCerbo, 2014; Plass et al., 2013; Rowe et al., 2015). Specifically, the conceptual assessment framework of an ECD includes three strongly related models which are the competency model, the task model and the evidence model (Mislevy et al., 2003).

The competency model, or the student model, defines what we intend to measure in the assessment such as skills and knowledge components. Notably, the variables in the competency model are latent, which cannot be directly observed but are inferred based on learners' observable performance indicators in the assessment (Mislevy et al., 2004).

The task model indicates how specific assessment tasks are designed based on which inferences on learners' levels of the targeted skills can be made. Specifically, tasks in a DGBA specify the concrete goals that learners are expected to achieve as they are progressing through the assessment. Tasks are largely different from the questions or items in conventional educational assessments because they are more complex, unstructured and

interactive. For example, for a standardized test on mathematics, all items are presented to learners in the same way and learners' responses to the items are typically predictable (e.g., correct/incorrect). However, for tasks in a DGBA, the same interface and game features might be presented to learners, but learners typically evolve the tasks into different paths or states during the gameplay, which constitute dynamic task model variables. That said, learners' actions and states in the assessment are hard to be predicted and modelled by standardized learning outcome modeling techniques.

The evidence model refers to how the assessment measures what it is designed to measure, which is a bridge between theory and data. More specifically, the evidence model is a mapping of learners' learning evidence from their interactions with the game engine to the targeted skills, which can be developed based on two phases: evidence identification and evidence accumulation. The evidence identification phase is the data reduction process which extracts appropriate observables from the performance data, while the evidence accumulation phase is the process of making inferences about learners' proficiencies of the latent skills based on the identified evidence. As mentioned earlier, learner performance data are recorded as both the product data and the process data in digital learning environments. The process data involves rich information on learners' unstructured observables such as mouse clicks and timestamps. Therefore, the evidence identification process is imperative for designing an effective DGBA given the limited interpretability of process data.

In summary, DGBAs use evidence extracted from learners' interactions with the game engine, which are explicitly elicited by the assessment tasks, to make inferences on their acquisition of the targeted skills. DGBAs have been successfully applied to a wide range of educational domains (e.g., language, Yukselturk et al., 2018; mathematics, Kiili & Ketamo, 2017; history, Kazanidis et al., 2018; computer science, Mathrani et al., 2016; geography, Gaydos, 2016). The success of DGBAs is due to its great potential for increasing

learners' interest, enjoyment, and motivation in learning (Erhel & Jamet 2013; Jackson & McNamara, 2013) as well as for helping individuals realize their learning goals (Divjak & Tomić, 2011) and improve their learning outcomes (Hsiao & Chen, 2016).

Intelligent Tutoring Systems

Another popular application of CBA for learning is an intelligent tutoring system (ITS), which is a computer system that diagnoses learners' cognitive states for individualized instruction and learning. ITSs have been widely used as an effective pedagogical approach since their inception in 1970 (Carbonell, 1970). According to the definition by Ma et al. (2014), an ITS involves two fundamental components: tutoring functions and student modeling functions. Tutoring functions are characterized by the learning and assessment content provided by the system and learners' interactions with the system. For example, in an ITS, the system provides learners with learning materials, assessment questions or tasks, and feedback or hints. Learners interact with the system by providing their problem-solving actions and answers. Student modeling functions involve the process of making inferences about learners' cognitive states based on learners' interactions with the system. For example, an ITS continuously estimates and updates a learner's skill mastery or understanding levels on the targeted skills measured by the system. Moreover, in an ITS, tutoring functions and student modeling functions work collaboratively. Tutoring functions enable student modeling functions to make inferences, which are in turn used to inform and adapt tutoring functions. For example, based on a learner's problem-solving actions and answer on a question, the system re-estimates his or her mastery level of the skill required by the question. If the system considers that the learner has improved the skill, advanced learning materials and harder assessment questions might be delivered to the learner in the next unit. Another well-accepted conceptualization of ITS includes four key components (Sottilare et al., 2013): a user interface model communicating information between learners and the system; a domain

model or cognitive model representing the knowledge, concepts, rules, and strategies of the domain to be learned by learners; a student model tracing learners' cognitive states based on their interactions with the system; and a tutoring model determining tutoring strategies and actions (e.g., offering a hint given an incorrect answer). Regardless of different conceptualizations, the core element of an ITS is the student model because it distinguishes ITSs from other CBAs for learning (Ma et al., 2014).

General Issues to Address

Irrespective of the content areas and the design features of CBAs, making accurate inferences about learners' cognitive states, or learner modeling, should be of high priority. For example, for the assessment of problem-solving skills in PISA 2012, it is desired to estimate each student's latent ability level on the problem-solving skill measured by the tasks. Generally, for CBAs with multiple independent assessment questions measuring a single latent skill, learners' cognitive states are inferred by modeling all question answers or item responses simultaneously (e.g., IRT models). However, for ITSs or other similar CBAs, it is required to trace learners' cognitive states based on the performance data. That being said, learners' history problem-solving attempts or item responses would affect their current or future problem-solving success because their cognitive states continuously change across multiple learning opportunities. In this regard, a sequential modeling technique (e.g., BKT) is required to monitor learners' cognitive states at each time point. For both sequential modeling and non-sequential modeling, from the methodological perspective, because inferences are made based on elicited learning outcomes by the system, a model is deemed effective for estimating learners' cognitive states if it is capable of accurately predicting or recovering item responses. For example, an effective sequential modeling technique should accurately predict a learner's present and future item responses given his or her previous item

responses; an effective non-sequential modeling technique should result in nonsignificant differences between observed item responses and model-predicted item responses.

In addition to learner modeling, learning outcome modeling is also used to make inferences on item-skill associations (i.e., domain modeling). Contrast to the mature methodological developments for learner modeling, approaches for domain modeling in the context of CBAs for learning are relatively underdeveloped. For most CBAs, domain experts play a role in specifying the skills, knowledge components and production rules for the assessment. As such, item-skill associations are pre-specified in most CBAs. However, as mentioned earlier, human judgements are not always guaranteed to result in precise item-skill associations, and it is costly and less feasible to rely purely on human efforts given a great number of assessment items. This is a call for more data-driving approaches developed to account for item-skill associations in the context of CBAs for learning.

Finally, the chapter emphasizes that for CBA for learning, irrespective of the content areas and design features, the "computer-based" nature enables that both learners' work products and problem-solving activities can be recorded and accessed by the system. Learners' work products on assessment tasks are often evaluated and quantified against objective criteria such as binary or polytomous scoring. The product data is used as the input for the majority of learning outcome modeling techniques. For example, IRT models estimate learners' latent ability levels based on the product data with binary or polytomous scores on each assessment item. However, the process data is mostly implicit and unstructured. Therefore, it is difficult to develop a generic approach for handling the process data produced by different CBAs. Process data analysis was often conducted in an exploratory manner to derive learners' problem-solving patterns or strategies (Abele, 2018; Greiff et al., 2015; Greiff et al., 2016; Molnár & Csapó, 2018; Stadler et al., 2019), which vary significantly across different CBAs. In recent years, some studies have demonstrated the potential of

analyzing process data based on conventional psychometric or statistical models (Chen et al., 2019; Liu et al., 2018; Shu et al., 2017). More details of these studies are reviewed in the following sections.

To sum up, CBA for learning is a popular type of formative assessments in education, which is capable of making deeper and finer-grained evaluation of learner performance as well as making inferences on more complex skills. Learner modeling is the core of CBA for learning, but the modeling techniques vary across different CBAs given their unique assessment purposes and design features. Moreover, regardless of CBA types, learner data is typically exploited solely in the form of explicit product data rather than process data. Developing generic approaches for analyzing process data in the context of CBA for learning is still in its infancy.

The following sections review a wide range of approaches for learner and domain modeling. Generally, these approaches can be categorized as psychometric measurement models, Bayesian networks, BKT, additive factors models, deep learning-based approaches, and collaborative filtering approaches. Moreover, the chapter provides an overview of some pioneering work on process data analysis.

Psychometric Measurement for Learning Outcome Modeling

Learning outcome modeling is an everlasting topic in the fields of educational measurement and psychometrics. Educational measurement is a discipline focusing on the use of methodologies for assigning scores obtained from educational assessments to students, based on which inferences about the abilities, knowledge, and skills of students can be made. In terms of analytic approaches, educational measurement overlaps psychometrics, a discipline focusing on the theory and methodologies of psychological measurement. The dissertation therefore uses the term “psychometric measurement” to indicate the commonalities between the two disciplines. Essentially, an analytical approach in

psychometric measurement is a type of learner modeling techniques estimating learners' latent ability levels or presence/absence of latent skills based on test scores. However, the majority of psychometric measurement approaches are theory-driven, which is a distinctive feature in comparison with data-driven approaches in computing science. In the following, the chapter briefly introduces the most basic modeling techniques in psychometric measurement. These techniques were developed based on three psychometric measurement theories: classical test theory (CTT), IRT, and CDM.

Classical Test Theory

CTT was the dominant approach prior to IRT and yet it is still used widely in practice due to its simplicity and interpretability. A key assumption of CTT is that a learner's observed test score X is equal to the sum of the learner's innate true score T and the measurement error E (Spearman, 1904):

$$X = T + E. \quad (1)$$

For example, if a learner has actually mastered 50% of the knowledge required by a test and 50% is the learner's true score, the learner might have an observed test score between 45% to 55% because there is 5% discrepancy from the true score due to errors of measurement. The errors of measurement E are assumed to follow a normal distribution with a mean of zero, which indicates that the average score of the distribution of observed test scores for a learner who takes a test an infinite number of times would be equal to that test-taker's true score. Based on a set of CTT assumptions (Kline, 2005), descriptive statistics of items such as mean and standard deviation, item difficulty levels and item discrimination indices can be derived to examine the quality of the assessment instrument. In CTT, a learner's cognitive state is typically calculated as the total score of the test. That said, multiple test items are assumed to measure a single latent skill, and the sum of learners' scores on each item indicates their proficiency levels on the skill. Regarding item-skill associations, CTT uses item

discrimination indices to indicate the associations between items and the latent skill. Item discrimination refers to the degree to which an item is capable of differentiating learners with high proficiency levels on the targeted skill from learners with low proficiency levels on the targeted skill, which is used as the hallmark of a good test item in practice. The point biserial correlation ρ between dichotomous item scores of an item and the continuous total test scores is used as an item discrimination index in CTT, which is given by:

$$\rho = \frac{(\mu_{correct} - \mu_X)}{\sigma_X}\sqrt{p/(1-p)}\,, \tag{2}$$

where $\mu_{correct}$ indicates the mean of the learners' total test scores who get the item correct, μ_X and σ_X denotes the mean and standard deviation of all learners' total test scores respectively, and p refers to the item difficulty which is the percentage of learners who get the item correct. Item discrimination is within a range between 0 and 1 and expected to be as large as possible given that higher discrimination levels indicate stronger affinity of an item to the latent skill.

Despite its simplicity, CTT has several disadvantages. The item parameters and learner ability estimates approximated by CTT are greatly dependent on the test items and the examinee group. That said, the difficulty and discrimination of items are likely to be different given different groups of learners tested by the items, and the learners' estimated ability levels are likely to vary if they are tested with different sets of items measuring the same latent skill. Moreover, the measurement error in CTT is the same for all learners given that they are estimated at the test level.

Item Response Theory

Compared with CTT, IRT demonstrates several advantages. In IRT, item parameters are invariant to the examinee groups and learner ability levels are invariant to the test items. In addition, the measurement error in IRT is estimated for different learner ability levels, which implies that the extent to which each test item precisely measures each learner's latent

ability can be informed by IRT. Also, IRT assumes that only one dominant skill is allowed to be measured in a test and the probability of a learner answering an item correctly is independent from his or her odds of success on other items (Reise et al., 2005).

IRT models are a type of latent variable models which estimate learners' probabilities of answering an item correctly through a set of item and learner parameters. Specifically, item parameters in IRT are item difficulty, item discrimination, and item guessing, and the learner parameter indicates a learner's proficiency level on the targeted skill. Different IRT models assume different degrees of item parameterization. For example, the Rasch model (Rasch, 1960) is the most parsimonious IRT model where items are only parameterized with item difficulty. By Rasch model, learner i's probability of correctly answering item j is given by:

$$P(R_{ij} = 1|\theta_i) = \frac{1}{1 + e^{-(\theta_i - b_j)}}, \tag{3}$$

where $R_{ij} = 1$ indicates that learner i gets a score of 1 on item j, θ_i denotes learner i's proficiency level on the latent skill, and b_j refers to the difficulty of item j. Compared with the Rasch model, the two-parameter logistic (2PL) model additionally parameterizes item discrimination. By the 2PL model, learner i's probability of correctly answering item j is given by:

$$P(R_{ij} = 1|\theta_i) = \frac{1}{1 + e^{-a_j(\theta_i - b_j)}}, \tag{4}$$

where $R_{ij} = 1$ indicates that learner i gets a score of 1 on item j, θ_i denotes learner i's proficiency level on the latent skill, b_j refers to the difficulty of item j, and a_j refers to the discrimination of item j. Compared with the 2PL model, the three-parameter logistic (3PL) model additionally parameterizes item guessing. Similarly, given the 3PL model, learner i's probability of correctly answering item j can be formulated as:

$$P(R_{ij} = 1|\theta_i) = c_j + \frac{1 - c_j}{1 + e^{-a_j(\theta_i - b_j)}}, \quad (5)$$

where c_j denotes the guessing parameter of item j and the other parameters are the same as the 2PL model. For IRT models, the item discrimination parameter can be used to indicate the item-skill associations. Higher item discrimination level implies stronger affinity of the item to the latent skill, and as a result, the item is more capable of differentiating high-performing learners from low-performing learners. Given the formulation of IRT models, it can be seen that the item-learner interaction is modelled by the linear combination of learner ability, item difficulty, item discrimination and item guessing, which is then non-linearly converted to a predicted probability of correct item response ranging from 0 to 1 through a sigmoid transformation.

The above IRT models are all assumed to be unidimensional. However, in reality, the majority of educational assessments are designed to evaluate multiple skills or knowledge components, and traditional IRT models fail to deal with multidimensional data. More recently, the approach of multidimensional item response modeling (MIRT) was proposed to address multiple latent skills of multidimensional data (Reckase, 1997; Yao & Boughton, 2007). Compared with conventional IRT models, the MIRT approach allows items to measure different skills which enables a finer-grained analysis of learner data. In MIRT, different item difficulties and learner abilities are estimated for multiple latent skills. For example, given the multidimensional 3PL model, learner i's probability of correctly answering item j can be formulated as:

$$P(R_{ij} = 1|\vec{\theta}_i) = c_j + \frac{1 - c_j}{1 + e^{-\vec{a}_j \odot \vec{\theta}_i{}^T + b_j}}, \quad (6)$$

where b_j and c_j denote item difficulty and item guessing of item j respectively, which are scalar parameters. $\vec{\theta}_i$ is a vector parameter indicating learner i's proficiency levels on multiple latent skills, and $\vec{a}_j$ is a vector parameter indicating item j's discrimination levels on

multiple latent skills. As such, given a vector of item discrimination, the MIRT approach is capable of providing each item with item-skill associations for multiple skills, and given a vector of learner ability, each learner can be estimated with multiple proficiency levels on multiple latent skills. However, given that MIRT models bear greater complexity, they have not been widely used in the digital learning environments (Desmarais & Baker, 2012).

Table 1

A Sample Q-Matrix with Five Items and Three Skills

	Skill 1	Skill 2	Skill 3
Item 1	0	1	0
Item 2	0	0	1
Item 3	1	0	0
Item 4	1	1	1
Item 5	1	0	1

Cognitive Diagnosis

Cognitive diagnosis is an approach for profiling learners with information on mastery or non-mastery of multiple skills (Rupp et al., 2010). CDMs calculate the probability of a correct response based on learners' mastery profile of the skills that are measured by an item (e.g., Henson et al., 2009; Tatsuoka, 1983). Given the mastery profile of the required skills, learners can be evaluated with fine-grained diagnostic information, which in turn supports targeted interventions for learning. Similar to MIRT, CDMs also address multiple latent skills. However, unlike MIRT, CDMs requires a pre-specified mapping of items to skills, which is called the Q-matrix, for item parameterization and model estimation. In a Q-matrix, the columns and rows represent the required skills and the test items respectively with matrix entries of 0s or 1s indicating the mapping of one item to one skill. Table 1 presents a sample Q-matrix which involves five items and three skills. It can be seen that item 1 only requires skill 2, whereas item 5 requires both skills 1 and 3. Moreover, the entries of Q-matrix can be polytomous (e.g., 0, 1, and 2), indicating the degree to which an item measures a skill (von

Davier, 2005). In this sense, the Q-matrix naturally represents the item-skill associations for learning outcome modeling.

Over the past several decades, a great number of CDMs have been proposed in the psychometric measurement literature. Despite a variety of modeling techniques designed for various purposes, the majority of CDMs can be characterized by several general modeling frameworks. With the general modeling frameworks, other specific CDMs can be derived through statistical constraints on model parameters. Therefore, given the limit of the space, the dissertation describes CDMs by a brief introduction to a general modeling framework, the log-linear cognitive diagnosis model (LCDM; Henson et al., 2009). However, the general modeling frameworks are saturated models developed at the sacrifice of model simplicity, which might not be optimal for practice. Given LCDM, learner i's probability of correctly answering item j can be formulated as:

$$P(R_{ij} = 1|\vec{\alpha}_i) = \frac{1}{1 + e^{-(\lambda_{j,0} + \vec{\lambda}_j' \mathbf{h}(\vec{\alpha}_i, \vec{q}_j))}}, \tag{7}$$

where $\vec{\alpha}_i = (\alpha_{i1}, \ldots, \alpha_{iK})$ denotes the skill mastery profile of learner i on K latent skills, $\lambda_{j,0}$ represents the intercept parameter of item j, $\vec{q}_j = (q_{j1}, \ldots, q_{jK})$ indicates the Q-matrix entries for item j, and $\mathbf{h}$ is a mapping function that linearly combines $\vec{\alpha}_i$ and $\vec{q}_j$:

$$\vec{\lambda}_j' \mathbf{h}(\vec{\alpha}_i, \vec{q}_j) = \sum_{k=1}^{K} \lambda_{j,1,(k)} \alpha_{ik} q_{jk} + \sum_{k=1}^{K-1} \sum_{k'>k} \lambda_{j,2,(k,k')} \alpha_{ik} \alpha_{ik'} q_{jk} q_{jk'} + \cdots. \tag{8}$$

For item j and a total K latent skills, the probability of a correct response is affected by the main effect of each skill and the interaction effects between skills. As such, in the equation, $\lambda_{j,1,(k)}$ indicates the main effect of skill k, and $\lambda_{j,2,(k,k')}$ refers to the two-way interaction effect between skills k and k'. Moreover, because item j might not require and learner i might not master all the K latent skills, the terms $\alpha_{ik} q_{jk}$ and $\alpha_{ik} \alpha_{ik'} q_{jk} q_{jk'}$, which are the product between the skill mastery profile of learner i and the Q-matrix entries for item

j, are used to control which main and interaction effects would be present for the learner-item interaction between learner i and item j. The LCDM accounts for all possible effects of the presence or absence of skills on item responses, which involves a great number of model parameters to be estimated. In addition, for a real-world educational assessment, its required skills might not pose all main and interaction effects on learners' item response. Therefore, more parsimonious models are typically desired in practice. Given LCDM, removing all main and lower-order interaction effects and only retaining the highest-order interaction effect result in the deterministic inputs noisy and gate model (DINA; Haertel, 1989; Junker & Sijtsma, 2001), and removing all interaction effects and retaining the main effects lead to the compensatory re-parameterized unified model (C-RUM; Hartz, 2002), which are parsimonious models with much fewer model parameters. Contrast to LCDM and its variants, another type of CDMs uses a statistical pattern recognition approach to diagnose learners' skill mastery profiles. The most well-known models are the rule space model (Tatsuoka, 1983) and the attribute hierarchy model (Leighton et al., 2004).

Applications and Challenges of Psychometric Measurement Models

Psychometric measurement models, especially IRT models, are used in a variety of application domains. For example, in the large-scale international assessment programs for evaluating students' academic achievement such as PISA (Organisation for Economic Co-operation and Development, 2014), the trends in international mathematics and science study (TIMSS; Mullis et al., 2016), and the progress in international reading literacy study (PIRLS; Mullis et al., 2017), IRT models are used to derive the final achievement scores and scale composite scores for the purposes of increasing the accuracy of the measurement and reducing the sampling bias. Moreover, because both computer-based and paper-based assessment formats might be used in these programs, IRT models can be used to examine the measurement invariance given different assessment modes (Organisation for Economic Co-

operation and Development, 2015). In addition to large-scale assessments, the potential and effectiveness of IRT models in practice are well revealed in a variety of application sectors such as clinical psychology (Reise & Waller, 2009), medical education (De Champlain, 2010), computing science (Martínez-Plumed et al., 2016), and management (Carroll et al., 2016). Compared with IRT models, most cognitive diagnosis research is simulation-based and mature applications of cognitive diagnosis in practice are relatively fewer. The major challenge for cognitive diagnosis in practice is that an accurate Q-matrix is hard to be pre-specified. Despite content experts' knowledge, flaws in pre-specified Q-matrices would significantly undermine the diagnosis performance (Hansen et al., 2016; Liu et al., 2016). However, the potential of cognitive diagnosis for a finer-grained analysis of education and psychology data has been examined in the application fields such as clinical psychology (de la Torre et al., 2018; Templin & Henson, 2006) and language testing (Jang, 2009).

Regarding the psychometric measurement analysis of learner performance data of CBAs, de Klerk et al. (2015) presented a comprehensive review of 31 articles on the topic. According to their review, the majority of CBAs measure mathematics and science concepts (e.g., Kerr & Chung, 2012; Quellmalz et al., 2013) and a few of them measure complex skills such as 21st century skills (Shute & Ventura, 2013), problem-solving skills (Shute et al., 2009), and causal reasoning (Shute, 2011). In addition, all of the CBAs they reviewed modeled learners' product data and 50% of them modeled learners' process data or the combination of process data and product data. Regarding the analytical approaches, both exploratory techniques and confirmatory techniques were employed to analyze the learner performance data of CBAs. The exploratory techniques were used to identify the patterns in the performance data and investigate how the patterns are related to the latent skills measured by the assessment (e.g., Gobert et al., 2012; Halverson & Owen, 2014). These analyses were mostly based on educational data mining techniques, such as cluster analysis. It should be

noted that exploratory analyses of performance data for identifying learning patterns are greatly dependent on specific learning contexts, assessment design features and learner characteristics, which are typically not generalizable across different CBAs. The confirmatory techniques were used to model performance data for making probabilistic inferences on learners' proficiency levels of the latent skills (e.g., Klinkenberg et al., 2011; Quellmalz, et al., 2012). Despite the fact that identifying learning patterns from performance data is a mainstream research topic in the area, given its purposes and scope, the dissertation only focuses on the confirmatory learning outcome modeling which makes probabilistic statements on learners' latent skills.

According to de Klerk et al. (2015), the confirmatory techniques used to analyze CBA performance data mainly involve confirmatory factor analysis, CTT and IRT models, and Bayesian networks. Although the review by de Klerk et al. (2015) contended that the Bayesian network is the most frequently used "psychometric model" for CBAs, using the typical taxologies of models in education measurement (Brennan, 2006) and data mining (Tan et al., 2016), the chapter discusses Bayesian networks for learning outcome modeling in the next section.

Regarding the purposes of psychometric measurement analysis of CBA data, in general, CTT and IRT models were mainly used to estimate learner ability and examine the psychometric quality of assessment items. In a CBA designed for architect registration examination, CTT and IRT models were used for automated scoring of complex responses constructed by examinees (Braun et al., 2006). In a CBA designed for students to practice arithmetic, an IRT model was used to adaptively estimate learners' latent ability levels and items' difficulty levels based on both the product data and the response time, which contributed to improved measurement precision and reliability and stronger validity (Klinkenberg et al., 2011). In their CBA for science inquiry, Quellmalz et al. (2010) analyzed

learner performance data with an IRT model to examine the dimensionality and technical quality of the items. In another CBA for learning about ecosystems, force, and motion, a MIRT model was used to examine the dimensionality and technical quality of the items (Quellmalz et al., 2012). In summary, despite some studies accounting for partial information provided by the process data (e.g., Klinkenberg et al., 2011), it is evident that conventional psychometric measurement models could not be used to fully address the complexity of process data for learning outcome modeling.

Moreover, as mentioned previously, psychometric measurement models require strong theoretical assumptions regarding how skills are measured by items. Particularly, CTT and IRT models assume a single skill to be measured, which is largely infeasible for fine-grained cognitive diagnosis and multiple skill modeling. However, for cognitive diagnosis models, they typically require an accurate Q-matrix prespecified by domain experts, which inevitably brings flaws and limits the scalability for modeling. Furthermore, given the strong theoretical assumptions, domain modeling is typically infeasible given the standard forms of psychometric models.

Bayesian Networks

Bayesian networks were widely used to make probabilistic inferences based on the learner performance data of CBAs (de Klerk et al., 2015). Bayesian networks are a type of probabilistic graphic models, which graphically represent a joint distribution of a set of random variables (Koller & Friedman, 2009). Essentially, building a Bayesian network requires the specification of a directed acyclic graph and a table of probability distributions for each variable, or called node, in the graph. Figure 3 presents an example Bayesian network with three nodes. In the network, each node represents a random variable and directional edge represents the dependency or the causal relationship between two random variables. The two bottom nodes of squares represent two assessment items indicating

learners' item responses (i.e., correct/incorrect) and the top node of oval represents the latent skill measured by the two items (i.e., mastery/non-mastery). As such, the example network depicts that learners' probabilities of giving correct or incorrect responses to item 1 and item 2 are dependent on their probabilities of having the latent skill mastered or not. To know the joint distribution of the network, it is required to define the distribution of the latent skill and the conditional distributions for item 1 and item 2. Mathematically, the joint distribution of the three nodes shown in the example network is given by

$$P(skill, item\ 1, item\ 2) = P(item\ 1|skill)P(item\ 2|skill)P(skill), \quad (9)$$

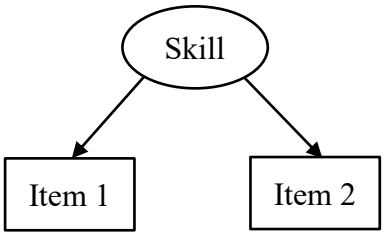
where $P(item\ 1|skill)P(item\ 2|skill) = P(item\ 1, item\ 2|skill)$ given that item 1 and item 2 are conditionally independent from each other. Practically, making inferences about learners' mastery levels on the latent skill given the information of their item responses to item 1 and item 2 can be characterized as the process of finding $P(skill|item\ 1, item\ 2)$; predicting learners' item responses to item 1 or item 2 can be represented as the process of finding $P(item\ 1|skill)$ or $P(item\ 2|skill)$. Estimating the conditional probabilities mentioned above can be formularized as a maximum likelihood estimation problem as other statistical learning models (Heckerman et al., 1995). More concretely, for the network shown in Figure 3, the parameters to be estimated include two conditional probabilities for the two possible values of item 1 (i.e., correct and incorrect), two conditional probabilities for the two possible values of item 2 (i.e., correct and incorrect), and two probabilities for the latent skills (i.e., mastery and non-mastery). The problem of estimating all the above parameters Θ given a dataset D can be formulated as

$$\Theta_{\mathrm{ML}} = \arg\max\{L(\Theta : D)\}. \quad (10)$$

Heckerman et al. (1995) provide more details regarding the parameter estimation for Bayesian networks.

Figure 3

An Example Bayesian Network with Two Items and One Latent Skill



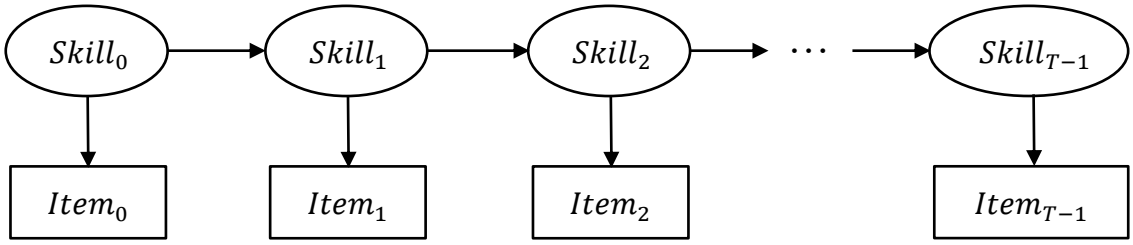
For the majority of CBAs, multiple learning opportunities are designed for learners to practice the latent skills. Bayesian networks, however, cannot address the temporal dependencies between the multiple learning opportunities presented through a CBA. For events occurring over a period, dynamic Bayesian networks (DBNs) can be used to account for the temporal dependencies between multiple timesteps in making inferences on the conditional probabilities of random variables. Essentially, DBN is an extended version of Bayesian networks with time information. Figure 4 presents an example DBN with one latent skill and one observable item across a total of T learning opportunities. Each learning opportunity represents one timestep. The T timesteps are connected by the temporal relationships of the latent skill between one timestep and its subsequent timestep. It should be noted that, without the temporal connection, the nodes and edges for each timestep constitute a simple Bayesian network, and the DBN can be considered as T connected copies of the simple Bayesian network. By modeling the temporal dependencies between timesteps, the state of the latent skill changes over time can be inferred. This feature of DBN is especially useful for CBAs because individuals' learning progress can be tracked by analyzing their item responses with DBN. Given the DBN shown in Figure 4, the state of the latent skill at a certain timestep (i.e., mastery/non-mastery) is dependent on both the state of the latent skill at the previous timestep and the current state of the item (i.e., correct/incorrect). Therefore, the joint distribution of the latent skill $Skill = \{Skill_0, Skill_1, \dots Skill_{T-1}\}$ and the item $tem = \{Item_0, Item_1, \dots Item_{T-1}\}$ over T timesteps is given by

$$P(Skill, Item) = \prod_{t=1}^{T-1} P(Skill_t|Skill_{t-1}) \prod_{t=0}^{T-1} P(Item_t|Skill_t)\, P(Skill_0), \qquad (11)$$

where $P(Skill_0)$ is the prior distribution of the skill, $P(Item_t|Skill_t)$ indicates the observation distribution of the item dependent on the skill at timestep t, and $P(Skill_t|Skill_{t-1})$ refers to the state transition distribution presenting how the state of the latent skill at timestep t is affected by its state at the previous timestep $t-1$. Given the above formulation, the problem of estimating how the state of the latent skill changes over time can be solved by finding the conditional probabilities $P(Skill|Item)$, where $Skill = \{Skill_0, Skill_1, \ldots Skill_{T-1}\}$ and $Item = \{Item_0, Item_1, \ldots Item_{T-1}\}$. It should be noted the DBN is established based on the Markov assumption, which states that the conditional probability of the latent skill at timestep t is only dependent on the state of the latent skill at timestep $t-1$; the states of the latent skill at timesteps prior to $t-1$ are of no influence (Koller & Friedman, 2009).

Figure 4

An Example Dynamic Bayesian Network with One Item and One Latent Skill



Regarding domain modeling, standard Bayesian networks and DBNs fail to automatically estimate the item-skill associations because a correspondence between each item and the skill it measures is required to be pre-specified to construct the graphical model. Therefore, similar to psychometric measurement models, domain expertise is required for Bayesian networks and DBNs.

Bayesian networks or DBNs showed great potential for learning outcome modeling in the literature because of their strong flexibility, high expressiveness, and sound computations

(Desmarais & Baker, 2012). For example, Bayesian networks have been used to model the performance data of CBAs on computer networking skills (Levy & Mislevy, 2004; Levy, 2013; Rupp et al., 2012; West et al., 2012), dental practice (Mislevy et al., 2002), systems thinking (Mislevy et al., 2014; Shute et al., 2010), creative problem solving (Shute et al., 2009), causal reasoning (Shute, 2011) and 21st century skills (Shute & Ventura, 2013). For the CBAs with multiple learning opportunities, DBNs have been used to address the temporal dependencies and model the performance data for CBAs on air combat (Poropudas & Virtanen, 2007), weather phenomenon (Cui et al., 2019), mathematics (Levy, 2014), and Navy damage control operations (Iseli et al., 2010; Koenig et al., 2010). Bayesian networks and DBNs were successfully applied in various CBAs in a wide range of application domains. However, the applications of Bayesian networks and DBNs in these studies did not address well learners' process data for learning outcome modeling, although the analyses were conducted in the context of CBAs. This is most likely due to Bayesian networks requiring their input data to be structured for defining the conditional probabilities of all possible variable values. However, learners' actions and time durations during gameplay are typically unstructured and no generalizable data pre-processing techniques are available to reorganize the process data. Hence, Bayesian networks are still limited in dealing with the performance data produced by CBAs.

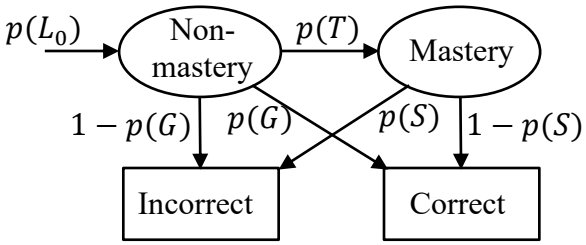
Moreover, in pursuit of strong flexibility, Bayesian networks are used subject to the curse of dimensionality. That is, a Bayesian network might involve a great number of latent variables, which results in complex computation of the conditional probabilities. To address this, Bayesian networks can be constructed with simplifying assumptions or in a data-driven way that handles the observable variables only and reduces the network complexity (Desmarais & Baker, 2012).

Bayesian Knowledge Tracing

BKT (Corbett & Anderson, 1994) is a learning outcome modeling approach extensively used in the community of educational data mining, especially for tracking learners' changes of cognitive states over time in intelligent tutoring systems. Essentially, BKT is a constrained and simplified version of DBN, where the number of conditional probabilities is reduced for modeling.

Figure 5

Graphical Representation of Bayesian Knowledge Tracing



Concretely, given the DBN case shown in Figure 4, the item has either correct or incorrect state, whereas the skill has either a mastery or a non-mastery state. As such, the horizontal directional edges represent the transition probability from non-mastery to mastery of the skill, denoted as $p(T)$, and the transition probability from mastery to non-mastery of the skill, denoted as $p(F)$. The transition probability, $p(T)$, indicates the learning of the skill and the transition probability, $p(F)$, indicates the forgetting of the skill. The vertical directional edges represent the emission probability of incorrectly answering the item given a mastery of the skill, denoted as $p(S)$, and the emission probability of correctly answering the item given non-mastery of the skill, denoted as $p(G)$. The emission probability $p(S)$ is the *slip* probability and the emission probability $p(G)$ is the *guess* probability. Moreover, the DBN requires a definition of the prior probability of mastering the skill, which is denoted by $p(L_0)$. These conditional probabilities can be used to calculate the conditional probabilities of

mastery or non-mastery of the skill given correct or incorrect responses, which are in turn used to infer the probabilities of mastery or non-mastery of the skill at each timestep.

BKT is a special case of the DBN described above, where the transition probability $p(F)$ is fixed as 0, indicating an assumption that learners will never forget the learned skill. The model parameters of BKT are graphically represented in Figure 5. Despite two ovals and two squares shown in Figure 5, they represent the two states of the skill and the item, rather than two skills and two items. Given the above definitions of model parameters, BKT estimates the conditional probabilities of mastering the skill given either correct or incorrect responses at the timestep t as:

$$\begin{aligned} p(L_t|response = correct) &= \frac{p(L_{t-1})[1-p(S)]}{p(L_{t-1})[1-p(S)] + p(G)[1-p(L_{t-1})]} \\ p(L_t|response = incorrect) &= \frac{p(L_{t-1})p(S)}{p(L_{t-1})p(S) + [1-p(G)][1-p(L_{t-1})]}. \end{aligned} \tag{12}$$

Having the two conditional probabilities, BKT proceeds to estimate the probability of mastering the skill at timestep t given a learner's correct or incorrect response as:

$$p(L_t) = p(L_t|response) + [1 - p(L_t|response)]p(T). \tag{13}$$

It should be noted that each latent skill should be estimated and updated by a different BKT model, indicating that different skills must work independently in influencing learners' item response in BKT. However, in DBN, skills can be interconnected by content knowledge for constructing the graphic model. That said, compared with BKT, DBN allows for relationships between skills, which contributes to stronger representation of the complexity of the learner performance data. However, given a limited sample size and relatively simple relationships between skills, the performance in inferring learners' cognitive states between DBN and BKT was found to be trivial in some applications (e.g., Cui et al., 2019).

Consistent with DBN, standard BKT cannot be used to directly model item-skill associations because the items for each skill are required to be known for the estimation of

conditional probabilities. Nevertheless, Lindsey et al. (2014) showed the potential of BKT for discovering item-skill associations by developing a BKT-based generative probabilistic model with experts' knowledge as a prior.

Additive Factors Model

Additive Factors Model (AFM; Cen et al., 2005; Cen et al., 2006) is a statistical model proposed in the community of educational data mining for modeling learners' probabilities of correctly answering items. Similar to IRT models, AFM estimates learners' cognitive states for a given skill, which are converted to probabilistic predictions of item responses by a logistic function (i.e., the sigmoid transformation). However, unlike IRT models, AFM can be considered as an alternative to BKT for sequential modeling of learners' changes of cognitive states over time. Concretely, the AFM models learner i's probability of correctly answering item j as:

$$P(R_{ij} = 1|\theta_i) = \frac{1}{1 + e^{-(\theta_i + \sum_{k=1}^{K} \beta_k q_{jk} + \sum_{k=1}^{K} \gamma_k q_{jk} t_{ik})}}, \quad (14)$$

where θ_i indicates learner i's latent ability level, β_k indicates the easiness of the skill $k \in \{1, \dots, K\}$, γ_k denotes the learning rate of the skill k, q_{jk} indicates whether skill k is measured by item j, t_{ik} denotes the total number of learning opportunities learner i has previously accessed for practicing skill k, and K is the number of latent skills measured by the assessment.

Given the above formulation, it can be seen that, because AFM accounts for multiple learning opportunities (i.e., t_{ik}), learners' progress of learning the latent skills can be tracked with AFM, which is a major difference from other logistic function-based models. Moreover, a pre-specified mapping of items to skills (i.e., q_{jk}) is required for constructing AFM, indicating that AFM cannot be used for domain modeling which learns item-skill associations from scratch. More recently, a dynamic and deep variant of AFM, dAFM, was proposed to

address the domain modeling issue of AFM (Pardos & Dadu, 2018). Compared with AFM, the dAFM models learners' changes of cognitive states over time by the following two changes: the mapping of items to skills (i.e., q_{jk}) is adjustable rather than fixed in dAFM and the counts of learning opportunities for practicing skills are dynamically updated rather than fixed as the mapping of items to skills changes in learning the data. To achieve this, a recurrent neural network layer is used as a learning opportunity counter in dAFM. Essentially, dAFM was developed based on a deep learning framework. More details on deep learning and recurrent neural networks would be given in the following sections.

Deep Learning for Learning Outcome Modeling

In recent years, deep learning has received a great deal of attention for its predictive capacity in a wide range of applications domains (e.g., biology and medicine, Ching et al., 2018; medical imaging, Suzuki, 2017; speech recognition, Amodei et al., 2016; image recognition, Ciregan et al., 2012; learning analytics and educational data mining, Coelho & Silveira, 2017). Deep learning can be defined as "a class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification" (Deng & Yu, 2014). As a subfield of machine learning, deep learning automatically makes predictions or decisions based on learning labelled or unlabelled sample data. In terms of learning outcome modeling, a deep learning-based model is often developed to predict learners' item responses for each item. Therefore, the output of the deep learning model should be item responses, such as correct or incorrect scores. However, the input for the deep learning model can be various. For example, given a simple item response matrix without any other information, a deep learning model can simply learn the identifications of each item and each learner as input. Given the availability of more information regarding the items and the learners (e.g., item text, learner background information), a deep learning model can learn the

additional information as input. In case that process data is available, a deep learning model can learn individuals' action sequences and time durations as input. In deep learning, how the input is integrated, analyzed, and learned for outputting the final predictions is determined by the deep learning architecture.

In a nutshell, deep learning is essentially an extension of artificial neural networks with multiple hidden layers. There are a variety of deep learning architectures developed for different application domains. In the following, the chapter briefly covers the two most fundamental architectures of deep learning, which are closely pertinent to the topic of learning outcome modeling and the proposed models in the dissertation: deep neural networks and recurrent neural networks. The former is typically used to capture the complexity of the input data and the latter is typically used to model the temporal dependencies between multiple timesteps of the input data. The two architectures of deep learning align well with the purposes of this dissertation, given that sequential modeling and process data learning are incorporated in the proposed approaches.

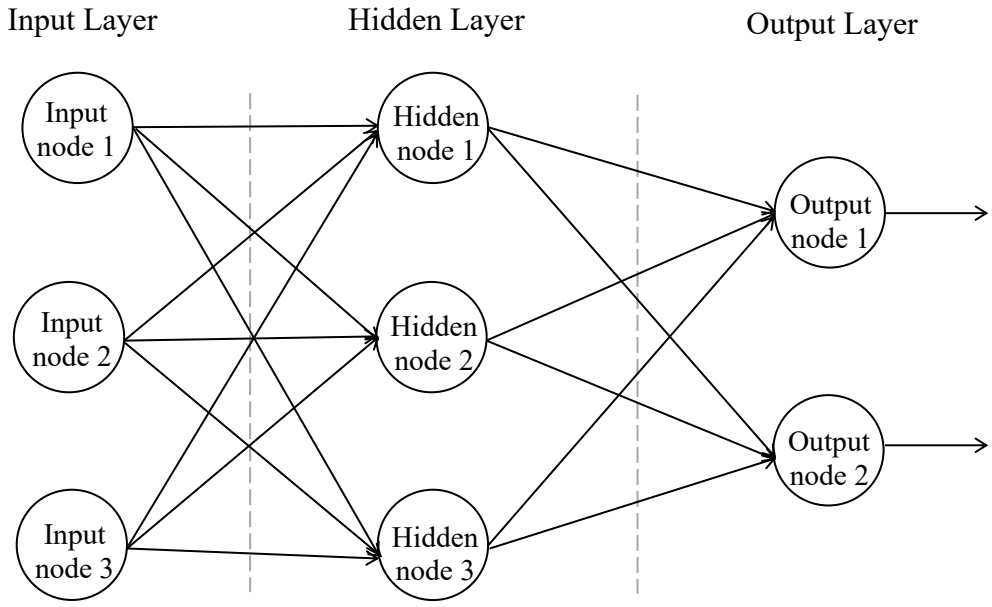
Deep Neural Networks

Deep neural networks are extensions of feedforward neural networks, or multilayer perceptron (Goodfellow et al., 2016), which are the basis for deep learning. The feedforward neural network is analogous to the biological neural network in the human brain controlling how the information is processed. Figure 6 presents a graphical representation of an example feedforward neural network. This network has one input layer of three nodes, one hidden layer of three nodes, and one output layer of two nodes. In the network, the basic components of a neural network are nodes, which receive inputs from the previous nodes and produce outputs for the following nodes. There are three types of nodes: the input nodes, the hidden nodes, and the output nodes. The input nodes receive the input data while each node represents a feature. For example, if the example neural network learns individuals' GPA,

motivation, and gender as inputs for predicting their responses to one item, the three input nodes represent the three features of GPA, motivation and gender. The output nodes represent the final predictions. The hidden nodes work as the bridge connecting the input and the output nodes, with information transferred in between. Between the nodes, there are directional edges, which connect the nodes with different strengths. Statistically, these directional edges are different weights indicating the relative connection strength between nodes.

Figure 6

Graphical Representation of an Example Feedforward Neural Network



Given the above, a feedforward neural network can be considered as a stacked multilayer regression, because each node looks like a predictor or an outcome variable of regression analysis and the weights act as the regression coefficient. However, neural networks largely differ from linear regression due to the use of largely non-linear activation functions. The activation function is generally a non-linear mapping of one variable to another variable, which non-linearly transforms the linear combination of inputs. For example, as shown in Figure 6, the three inputs are first combined by the nine weights in the

same manner as linear regression, and then the three combined inputs are non-linearly transformed by an activation function, which produce the three hidden nodes. Likewise, the two output nodes are obtained through the non-linear transformation of the linear combination of the three hidden nodes using an activation function. There is a variety of activation functions. The *Sigmoid* function, or the *logistic* function, which transforms a real-valued variable to a variable ranging from 0 to 1, is widely used in deep learning models. This feature of *Sigmoid* activation is especially handy for the task of learning outcome modeling because the model is expected to make probabilistic predictions of learners' item responses. In addition to machine learning models, as mentioned earlier, the majority of psychometric measurement models use the *Sigmoid* activation to transfer the linear combination of model parameters to predicted probabilities of item responses. In this sense, due to the complexity of model structure and the flexibility of activation functions, deep learning-based models have the potential of representing other machine learning and statistical learning models (see the Chapter 2 of the book by Aggarwal [2018]).

The deep neural network can be considered as a feedforward neural network with more than one hidden layer. Mathematically, given a deep neural network, the output nodes $\mathbf{Y}$ can be predicted by the non-linear transformation of the combination of input nodes $\mathbf{X}$ as:

$$\mathbf{Y} = f_H(\mathbf{W}_H^T f_{H-1}(\dots f_2(\mathbf{W}_2^T f_1(\mathbf{W}_1^T \mathbf{X}))\dots)), \tag{15}$$

where $\mathbf{W}_1$ to $\mathbf{W}_H$ indicate the weights for the H neural network layers, and f_1 to f_H denote the activation function for each layer. The predictive capacity and the model complexity are controlled by the number of hidden layers and the number of hidden nodes in each layer, both of which are hyperparameters to be tuned for a deep neural network. For a simple classification problem, according to the recommendations by Goodfellow et al. (2016, p. 192) and Lippmann (1987), one or two hidden layers is sufficient for a neural network. However, to capture the complexity of training data, a deep neural network typically uses more than

one hidden layer. Regarding how to determine the optimal numbers of hidden layers and the hidden nodes for each hidden layer, for a simple model structure and a small training sample size, a k-fold cross-validation technique can be possibly used to optimize the model. However, in practice, given the complexity of model architecture and the large sample size, the k-fold cross-validation is infeasible and there are no rules of thumb for determining the two hyperparameters. Alternatively, it is suggested to configure the numbers of hidden layers and the hidden nodes by trial and error until satisfactory prediction accuracy is met (Goodfellow et al., 2016).

A major concern for constructing and training a deep neural network is how to prevent overfitting, which occurs when the model performs well on the training dataset but shows poor prediction performance on an external test dataset (e.g., test or validation datasets). The training dataset is used to optimize the model hyperparameters and learn the model weights, while the test dataset is used to evaluate the predictive capacity of the model, which is not allowed to be seen by the model in training. In deep learning, there are two categories of approaches to prevent or reduce overfitting: training the model with larger samples and reducing the model complexity. For the former approach, it is intuitive that a greater sample size means a greater coverage of the data variance. With larger samples, the model can learn a wider range of features and is more likely to account for the characteristics of the test samples, which contribute to satisfactory prediction performance on the test dataset. For the latter approach, there are two major ways to reduce the model complexity: "changing the number of adaptive parameters in the network" and "the use of regularization which involves the addition of a penalty term to the error function" (Bishop, 1995, p. 332). To put it simply, "changing the number of adaptive parameters in the network" means we can control the model structure stability by optimizing the numbers of hidden layers and hidden nodes for each layer. For the use of regularization, it means we can control the model

complexity by keeping the model weights small. Small weights indicate that model prediction would not change substantially given a big variance of input data, which contributes to satisfactory prediction performance on the test dataset. A widely used regularization method is weight decay, which makes the weights of less useful nodes as close as possible to zero, and the weights of influential nodes as small as possible. As such, only the influential nodes take effect in prediction for reducing overfitting. In addition to weight decay, regularization methods also include activity regularization, dropout, early stopping, and weight constraint, which serve the same purpose of penalizing the weights (see Goodfellow et al., 2016). The weights of a deep neural network can be learned through back propagation with optimizers based on gradient descent, such as RMSprop (Tieleman & Hinton, 2012), Adam (Kingma & Ba, 2014), and Adagrad (Duchi et al., 2011).

Recurrent Neural Networks

Recurrent neural networks (RNNs) are essentially neural networks with the capacity to model temporal information. That said, unlike the input data for deep neural networks, the input data fed into RNNs additionally involves a temporal dimension. For example, in the context of leaning outcome modeling, for the same skill and associated item, learners might take multiple learning opportunities to practice the skill. Therefore, learners' item responses are in a temporal form with an item response for each learning opportunity. Moreover, the item responses of earlier learning opportunities should affect those of later learning opportunities. In this case, RNNs can be used to model the temporal dependencies in the data.

Figure 7

Graphical Representation of Recurrent Neural Networks with Unrolled Form

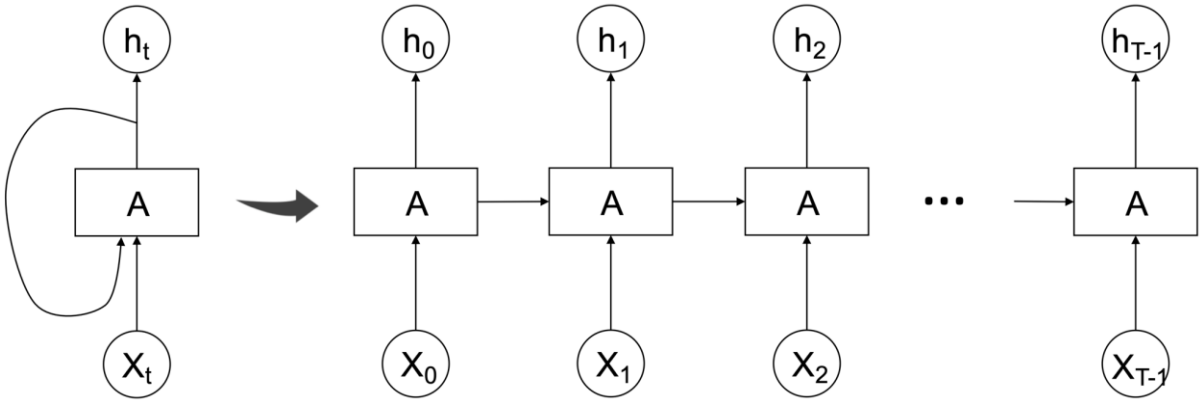


Figure 7 graphically presents the structure of an RNN and its unrolled form for each timestep. The RNN involves an input feature with *T* timesteps and outputs the prediction with *T* timesteps as well. Given multiple timesteps, the RNN allows information transmitted from one timestep to its subsequent timestep. When unrolled, the RNN looks like multiple copies of a conventional neural network with information passing from one network to a successor. In the RNN, the output of one timestep would be fed into its subsequent timestep as the input. This property of RNN enables that how the information at each timestep depends on each other can be learned. For example, if we want to predict the last word in the sentence "people have breakfast in the *morning*", each word can be modelled at one timestep in a sequential order in RNNs. In this case, the last word "morning" can be predicted by its very recent context information. However, if predicting "morning" needs sentences from much further back, which means the gap between relevant information for prediction and the place where it is needed is very large, RNNs fail to learn the model with faraway context information. This is due to RNN multiplying gradients or weights multiple times as it models dependencies between timesteps separated by many other timesteps, which leads to either vanishing or exploding gradients during backpropagation and in turn makes weights of early features unlearnable. Bengio et al. (1994) provides more details regarding this problem.

Fortunately, the long short-term memory (LSTM) networks, first introduced by Hochreiter and Schmidhuber (1997), can be used to address the long-term dependency problem due to vanishing or exploding gradients mentioned above. LSTM networks are widely used because they are tremendously powerful on a wide range of practical problems. In an LSTM network, a standard neural network layer is substituted by the LSTM cell blocks including four interacting layers. Figure 8 presents a diagram for an LSTM cell showing how LSTM works internally learning the feature information at the current timestep along with the output information produced by the last timestep. In the diagram, the lines transmit information from one timestep to another. Circles indicate pointwise operations like addition and multiplication. The four bottom rectangles indicate the four learned neural network layers. A line merging indicates concatenating vectors and a line forking indicates that a vector and its copy go for different directions. The top dashed horizontal line denotes the internal state of an LSTM network cell, which is the key idea behind LSTM networks. The cell state includes some linear interactions and conveys information changed or added by gates. The gate is a mechanism regulating and operating information from the current inputs and the outputs of the last timestep. The three gates of the LSTM network are constructed by a neural network layer with *Sigmoid* activation and a pointwise multiplication operation (see the rectangles with σ and an arrow directing to the multiplication signs in the diagram). From left to right in the diagram, the first gate is the forget gate, which determines what information of the previous cell state S_{t-1} should be remembered (the forget gate controls the output close to 1) and what should be forgotten (the forget gate controls the output close to 0). Specifically, the forget gate f_t is given by:

$$f_t = \sigma\left(W_f h_{t-1} + W_f x_t + b_f\right), \tag{16}$$

where W_f are the weights for the concatenation of the outputs of the last timestep h_{t-1} and the current features x_t; b_f denotes the bias; and σ indicates the *Sigmoid* activation. The forget gate enables information forgotten or remembered by $f_t * S_{t-1}$. The next step in the LSTM cell is to determine what information from h_{t-1} and x_t should be added to the cell state. This operation involves two components, the candidate values to be added to the cell state produced by h_{t-1} and x_t with *tanh* activation, and the input gate controlling which values should be added. Similar to the forget gate, the input gate i_t is given by:

$$i_t = \sigma(W_i h_{t-1} + W_i x_t + b_i). \tag{17}$$

The candidate value vector g_t generated by the *tanh* layer can be expressed as:

$$g_t = \tanh\big(W_g h_{t-1} + W_g x_t + b_g\big). \tag{18}$$

As such, the new information added to the cell state is obtained by $i_t * g_t$. Now we can update the old cell state S_{t-1} to the new cell state S_t by forgetting or remembering information in S_{t-1} and storing new information from the inputs, which can be expressed as:

$$S_t = f_t * S_{t-1} + i_t * g_t. \tag{19}$$

The last step is to determine what to output from the LSTM cell. The outputs from the LSTM cell are based on the cell state, regulated by the output gate. First, a *tanh* function is used to change the range of the cell state values as between -1 and 1. Then again, a *Sigmoid* function is applied for the output gate to generate the weights for controlling which values to output. The output gate can be expressed as:

$$O_t = \sigma(W_o h_{t-1} + W_o x_t + b_o). \tag{20}$$

and the final outputs are expressed as:

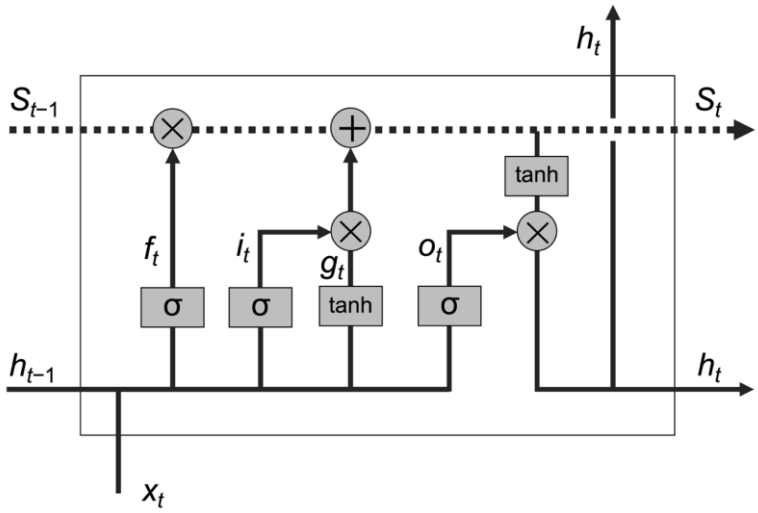
$$h_t = O_t * \tanh\,(S_t). \tag{21}$$

From what introduced above, the three gates contribute to the flexibility of LSTM networks by controlling the information of inputs, the information to be remembered or forgotten in the internal cell state, and the information of outputs. It is also worth noting that the vanishing or

exploding gradient problem is mitigated in LSTM networks because the partial derivatives of cell states involve no fast decaying factors. This can be seen from the equation of obtaining S_t, where its gradients drop the term of $i_t * g_t$ and only keep the term of f_t. The multiple multiplications of f_t however, unlike multiplying the gradients of RNNs, would not vanish or explode rapidly.

Figure 8

Graphical Representation of an LSTM Cell



Like deep neural networks, in practice, a deep learning model might involve multiple LSTM networks for capturing a greater degree of temporal complexity in the data. The number of LSTM layers and the number of the LSTM output dimensions (similar to the number of nodes for deep neural networks) are hyperparameters to be tuned for an LSTM-based model.

Deep Knowledge Tracing

In the context of educational data mining, a representative deep learning-based approach for learning outcome modeling is Deep Knowledge Tracing (DKT; Piech et al., 2015). Given a sequence of learners' item responses for multiple skills, in essence, DKT predicts a specific item response through learning the temporal dependencies between item responses prior to the current one based on RNNs. Figure 9 demonstrates a graphical

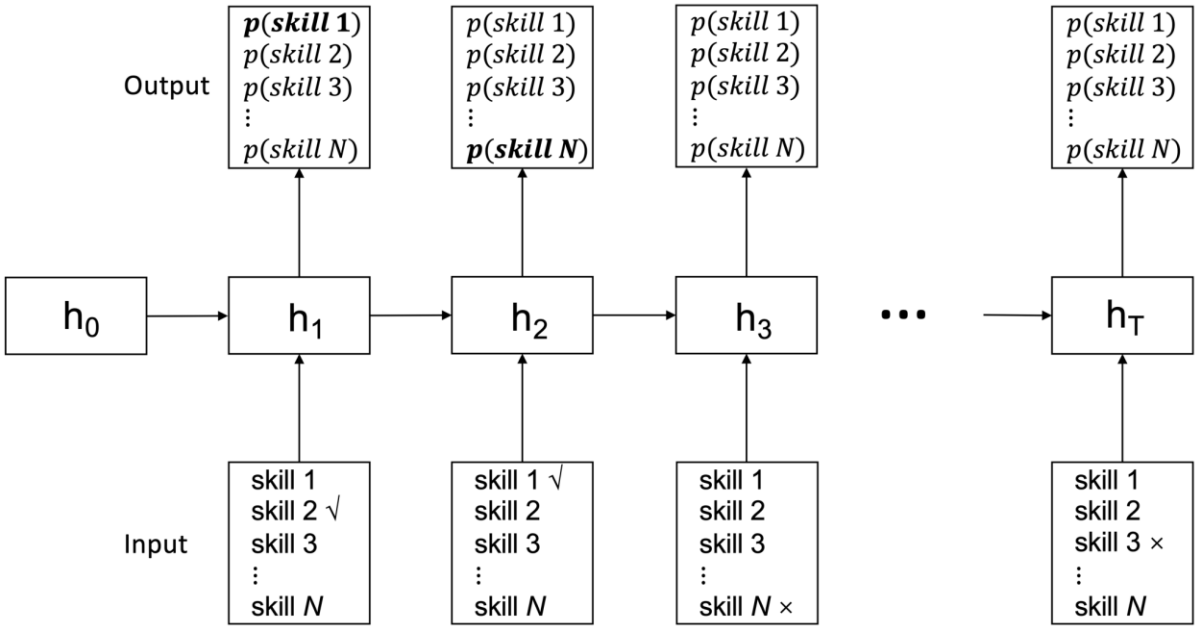
representation of DKT. In the diagram, the multiple opportunities of applying each skill are connected with an RNN for learning their temporal dependencies. The input data for the DKT framework is the information regarding which skills a learner accessed and what the outcomes of answering their associated items were (i.e., correct or incorrect). It should be noted that DKT also works at the item level. That said, the inputs for DKT can also be the items and associated responses. If modeling at the skill level, DKT does not recognize the differences between items measuring the same skill. In DKT, each item can be considered as a learning opportunity for practicing its associated skill. For example, for the first item response in the diagram, it shows that the learner practiced skill 2 and had a correct answer on an item measuring skill 2. Subsequently, the learner proceeded to practice skill 1 and answered an item measuring skill 1 correctly. Then the learner practiced skill *N* but incorrectly answered its associated item. The outputs of DKT are the predicted probabilities of correctly answering the items for each skill. For example, at the first timestep, the learner's item response for skill 2 is fed into DKT to produce his or her probabilities of getting items correct for each skill. Because at the second timestep the learner practiced skill 1, the first element of the vector of predicted probabilities can be used to infer his or her success likelihood of getting an item for skill 1 correct. In addition, for domain modeling, standard DKT can be used to model the relationships among skills or item-skill associations depending on skill- or item-level modeling.

After DKT was proposed, in recent years, several studies have examined the performance of DKT and extended the DKT framework for more complex learning outcome modeling problems. For example, Xiong et al. (2016) re-examined the performance of DKT with performance factors analysis and BKT as baselines on multiple datasets and found that DKT outperformed the baselines. Moreover, DKT was adapted to model open-ended item responses such as programming exercises (Wang et al., 2017), revised by introducing

regularized loss function to enhance the prediction consistency (Yeung & Yeung, 2018), and integrated with psychometric models such as IRT to improve its interpretability (Yeung, 2019).

Figure 9

Graphical Representation of Deep Knowledge Tracing



Other Deep Learning Approaches for Learning Outcome Modeling

In addition to DKT, in recent years, deep learning architectures are often incorporated in other approaches for learning outcome modeling. For example, as mentioned in the section "Additive Factors Model", the LSTM network can be incorporated with AFM to dynamically model learners' item responses for refining or learning from scratch the expert-specified item-skill associations (Pardos & Dadu, 2018). Moreover, based on deep learning architectures, the item-skill associations can be derived without learning learner product data. For example, Chaplot et al. (2018) proposed a framework named Cognitive Representation Learner to automatically extract the skills required by each item through learning the representations of item text or item content based on convolutional neural networks or RNNs. As stated by the authors, their framework is capable of discovering item-skill associations

without any learner product data, which is especially beneficial for items in ill-structured domains where data and human knowledge are both not available.

Contrast to learning outcome modeling without product data, another stream of research focused on how to exploit a variety of auxiliary information along with product data for enhanced learning outcome modeling. For example, based on the IRT framework, Cheng et al. (2019) proposed that item content and item-associated latent skills can be learned by deep neural networks and LSTM networks to automatically generate item difficulties, item discriminations and learners' latent ability levels. These learned item and learner parameters are then used to produce the predicted probabilities of correct item responses. Their framework was found to outperform conventional IRT models because more information is exploited for estimating the model parameters. Moreover, Su et al. (2018) developed a sequential modeling framework based on the LSTM network for predicting learners' item responses based on their history item responses. Particularly, their framework integrates a representation learning architecture for exploiting the item content associated with each item response, which contributes to higher predictive capacity in comparison with conventional approaches such as IRT, BKT, and DKT. Furthermore, deep learning techniques showed potential of detecting learners' affective states for learner modeling. Contrast to traditional affective detection approaches leveraging physical and physiological sensors, Botelho et al. (2017) developed a novel sensor-free affect detector based on multiple RNN variants to automatically recognize learners' affective states from their interactions with the system for learner modeling, which demonstrated higher prediction performance than conventional machine learning-based approaches.

Collaborative Filtering for Learning Outcome Modeling

As mentioned in the introduction, CF is a promising approach for learning outcome modeling examined by an increasing number of studies in recent years (e.g., Almutairi et al.,

2017; Desmarais & Naceur, 2013; Durand et al., 2015; Lan et al., 2014; Matsuda et al., 2015). CF is originally used for recommender systems, but its idea has been extended to address issues in other domains such as disease diagnosis (Shen et al., 2017), online learning (Wang & Yang, 2012), and social media analysis (Starbird et al., 2012). CF makes recommendations for a user on new items based on the fundamental assumption that if two users have similar behaviors on items (e.g., similar item responses, buying or watching decisions), their behaviors on other items are also similar (Goldberg et al., 2001). The CF approaches deal with a dataset to make recommendations in the following form: there are a set of items and a list of users, and each user has a value on partly or all of the items. Represented as a user by item matrix, the dataset looks like a sparse matrix (i.e., a matrix with many missing entries). Those missing entries are the values to be predicted by the CF approaches. It should be noted that the values can be both explicit and implicit. Explicit values refer to quantified item responses such as ratings ranging from 1 to 5 or scores of 0 and 1. Implicit values refer to unquantified item responses such as actions of buying an item, watching a movie, or clicking an item.

According to the review by Su and Khoshgoftaar (2009), there are three categories of CF approaches: memory-based CF, model-based CF, and hybrid recommenders. The memory-based CF approach makes recommendations through computing the similarity between users or items. For example, given the user-based top-N recommendation algorithm, a user's predicted rating on an item is simply the aggregated ratings on the item provided by some other users who are most similar to the user in the dataset. Likewise, given the item-based top-N recommendation algorithm, an item's predicted rating by a user is simply the aggregated ratings by the user on some other items which are most similar to the item in the dataset. The key idea of memory-based CF approaches is to quantify the similarity between users and items, which can be calculated by a variety of measures (see Su & Khoshgoftaar,

2009). Memory-based CF approaches are easy to implement, but suffers disadvantages such as depending on human ratings, not working well for sparse data, cold-start problems, and limited scalability (Wang et al., 2014; Zhang et al., 2020). The model-based CF approaches can be used to address these disadvantages as they are developed based on a variety of data mining and machine learning models such as Bayesian networks, matrix factorization, clustering algorithms, and regression models (Aggarwal, 2016; Mehta & Rana, 2017). Compared with the memory-based CF approaches, the model-based CF approaches are more complex for computation, but they are more scalable, more capable of dealing with sparse data, and more accurate in prediction. The hybrid recommenders are combinations of CF approaches and content-based recommenders (Dong et al., 2017; Kumar & Fan, 2015; Zhang et al., 2017). The content-based recommenders make recommendations based on the analysis of a variety of contextual information and item content but are not scalable and suffer the cold-start problem. As such, the hybrid recommenders integrate the advantages of CF approaches and the use of contextual information for elevated prediction accuracy.

In the next section, a widely used model-based CF approach, matrix factorization, is introduced, given its great popularity in recommenders systems. In the following, “users” will be replaced by “learners” when describing the technical details because of the context of the current research topic.

Matrix Factorization

Matrix factorization is exceptionally effective for building recommender systems (Koren et al., 2009). In essence, matrix factorization deals with a sparse high dimensionality learner-item matrix with missing responses by introducing a set of latent factors for dimensionality reduction. The association between latent factors and learners and the association between latent factors and items are two lower-dimensionality matrices factorized from a complete or incomplete learner-item matrix. Mathematically, through matrix

factorization, an item response matrix $R \in \mathbb{R}^{m \times n}$ of m learners and n items can be decomposed into two low-rank matrices $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{n \times k}$:

$$R \approx UV^T. \tag{22}$$

The former is a learner-by-factor matrix representing the learner-skill associations and the latter is an item by factor matrix representing the item-skill associations. The dimension k indicates that there is a total of k latent factors modeled by matrix factorization.

In the matrix U, the ith row $\overrightarrow{u_i}$ indicates the associations between a learner and the k latent factors; in the matrix V, the jth row $\overrightarrow{v_j}$ indicates the associations between an item and the k latent factors. Therefore, an entry of the item response matrix, r_{ij}, can be approximately recovered by the dot product of the learner factor $\overrightarrow{u_i}$ and the item factor $\overrightarrow{v_j}$:

$$r_{ij} \approx \overrightarrow{u_i} \cdot \overrightarrow{v_j}. \tag{23}$$

Estimating the lower dimensionality matrices in matrix factorization can be also formulated as a maximum likelihood estimation problem like most machine learning models. That is, we seek two lower-dimensionality matrices U and V that minimize the differences between the original item response matrix entries and the predicted item response matrix entries given by equation 23. The gradient descent algorithms can be used for optimization to solve the problem. Similar to other machine learning or deep learning models, as mentioned in the section of deep neural networks, the regularization technique can be used to decay large latent factor values to prevent or reduce overfitting in matrix factorization. For example, the L_2 regularization technique can be applied to the matrices U and V, which changes the problem as minimizing the following objective function:

$$\arg \min_{U,V} J = \frac{1}{2}\|R - UV^T\|^2 + \frac{\lambda}{2}\|U\|_F^2 + \frac{\lambda}{2}\|V\|_F^2, \tag{24}$$

where λ denotes the regularization weight controlling the degree to which the latent factor values are decayed, and $\|\bullet\|_F^2$ represents the Frobenius norm.

The above formulation of matrix factorization stands for its most basic form, which is not typically used in practice. This is due to that given no model constraints, the solution of lower dimensionality matrices U and V are hard to be fixed, which leads to an ill-posed problem. Therefore, various model constraints per application domain are typically imposed on matrix factorization to enhance prediction performance and interpretability. Some representative model constraints for matrix factorization are non-negativity, orthogonality, and sparseness of model weights (e.g., Ding et al., 2006; Hoyer, 2004; Lee & Seung, 2001). For example, the L_1 regularization technique is typically used to encourage the matrix sparseness. However, regardless of model constraints, the matrix factorization-based approaches are generally developed to stably estimate meaningful lower dimensionality representations of users/learners and items in terms of how they are connected with a limited number of latent factors.

Moreover, in this dissertation, the matrix factorization approach, in addition to being popular, is mainly described for the purpose of demonstrating how the CF approaches make inferences about item responses based on learning the lower dimensionality representations of learners and items. How learner and item latent representations contribute to the predicted probabilities of item responses is not necessarily modeled by the matrix multiplication. For example, the interaction between the two representations can also be learned through deep neural networks. More details about other relevant work will be given in the next section.

Collaborative Filtering-Based Approaches for Learning Outcome Modeling

Overall, the application of CF in education is still in its infancy. In the past ten years, a growing number of studies are advancing the field by proposing a variety of CF-based approaches for analyzing learner data. Particularly, since matrix factorization can represent items with latent factors, the majority of these approaches were developed with a focus on learning from scratch or refining item-skill associations based on learner data. For example,

several studies (e.g., Desmarais, 2012; Desmarais & Naceur, 2013; Durand et al., 2015; Matsuda et al., 2015; Sun et al., 2014) used the CF framework to evaluate expert-specified Q-matrices or automatically generate data-driven Q-matrices. In their work on evaluating the predictive capacity of expert-specified Q-matrices, Durand et al. (2015) proposed an evaluation method based on cross validation and found that their approach was capable of efficiently and quickly evaluating expert-specified Q-matrices without complex computation of multiple model parameters as in sophisticated CDMs. Desmarais (2012) examined the potential of non-negative matrix factorization for recovering the Q-matrix and found that it is a highly effective approach for deriving the Q-matrix given the assumption of skill independence, but it is less effective if the values of learner, item and skill parameters vary a lot in the data. Desmarais and Naceur (2013) compared the performance between the expert-specified Q-matrix and the data-drive Q-matrix by matrix factorization and found that they shared similar patterns between item-skill mappings, but the matrix factorization approach lightly outperformed the expert-specified Q-matrix. Similarly, Sun et al. (2014) found that their proposed approach based on the Boolean matrix factorization could successfully recover the original Q-matrix from learner data. In the context of large-scale online courses, compared with the expert-specified Q-matrix, the approach developed by Matsuda et al. (2015) based on the matrix factorization framework was found to be faster, more predictive, and more scalable for discovering item-skill associations.

To sum up, the previous work on learning the Q-matrix from learner data generally show that CF approaches, especially matrix factorization, could be successfully used for domain modeling and they had the potential of outperforming the expert knowledge in some contexts (e.g., Desmarais & Naceur, 2013; Matsuda et al., 2015). However, researchers also indicated that matrix factorization-based approaches such as alternating least square, non-negative matrix factorization, and Boolean matrix factorization still showed limited capacities

to learn the expert-specified Q-matrix from scratch (Desmarais, 2011; Desmarais and Naceur, 2013), and they could be more effectively used to refine expert-specified Q-matrices.

The above studies mainly focused on learning the Q-matrix from the learner data in comparison with the original expert-specified Q-matrix. Studies also strived to learn item-skill associations from scratch leveraging the idea of matrix factorization. A representative work is the sparse factor analysis algorithm proposed by Lan et al. (2014), which is capable of learning item- and learner-skill associations and item difficulties from binary-valued item responses without any auxiliary information. Their approach showed strong predictive capacity and interpretability. Moreover, another stream of research emphasized the usefulness of contextual information in learning outcome modeling. For example, based on matrix factorization and tensor factorization under the CF framework, Almutairi et al. (2017) proposed three methods to model students' grade data and found that the time when a learner was graded was helpful for improving the prediction performance. Similarly, Sahebi et al. (2016) used learners' interactions with the learning resources to model their learning progress based on the tensor factorization approach and found that their approach was significantly more predictive of learner performance than BKT and another tensor factorization approach. In addition, the sequential modeling approach based on tensor factorization proposed by Thai-Nghe et al. (2012) was successfully used to predict learners' future item responses based on learning history item responses.

Deep Learning-Based Collaborative Filtering

In recent years, informed by the deep learning advances, more and more studies have focused on incorporating the CF framework with deep learning architectures to improve model predictive capacity. Despite not being proposed specific to learning outcome modeling, these deep learning-based CF approaches are very promising in learner modeling and domain modeling. Most of the deep learning-based CF approaches utilize a deep neural

network architecture to learn more complex or non-linear interactions between learner and item representations for prediction. For example, in the "two-stream neural network architecture for matrix completion" proposed by Nguyen et al. (2018), rows and columns of a user-item matrix, which represent user and item vectors, are separately fed into multiple neural network layers to learn more effective item and learner representations, which can be extended to new users and new items. In the deep CF framework proposed by Li et al. (2015), the item and user representations are learned through marginalized denoising stacked auto-encoders based on additional sources of information on users and items, which are in turn incorporated into the matrix factorization framework for prediction. In the neural CF framework developed by He et al. (2017), solely based on the user-item rating matrix, the concatenation of user and item representations (user and item embeddings) is fed into multiple neural network layers to learn the non-linear interactions between users and items, which are incorporated into a generalizable matrix factorization framework for prediction. These CF approaches based on deep learning architectures generally outperformed other state-or-art methods in terms of prediction performance.

In summary, compared with other types of CF approaches, deep learning-based CF approaches are more capable of capturing the complexity of interactions between learners and items in affecting item responses. This means that the model predictive capacity benefits from the finer-grained representations learned by deep learning as more information can be extracted to know about learners and items for prediction. Moreover, deep learning architectures are exceptionally effective for learning additional information about learners and items, such as learners' background information, item content, and potentially, the process data associated with item responses. Leveraging the representation of these additional information, the CF approaches can be improved in terms of two aspects. First, they can predict missing responses with higher accuracy because the system knows items and learners

better. Second, with respect to learner modeling and domain modeling, the learner and item representations can be refined as well through learning additional information. For example, item-skill associations can be more accurately estimated through learning learners' actions and time durations for answering each item. Methodologically, adding more information in model learning can be considered as a regularization technique which makes the weights of item-skill associations more stable, interpretable, and generalizable. This is a desirable feature for learner modeling and domain modeling. Unfortunately, very few, if any at all, established deep learning-based CF approaches were developed specifically for learning outcome modeling and process data learning. Investigations of deep learning-based CF approaches for learning outcome modeling are acutely needed.

Process Data Analysis for Learning Outcome Modeling

The previous sections presented a comprehensive survey of existing mainstream approaches for learning outcome modeling. Despite a methodological lens, the introduction to the approaches emphasizes the importance of exploiting all the available information about learners and items in addition to explicit item responses for learning outcome modeling. In this section, the chapter reviews some pioneering work revealing the potential of process data for learning outcome modeling.

As mentioned, there exist a number of case studies showing how to analyze process data to inform learning in the settings of CBA. For instance, Greiff et al. (2015) analyzed the process data of one question on complex problem solving in PISA 2012 for identifying learners' problem-solving strategies. They extracted a set of frequency-related and time-related features from the process data and examined how these features predicted learners' problem-solving success. Notably, they identified a dominant strategy for solving the question. However, their analyses were conducted in an exploratory fashion with only one item, which is not scalable and extendable in other settings. With the data of an item from the

same assessment, Liu et al. (2018) proposed to use a modified multilevel mixture IRT model to analyze learners' process data, which identified different latent classes of problem-solving strategies and estimated learners' abilities at both the process and item levels. Their approach was also examined with the data of one item and showed limited generalizability. The PISA dataset was also analyzed by the event history analysis model proposed by Chen et al. (2019). Their approach was developed to model the problem-solving process with the aim of predicting both the remaining time a learner needs for completing the item and the final problem-solving outcomes (success or failure). However, their approach suffers the limitation of single-item analysis as well, which cannot be well extended to multiple-item analysis. Similarly, Shu et al. (2017) proposed a Markov-IRT model to extract features from learners' problem-solving processes as evidence for psychometric measurement. However, the Markov property assumed by their approach limits the temporal dependencies in problem solving between two consecutive actions.

More recently, Tang et al. (2019) proposed a more generalizable approach for extracting informative features from learners' action sequences in solving a problem based on the sequence-to-sequence autoencoder. The learned latent features are indicative of how learners attempt a problem, which can be used for subsequent statistical or machine learning analysis. Essentially, their approach is representation learning of action sequences. However, it cannot deal with multiple items simultaneously and fails to model the time information. Moreover, in terms of learning outcome modeling or other predictive analyses, a sophisticated model is still needed to connect representation learning of action sequences with other model architectures.

In summary, the existing approaches for process data analysis were mainly developed and examined in specific contexts with a single item. Moreover, they heavily rely on the assumptions of statistical or psychometric models and require human-specified rules, which

limits their scalability and generalizability. In terms of learning outcome modeling, none of these approaches is capable of modeling item responses with process data at a large scale across multiple items.

An Overview of Approaches for Learning Outcome Modeling

Table 2 presents a summary of the mainstream approaches for learning outcome modeling used in both communities of psychometric measurement and educational data mining. In general, psychometric measurement models such as IRT and cognitive diagnosis models require strong assumptions about how latent skills affect item responses. Moreover, they are limited in dealing with highly unstructured data which is most accessible in the context of digital learning. The Bayesian family approaches require complex computation of great amounts of conditional probabilities given multiple items and skills, which is challenging for computational resources. Moreover, constructing Bayesian networks requires human knowledge on the links between variables. Although the DKT approach is very promising, how it can be used to tackle unstructured process data in addition to structured item responses is still under-investigated. The AFM approach requires pre-specification of item-skill associations and it models the data at the skill level. In general, deep learning approaches with a well-designed model architecture and well-tuned model hyperparameters are very effective in learning outcome modeling, shown by their higher prediction capacities, stronger generalizability, and higher scalability than conventional approaches. Particularly, this chapter emphasizes the potential of deep CF approaches for learner and domain modeling because learner and item differences can be strongly represented by latent factors. Finally, this chapter reviews the existing work on process data analysis and advocates its potential for enhanced learning outcome modeling.

To sum up, the major gaps in the literature on learning outcome modeling are:

- Most sequential modeling techniques for tracing skill mastery and predicting item responses model the product data at the skill level and fail to address the differences between items.
- The vast majority of existing learning outcome modeling approaches fail to account for the process data.
- The existing approaches for process data analysis are not generic and scalable.

The dissertation strives to bridge the aforementioned gaps leveraging the promising features of CF approaches and deep learning architectures. The CF framework is used to account for learner and item differences at a fine-grained level, and the deep learning framework is used to capture the complexity of both product data and process data. Therefore, the dissertation proposes three generic modeling frameworks incorporating deep learning-based CF with process data learning to improve the accuracy of learner modeling as well as to discover item-skill associations from scratch.

Table 2

A Summary Table of Key Approaches for Learning Outcome Modeling

Approach	Temporal Modeling	Input Data	Learner Modeling	Domain Modeling	Multiple Skills	Multiple Items	Major Challenges
CTT	No	Product	Yes	No	No	Yes	Item and learner dependent; untestable assumptions; high measurement error; limited use in the context of CBAs
IRT	No	Product	Yes	No	No	Yes	Unidimensionality; strong assumptions; requires structured and complete data; no temporal modeling
Cognitive Diagnosis	No	Product	Yes	No	Yes	Yes	Hard to specify accurate Q-matrices; strong assumptions; requires structured and complete data; no temporal modeling
Bayesian Network	No	Product	Yes	No	Yes	Yes	High demands on computational resources; requires expert knowledge on model construction; no temporal modeling
DBN	Yes	Product	Yes	No	Yes	Yes	High demands on computational resources; requires expert knowledge on model construction
BKT	Yes	Product	Yes	No	No	Yes	Only models one skill; skill-level modeling without item parameters
AFM	Yes	Product	Yes	No	Yes	Yes	Hard to specify accurate Q-matrices; skill-level modeling without item parameters
DKT	Yes	Product	Yes	Yes	Yes	Yes	Deals with structured item responses only
Deep Learning	Yes	Product Process	Yes	Yes	Yes	Yes	Requires sophisticated design of model architecture; requires large amounts of data; requires large amounts of computational resources
Matrix Factorization	Yes	Product	Yes	Yes	Yes	Yes	Limited in recovering item-skill associations; limited interpretability of model weights
Deep Learning-Based CF	Yes	Product Process	Yes	Yes	Yes	Yes	Requires sophisticated design of model architecture; requires large amounts of data; requires large amounts of computational resources
Process Data Analysis	No	Process	No	No	No	No	Exploratory; mainly works for one item; limited generalizability and scalability

Note. The approaches summarized in the table mostly refer to their standard forms. Their variants might have different features with respect to the indicators in the table. Domain modeling refers to estimating or refining item-skill associations.

Research Problems

Specifically, the dissertation approaches the research objective by addressing the following three specific research problems:

1. **Sequential modeling of product data (i.e., item responses)**. As mentioned previously, approaches for learning outcome modeling can be categorized as sequential modeling and non-sequential modeling. For non-sequential modeling (e.g., standard IRT models), item responses are assumed to be conditionally independent from each other, which means learners' history item responses have no influence on their current and future ones. In addition, learners' mastery levels of latent skills are assumed to be constant throughout the assessment. However, for sequential modeling (e.g., BKT and DKT), the temporal dependencies between history item responses are modelled, which are assumed to affect learners' current and future item responses. In addition to predicting learner future item responses, sequential modeling of item responses in this work also discovers item-skill associations without expert information.
2. **Learning outcome modeling with process data**. The dissertation also addresses how to improve learning outcome modeling by exploiting process data. As mentioned, the process data carries more information regarding how learners attempt a problem, which can be used to improve the model prediction accuracy and the interpretability of item- and learner-skill associations.
3. **Sequential modeling of product and process data**. Informed by the first two research problems, the third research problem is to sequentially model both product data and process data. More specifically, for this research problem, the dissertation aims to improve the accuracy of predicting future item responses and discover item-

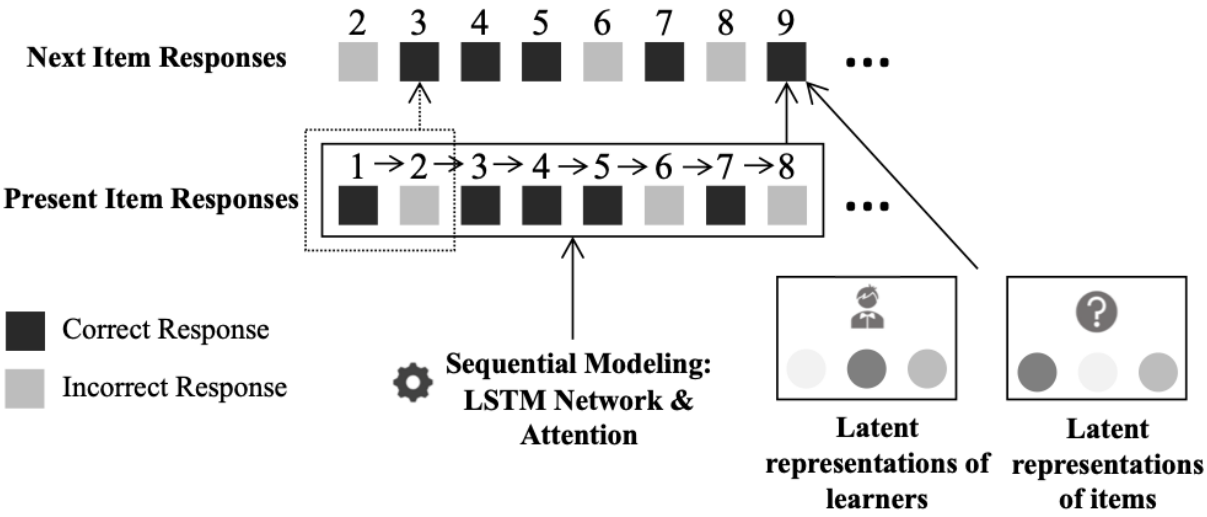
skill associations from the scratch by modeling learners' history item response sequences along with associated actions and time durations.

Proposed Approaches

The dissertation proposes three models to address the three research problems, which are all based on the CF framework.

Figure 10

Simplified Diagram of SDCF

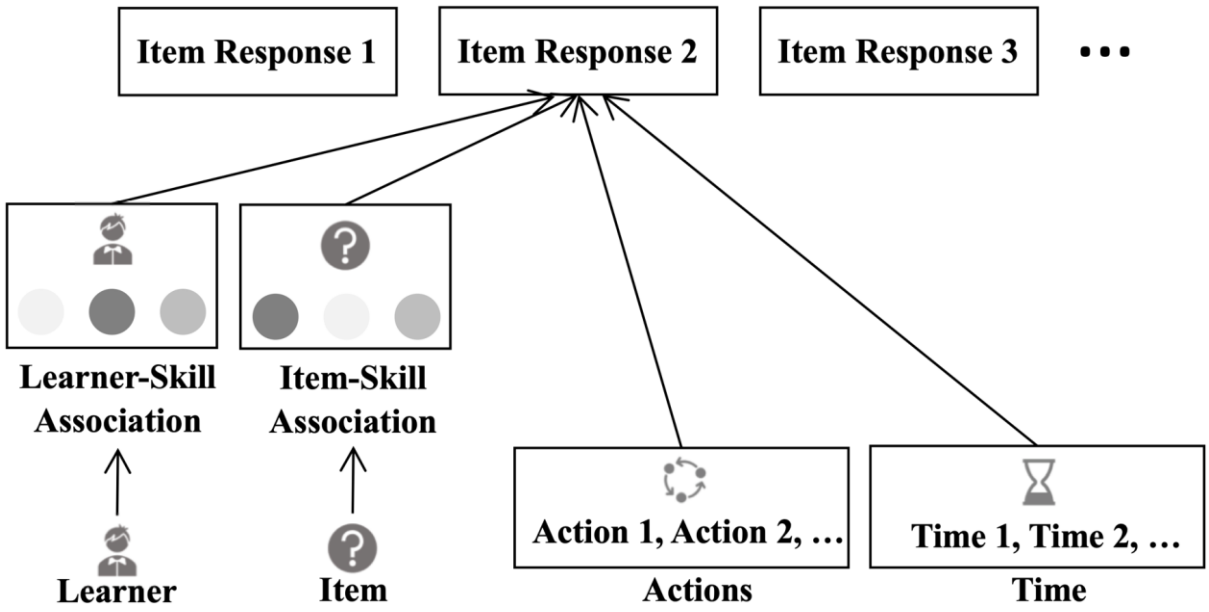


1. SDCF: Sequential Deep Collaborative Filtering Model. The SDCF model adopts the LSTM network for sequentially modeling of history item responses. For each item response within a learner's item response sequence, the probability of the individual correctly solving the current item is predicted by all the prior item responses. As such, compared with earlier item responses, the later item responses can be modelled with more information. Figure 10 presents a simplified diagram of SDCF (more details are given in Chapter 3). It can be seen that predicting response on item 9 requires the response sequence from item 1 to item 8, while predicting response on item 3 only requires responses on items 1 and 2. The temporal dependencies between present item responses are modelled by the LSTM network with the attention mechanism. Moreover, the model also discovers the item clusters in terms

of their associations with latent skills. In summary, SDCF is a sequential modeling technique built based on the LSTM network and the CF framework.

Figure 11

Simplified Diagram of LogCF



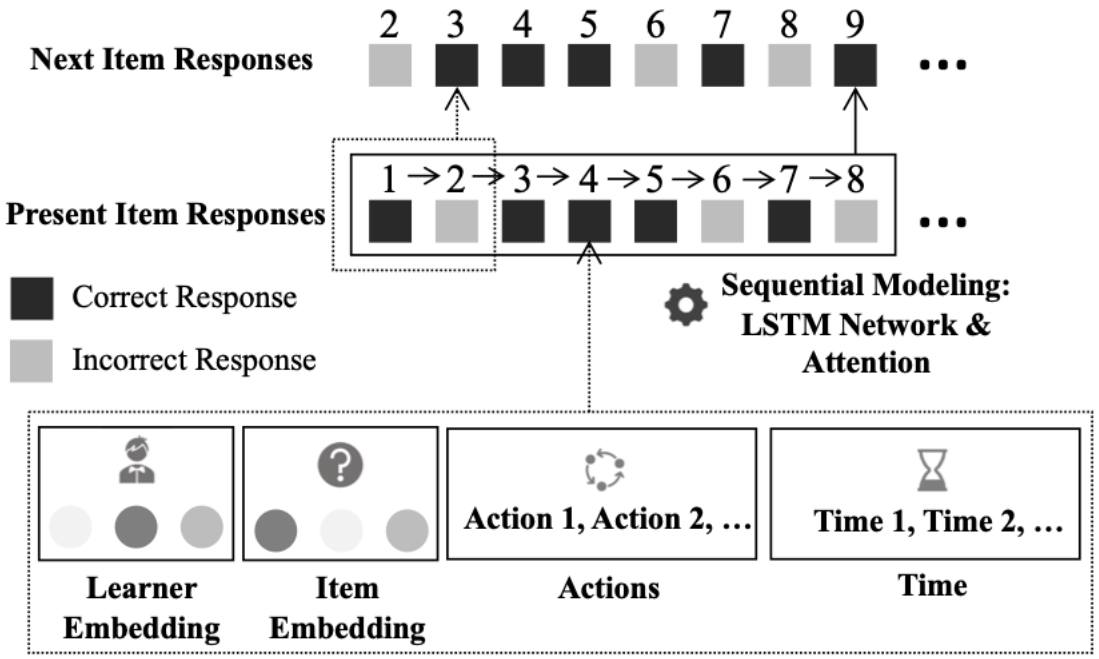
2. LogCF: Deep Collaborative Filtering with Process Data. To address the second research problem, LogCF is developed for learning outcome modeling of learning process data (see Figure 11). LogCF models item responses at the entry level, which means no temporal dependencies between item responses are involved. Specifically, for the prediction of a learner's item response on an item, the model learns the learner's representation (i.e., learner-skill associations) and the item's representation (i.e., item-skill associations) as intermediate inputs. Moreover, the model also learns the representations of the learner's actions and time durations associated with the item response, which are used as another set of intermediate inputs. These intermediate inputs are then used for the final prediction. In LogCF, the representations of raw data work and the final prediction are processed through deep learning techniques, such as embedding, LSTM networks, and multiple neural network layers (more details are given in Chapter 4).

3. LogSDCF: Sequential Deep Collaborative Filtering with Process Data.

Informed by SDCF and LogCF, LogSDCF is developed to account for both temporal dependencies between item responses and process data (see Figure 12). As such, for each item response, the model learns the representations of the learner, the item, the actions, and the time durations. These representations are integrated and modelled with the LSTM network with the attention mechanism for learning the temporal dependencies. The prediction of an item response is based on all its prior item responses and associated actions and time durations. Compared with SDCF, LogSDCF is supposed to result in improved prediction accuracy since additional information of process data is used.

Figure 12

Simplified Diagram of LogSDCF



Chapter 3 SDCF: Sequential Deep Collaborative Filtering Model

The purpose of this chapter is to address the first research problem by proposing a CF-based general framework predicting learners' future item responses based on their history item responses. Given the nature of sequential modeling, the model proposed in this chapter, SDCF, adopts the LSTM network to address the issue of learning temporal information of item responses. To capture higher complexity of learner-item interactions, SDCF integrates learner and item latent representations through a deep neural network architecture for prediction. Moreover, to discover how items are associated with each other in terms of latent skills, SDCF adopts the self-attention mechanism to learn the relevance of different items from scratch.

The following sections start with the problem formulation, followed by an introduction to the SDCF architecture.

Problem Formulation

Suppose that a hypothetical assessment with n items measures k latent skills and m independent learners take the assessment. Each learner's item responding process can be denoted as $\mathbf{R}_i = \{(\mathbf{m}_i, \mathbf{n}_1^i, R_1^i), (\mathbf{m}_i, \mathbf{n}_2^i, R_2^i), \dots, (\mathbf{m}_i, \mathbf{n}_T^i, R_T^i)\}$, where $\mathbf{m}_i$ denotes the identification of the learner, $\mathbf{n}_t^i$ denotes the item $\mathbf{n}_t$ responded by learner $\mathbf{m}_i$ at the tth timestep, and R_t^i denotes the corresponding item response result (correct/incorrect). If learner $\mathbf{m}_i$ correctly solves item $\mathbf{n}_t$, $R_t^i = 1$, otherwise $R_t^i = 0$. Learners and items are denoted by $\mathbf{m}_i$ and $\mathbf{n}_t$ instead of their subscripts i and t because learners and items can be characterized with various information. In SDCF, $\mathbf{m}_i$ and $\mathbf{n}_t$ are simply learner and item identifications, which are embedded as learner and item latent representations. However, $\mathbf{m}_i$ and $\mathbf{n}_t$ can be extended to indicate other learner and item features such as learners' background profiles and item texts, if a sophisticated model is devised. Having the item responding process $\mathbf{R}_i$ of each learner across the first T item response opportunities, our goal is to learn a model $\mathcal{M}$ which is

capable of predicting $\hat{R}^i_{T+1}$ on the next item $\mathbf{n}^i_{T+1}$ at the timestep $T+1$. In the meantime, the model is capable of discovering item-skill associations based on the relevance between items.

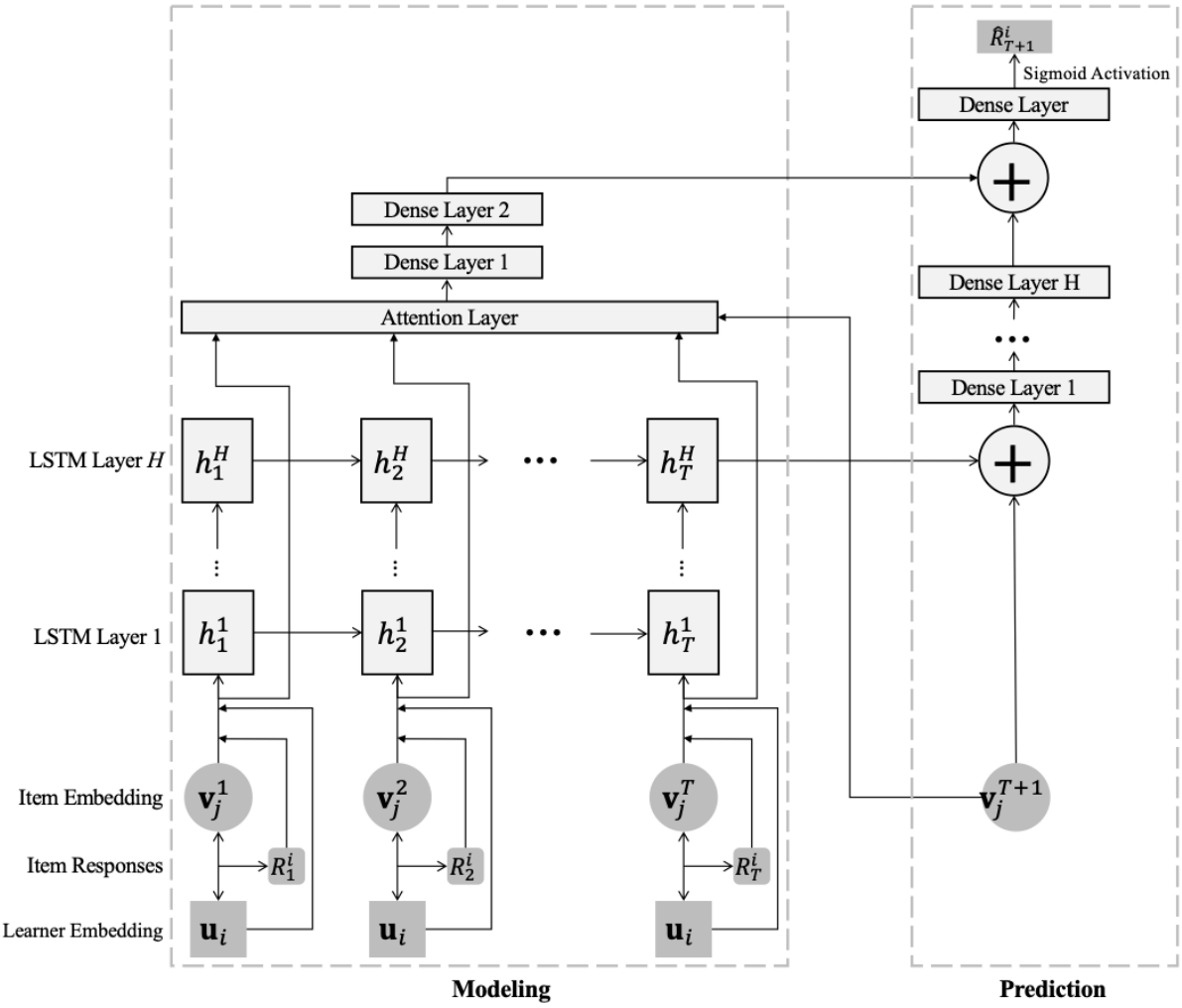
The proposed model SDCF is analogous to the knowledge tracing models such as BKT and DKT, serving the purpose of predicting future item responses based on modeling learner history item responses. The major difference of SDCF from them is that it is an item-level modeling technique accounting for the differences between items and the differences between learners. However, BKT and DKT label each item as its associated skill, requiring pre-specification of item-skill associations. In other words, they are not capable of utilizing item-level information.

Modeling Process of SDCF

Figure 13 presents the graphical representation of SDCF with its two main architectural components: an architecture for modeling the history of item responses and an architecture for predicting future item responses.

Figure 13

Graphical Representation of SDCF



Item and Learner Embedding

Given the raw data fed into the model, which is each learner's item responding process $\mathbf{R}_i = \{(\mathbf{m}_i, \mathbf{n}_1^i, R_1^i), (\mathbf{m}_i, \mathbf{n}_2^i, R_2^i), \ldots (\mathbf{m}_i, \mathbf{n}_T^i, R_T^i)\}$, the modeling process of SDCF first learns latent representations of learners and items based on their identifications $\mathbf{m}_i$ and $\mathbf{n}_t$. Because learner and item identifications are categorical variables, the model first converts them to sparse binary vectors by one-hot encoding. For instance, for a dataset with 100 unique learners, each learner can be represented as a 100-dimensional vector. In each vector, of the 100 dimensions, in one-hot encoding, only one dimension is valued at 1 and all other dimensions are valued at 0, which indicates the unique representation of one learner.

Likewise, each item can also be represented as a sparse vector. However, one-hot encoding is too sparse for learning, the model thus converts the sparse item and learner vectors to dense vectors with a dimension of the number of latent factors. For example, given 100 latent factors, each item and each learner can be represented as a 100-dimensional vector with each dimension indicating one latent factor. As such, given k latent skills, the model stacks an embedding layer on the input layer of learner and item identifications to project them onto a k-dimensional dense vector, which produces the learner latent representation $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$ and the item latent representation $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_n]$. Compared with one-hot encoding, embedding largely reduces the dimensions of item and learner representations and makes them more compact.

Concatenation of Embeddings and Item Responses

After item and learner embedding, the next step of the modeling process of SDCF is to concatenate the three types of inputs — the item embeddings $\mathbf{v}_j$, the learner embeddings $\mathbf{u}_i$, and the item responses R_t^i — for sequential modeling. The model first concatenates learner and item embeddings. Since both learner and item embeddings have k dimensions, after concatenation, the k-dimensional item and learner embeddings are combined as a $2k$-dimensional embedding vector, $\mathbf{e}_{ij}$. Subsequently, the model combines the concatenated embedding vector $\mathbf{e}_{ij}$ with the item response R_t^i at timestep t. Because a correct and an incorrect item response reflect different states of learners' item responding process, their different effects in modeling need to be accounted for. The model, therefore, extends the item response R_t^i to a feature vector $\mathbf{0} = (0,0, \ldots,0)$ with the same $2k$ dimensions of the concatenated embedding vector $\mathbf{e}_{ij}$, which is then concatenated with $\mathbf{e}_{ij}$ to produce a final concatenated vector $\mathbf{e}_{ij}^t$ as:

$$\mathbf{e}_{ij}^{t} = \begin{cases} [\mathbf{e}_{ij} \oplus \mathbf{0}] & \text{if } R_t^i = 1 \\ [\mathbf{0} \oplus \mathbf{e}_{ij}] & \text{if } R_t^i = 0 \end{cases}, \tag{25}$$

where $\oplus$ denotes the concatenation operator.

Deep LSTM Network Architecture for Sequential Learning

After the concatenation of embedding vectors and item responses, the model feeds the concatenated input $\mathbf{e}_{ij}^{t}$ into a deep learning architecture of multiple LSTM network layers for learning the temporal dependencies between item responses. Within an LSTM network layer, each item responding input $\mathbf{e}_{ij}^{t}$ at the tth timestep has a hidden state h_t, which is recurrently updated with the previous hidden state h_{t-1}:

$$\begin{aligned} f_t &= \sigma\left(W_f\left[h_{t-1}, \mathbf{e}_{ij}^{t}\right] + b_f\right), \\ i_t &= \sigma\left(W_i\left[h_{t-1}, \mathbf{e}_{ij}^{t}\right] + b_i\right), \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tanh\left(W_C\left[h_{t-1}, \mathbf{e}_{ij}^{t}\right] + b_C\right), \\ o_t &= \sigma\left(W_o\left[h_{t-1}, \mathbf{e}_{ij}^{t}\right] + b_o\right), \\ h_t &= o_t \cdot \tanh(C_t), \end{aligned} \tag{26}$$

where f_t, i_t and o_t indicate the forget gate, the input gate, and the output gate of an LSTM cell respectively, C_t denotes the hidden state at the t^{th} timestep, σ denotes the *Sigmoid* activation function and $\tanh$ denotes the hyperbolic tangent activation function. In addition, the weights and bias of the forget gate, the input gate and the output gate are represented by W_f and b_f, W_i and b_i, and W_o and b_o respectively. As mentioned, the three gates control the information inputted by the cell, the information remembered or forgotten in the cell state, and the information outputted by the cell, which makes the LSTM network very flexible and successful in modeling temporal dependencies.

Equation 26 only depicts the modeling process within one LSTM network layer. Two or more LSTM network layers can be stacked in the same model to deal with greater data complexity. Given multiple LSTM network layers, the output sequence of the last layer $\mathbf{S} =$

$\{s_1^i, s_2^i, \ldots, s_T^i\}$ can be considered sequential learner-item interaction over the past T timesteps, which are recurrently updated in the item responding process. The number of LSTM network layers and the number of hidden nodes (i.e., output dimension) for each LSTM network layer are two hyperparameters to be tuned in training.

To predict the next item response based on learner-item interaction over the past T timesteps, the output s_T^i produced by the deep LSTM network architecture is then concatenated with the embedding vector of the next item at timestep $T+1$, $\mathbf{v}_j^{T+1}$, and fed into a deep neural network architecture with multiple layers. Formally, it can be stated as:

$$D_{T+1}^i = f_H(\mathbf{W}_H^T f_{H-1}(\ldots f_2(\mathbf{W}_2^T f_1(\mathbf{W}_1^T \begin{bmatrix} s_T^i \\ \mathbf{v}_j^{T+1} \end{bmatrix})) \ldots)), \quad (27)$$

where $\mathbf{W}_1$ to $\mathbf{W}_H$ indicate the neural network weights for the H neural network layers, and f_1 to f_H represent the activation function applied in each neural network layer. The output of the deep neural network architecture, D_{T+1}^i, indicates the learned interaction between the current item for prediction and the past item responding process. The number of neural network layers and the number of nodes for each layer are two hyperparameters to be tuned in training.

Self-Attention Mechanism

In addition to LSTM networks, SDCF also uses a self-attention layer (Vaswani et al., 2017) to further model the relevance of the current item for prediction with the past solved items. An attention layer involves three inputs: query, key, and value, which are all vectors. For a query and its corresponding keys, the attention layer uses a compatibility function to compute attention weights, which represent the relevance of the query with different keys. The attention layer then uses the attention weights to calculate a weighted sum of the values, which is the layer output. In this study, the query refers to item embeddings of the next item for prediction, and both key and value refer to each learner's previous item responses $\mathbf{e}_{ij}^t$. As

such, for each item for prediction, SDCF computes its attentions weights connecting to each of its previous solved items, indicating the relevance of the current item with previous items.

The particular attention used in this study is the scaled dot-product attention (Vaswani et al., 2017). Given the query, key, and value matrices of dimension k (denoted as $\mathbf{S}$, $\mathbf{S}$, and $\mathbf{V}$ respectively), the scaled dot-product attention output is calculated as:

$$\text{Attention}(\mathbf{V},\ \mathbf{S},\ \mathbf{S}) = \text{softmax}(\mathbf{V}\mathbf{S}^T/\sqrt{k})\mathbf{S}, \tag{28}$$

where $\text{softmax}(\mathbf{V}\mathbf{S}^T/\sqrt{k})$ indicates the attention weights. It should be noted that when predicting the item response at timestep $T+1$, only the learner-item interaction over the previous T timesteps should be considered. As such, for any query at timestep t, keys at timesteps later than t should be omitted for computing the weights.

According to Vaswani et al. (2017), to impose non-linearity on the weighted attention output, a neural network architecture consisting of one feedforward layer and one layer with ReLU activation is applied to each timestep separately on top of the attention layer. Formally, the output of the neural network architecture is calculated as:

$$\text{F} = \max(0, xW_1 + b_1)\, W_2 + b_2, \tag{29}$$

where W_1, W_2, b_1, and b_2 are learnable parameters of the neural network layers.

In addition, a residual connection (He et al., 2016) followed by layer normalization (Ba et al., 2016) is applied to both the attention layer and the two-layer neural network architecture. Specifically, a residual connection adds the input and the output of each layer as the final output, so the importance of lower-layer features could be better captured for prediction.

Prediction

The prediction module of SDCF is shown by the right-hand part of the graphical representation in Figure 13. Specifically, the output of the deep LSTM network architecture

D is concatenated with the output of the attention mechanism F, and they are fed into a neural network layer of output dimension one with *Sigmoid* activation for prediction:

$$\hat{R}^i_{T+1} = \text{Sigmoid}\left(\mathbf{W}^T \begin{bmatrix} \text{D} \\ \text{F} \end{bmatrix}\right). \quad (30)$$

SDCF Learning

The following model parameters are to be updated in training: the embedding weights for items and learners, the weights of the deep LSTM network architecture for sequential learning, the weights of the attention mechanism, and the weights of the final neural network layer for prediction. The objective function for learning the model weights can be obtained by taking the negative logarithm of the likelihood of the observed sequence of learner item responding process, which is the binary cross-entropy loss:

$$J = -\sum_{t=1}^{T} R^i_t \log \hat{R}^i_t + (1 - R^i_t)\log\,(1 - \hat{R}^i_t), \quad (31)$$

where $\hat{R}^i_t$ denotes the predicted probabilities of correct item responses at the tth timestep. The model weights are recursively updated to minimize the objective function. The optimization method of Adaptive Moment Estimation (*Adam*; Kingma & Ba, 2014) is selected to determine how to update the model weights in training. *Adam* is an exponentially powerful and popular optimizer in deep learning, given its feature of individualizing learning rates for different model parameters.

Experimental Setup

In the following sections, extensive experiments are conducted to evaluate the effectiveness of SDCF with a simulated dataset and a real-world dataset. The experiments address the following specific research questions.

- Does SDCF show higher predictive capacity than DKT?
- Does using learners' fewer history item responses for training lead to lower prediction performance for SDCF?
- How interpretable are the item-skill associations estimated by SDCF?

Dataset Description

The synthetic datasets were obtained by simulating 4000 virtual learners' item responses on 30 items measuring one of three latent skills. All students answer the same sequence of 30 items, but the order of items varies by student. The probability of a learner getting an item correct given his or her latent skill level was modelled using the Rasch model (see Equation 3). Specifically, each of the three skills was measured by 10 items, and items measuring the same skill were of different difficulties, randomly sampled from a normal distribution with a standard deviation of 1 and a mean of 0. Each learner has a latent ability level on each of the three skills, which were randomly sampled from a normal distribution with a standard deviation of 1 and a mean of 0. Moreover, learners' skill levels were set to slightly increase by 0.1 each time they are tested on items of the same skill. The simulation performed ten times, resulting in ten different synthetic datasets. The models were trained and tested without seeing the mapping of items to skills.

The real-world dataset — "Lab study 2012 (cleanedLogs)" under the project "Fractions Lab Experiment 2012" led by Vincent Aleven — is a web-based tutoring dataset obtained from the PSLC DataShop[2] (Koedinger et al., 2010). The dataset is of 74 learners, 14,959 problem-solving steps, and 37,889 transactions. Regarding the number of latent skills, there is a total of six latent skill models specifying different numbers of latent skills. The model labelled "KC (DefaultFewer_corrected)" was selected for DKT training. In the web-based tutoring system, learners attempted to solve mathematical problems on fractions. Particularly, learners might be assigned with different sets of problems with different problem content, implying that the item sequences for each learner are not identical. To solve a problem, learners needed to take a set of problem-solving steps, each of which was considered as an independent item in this study. In the log data, each step (i.e., item) is

[2] https://pslcdatashop.web.cmu.edu/

associated with a set of transactions, indicating learners' problem-solving actions interacting with the system. Moreover, in the log data, each action is associated with a time variable, indicating how long a learner took for each action. It is required to preprocess the dataset prior to training the model. First, all transactions produced by the tutor system rather than the learner, and all transactions without time information are removed. Second, steps with an outcome of "hint" are treated as intermediate actions for solving corresponding items. Third, because the system used the same labels for actions of the same categories, actions are combined with associated selections by learners to improve their differentiability. Finally, because the dataset has a limited number of samples (i.e., 74 learners) and most samples have an item sequence of more than 200 items, the 74 item sequences are split into multiple subsequences with a fixed length of 20. As a result, there are 866 item sequences used for training and testing the model. After data pre-processing, the final dataset includes 32 unique items and 15 unique skills, and approximately 73% of item responses are correct.

SDCF Training Setting

Hyperparameter tuning is conducted as follows. For item and learner embedding weights, a hyperparameter search was conducted on the following four candidate regularization weights: 0, 0.001, 0.01, and 0.1. Larger regularization weights lead to sparser embedding weights (i.e., item- and learner-skill associations). Regularization weights of 0 and 0.001 were selected for the real-world and synthetic datasets respectively. In addition, prior to each neural network layer, a dropout layer with a dropout rate of 0.2 (selected from candidate rates of 0, 0.2, and 0.5) was used to prevent overfitting for both datasets (Srivastava et al., 2014). The deep LSTM network architecture contained one layer with an output dimension of five; the deep neural network architecture for prediction contained one layer with an output dimension of three. Moreover, latent dimensions of 100 and 300 were selected for embedding item and learner IDs for the real-world and synthetic datasets respectively.

Regarding the learning rate for *Adam*, a hyperparameter search was conducted on the following four candidate learning rates: 0.0001, 0.001, 0.01, and 0.1. Specifically, 0.0001 and 0.001 were selected for the real-world and synthetic datasets, respectively. Regarding batch sizes, a hyperparameter search was conducted on the following values: 8, 32, 64, 128, and 256. The model was trained with batch sizes of 256 and 8 for the real-world and synthetic datasets respectively. SDCF (as well as LogCF and LogSDCF) was programmed and implemented with the deep learning library *keras* (Chollet, 2015) in Python (Python Software Foundation, 2019).

Baseline

To evaluate its effectiveness and predictive capacity, SDCF was compared with DKT. The latter learns individuals' sequential item responses as inputs for predicting probabilities of correct item responses based on RNNs (Piech et al., 2015), which was introduced in Chapter 2. DKT was used as a baseline in the majority of educational mining papers. As a sequential modeling technique, it was found to outperform conventional models such as BKT and AFM (e.g., Xiong et al., 2016). In this study, for the real-world dataset, using item IDs for DKT training resulted in slightly worse prediction performance than using skill IDs, so DKT was learned using skill IDs with a learning rate of 0.001 and a hidden node size of 100 for the LSTM layer. For the synthetic dataset, DKT was learned using item IDs with the same learning rate and hidden node size as for the real-world dataset.

In addition to DKT, this study also compares SDCF with its two variants, SDCF-Attention and SDCF-LSTM, which are two sub-architectures of SDCF (see Figure 13). Specifically, in SDCF, the output of the attention mechanism is concatenated with the output of the LSTM architecture for the final prediction. SDCF-Attention and SDCF-LSTM, however, use the outputs of the two sub-architectures separately for the final prediction.

Evaluation

To address the research questions, the prediction performance of SDCF was evaluated under the condition of different training/test partition rates. Specifically, for each learner item response sequence, the first 30%, 50%, and 70% of item responses were used for training and the remaining ones were used for testing.

The predictive capacities of each model were evaluated from both the regression and classification perspectives. Accuracy (ACC) and the Area Under the Receiver Operating Characteristic (ROC) Curve (*AUC*; Ling et al., 2003) were used as the classification evaluation metrics. The ACC is simply calculated as the percentage of samples that are correctly predicted. As the model yields predicted probabilities of correct responses ranging from 0 to 1, the ACC is typically calculated based on a cut-off value of 0.5 for classifying an item response as correct or incorrect. In contrast to the ACC, the AUC is calculated without specifying any cut-off values. As its name suggests, it is calculated as the area under the plot of sensitivity rates against the false-positive rates. The sensitivity rates and the false-positive rates are calculated at a wide range of possible cut-off values, which makes AUC insensitive to class imbalance (i.e., many correct responses but few incorrect responses or versa vice). Moreover, the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) were used as the regression evaluation metrics (Willmott et al., 2005). Specifically, with respect to this study, if the model predicts that the probability of learner i correctly answering item j is P_{ij} with a ground truth R_{ij} and a total of N predictions, MAE is calculated as

$$\text{MAE} = \frac{\sum_{i,j}\left|P_{ij} - R_{ij}\right|}{N}, \tag{32}$$

and RMSE is calculated as

$$\text{RMSE} = \sqrt{\frac{\sum_{i,j}(P_{ij} - R_{ij})^2}{N}}. \tag{33}$$

Experimental Results

Main Prediction Results

Tables 3 and 4 show the testing performance of each model given different training/test partition ratios for the "Lab study 2012" dataset and the synthetic dataset respectively. Generally, for both datasets, SDCF outperforms DKT on all evaluation metrics, except for the higher RMSE of SDCF when the training ratio is 0.5 or 0.3 for the synthetic dataset. Moreover, using more history items for training improves the prediction accuracy of SDCF for the real-world dataset, shown by higher ACC and AUC rates and lower MAE and RMSE rates for higher training ratios. However, for the synthetic dataset, the differences in prediction performance between different training/test partition ratios are less substantial, and only using the first 30% items for training could result in satisfying prediction performance.

Table 3

Model Prediction Performance of SDCF for the Real-World Dataset

Model	ACC	AUC	MAE	RMSE
Training ratio: 0.7				
DKT	0.7037	0.7157	0.3786	0.4339
SDCF	**0.7143**	0.7347	0.3583	**0.4298**
SDCF-Attention	0.7141	**0.7352**	**0.3582**	0.4300
SDCF-LSTM	0.7106	0.7230	0.3907	0.4337
Training ratio: 0.5				
DKT	0.6890	0.6974	0.3739	0.4422
SDCF	**0.7076**	**0.7266**	**0.3587**	**0.4342**
SDCF-Attention	0.7070	0.7263	0.3589	0.4346
SDCF-LSTM	**0.7076**	0.7149	0.3799	0.4356
Training ratio: 0.3				
DKT	0.6672	0.6439	0.3764	0.4748
SDCF	**0.7065**	0.7182	0.3595	**0.4382**
SDCF-Attention	0.7044	**0.7183**	0.3602	0.4387
SDCF-LSTM	**0.7065**	0.6945	**0.3548**	0.4473

Note. ACC = Accuracy; AUC = Area under the ROC Curve; MAE = Mean Absolute Error; RMSE = Root Mean Square Error.

Table 4

Model Prediction Performance of SDCF for the Synthetic Dataset

Model	ACC	AUC	MAE	RMSE
Training ratio: 0.7				
DKT	0.8279	0.9226	0.2290	0.3413
SDCF	0.8592	0.9473	0.1483	0.3389
SDCF-Attention	0.8574	0.9483	0.1507	0.3400
SDCF-LSTM	0.7616	0.8680	0.3204	0.3986
Training ratio: 0.5				
DKT	0.8589	0.9426	0.1912	0.3097
SDCF	0.8741	0.9531	0.1331	0.3210
SDCF-Attention	0.8782	0.9536	0.1287	0.3160
SDCF-LSTM	0.7664	0.8643	0.3079	0.3941
Training ratio: 0.3				
DKT	0.8491	0.9299	0.1930	0.3247
SDCF	0.8544	0.9368	0.1526	0.3482
SDCF-Attention	0.8574	0.9369	0.1500	0.3436
SDCF-LSTM	0.7478	0.8434	0.3056	0.4051

Note. ACC = Accuracy; AUC = Area under the ROC Curve; MAE = Mean Absolute Error; RMSE = Root Mean Square Error.

Regarding the comparison between SDCF and its two variants, it is evident that SDCF slightly outperforms its LSTM and attention sub-architectures in terms of prediction performance for the real-world dataset. However, for the synthetic dataset, SDCF-Attention slightly outperforms SDCF in terms of prediction performance when the training ratio is small. This suggests that SDCF is a flexible modeling framework weighing its two sub-architectures differentially in training for datasets of different characteristics.

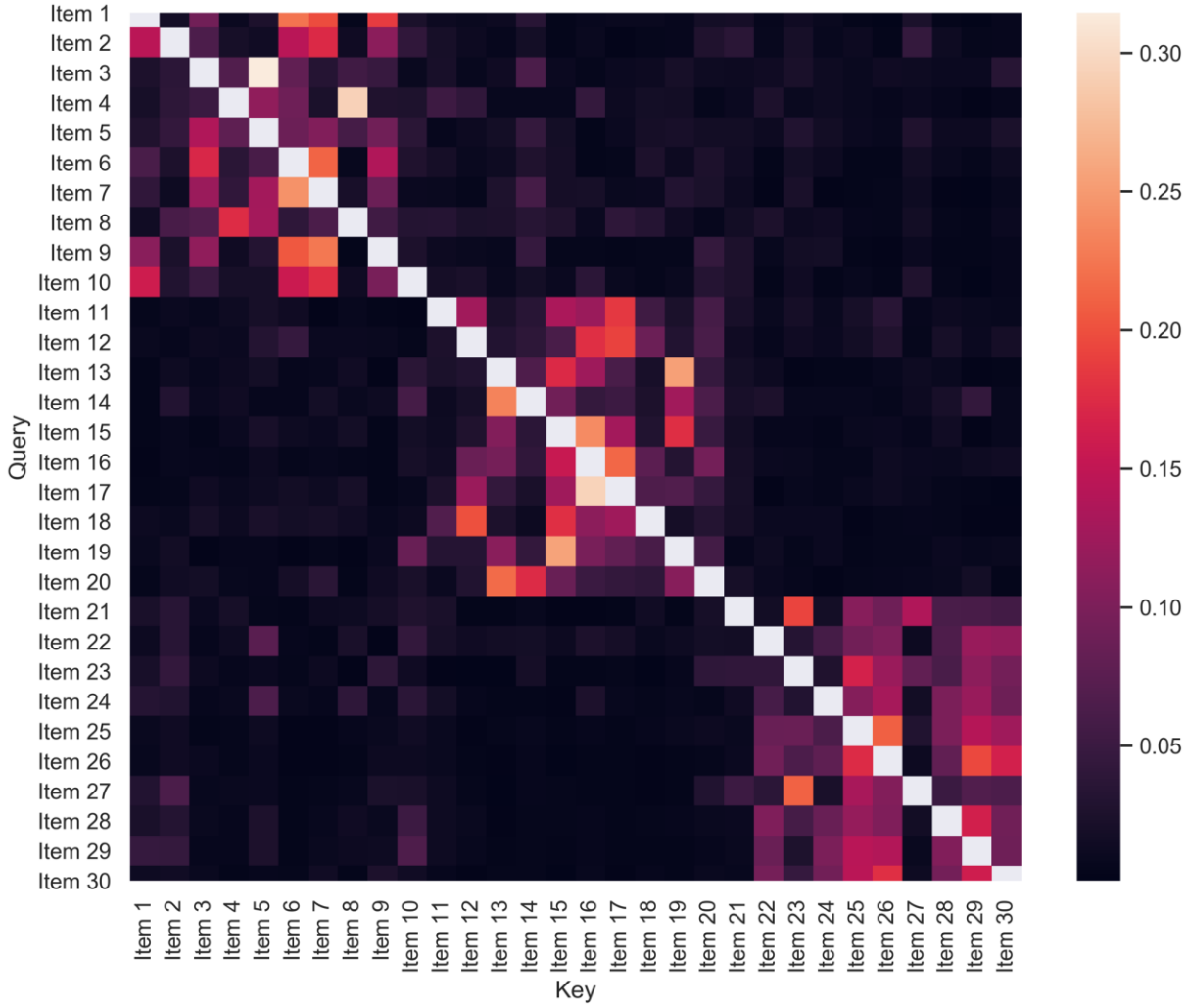
Item-Skill Associations Discovered by SDCF

This study follows the approach proposed by Pandey and Karypis (2019) to demonstrate the interpretability of SDCF. Specifically, for each of a learner's item responses (i.e., the query), SDCF estimates its attention weights connecting to each of its previous item responses (i.e., the keys), indicating the relevance of the current item with the previous items. As such, the relevance weight for each item pair (i.e., [query item, key item]) can be calculated as the sum of its attention weights across all learners. For each query item, its relevance weights are then normalized so that they all sum to one. The relevance weights of

each item indicate the strengths of its connections to other items. Given the relevance weights, items measuring the same skills can be discovered by identifying which items have the strongest connections to each item.

Figure 14

Heatmap of Item Relevance Weights Estimated by SDCF for the Synthetic Data

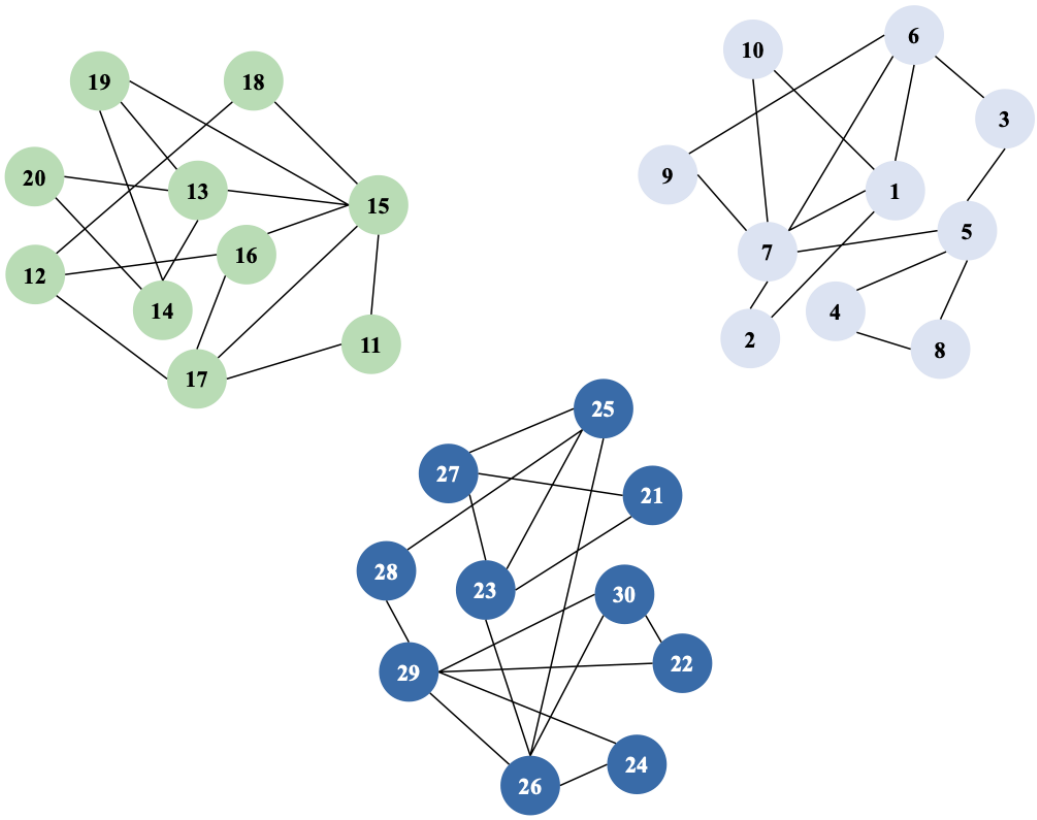


For example, for the synthetic dataset, learners provide responses to 30 items in different orders. In the synthetic assessment, items 1 to 10, items 11 to 20, and items 21 to 30 measure the hidden skills 1, 2, and 3 respectively. For each of the 30 items, its relevance weights for the other 29 items can be calculated as the normalized sum of attention weights across 4000 learners. Figure 14 presents the heatmap of relevance weights for all item pairs. In general, it can be seen that the relevance weights between items measuring the same skill

are much stronger than those between items measuring different skills. Using the relevance weights, the top two items of the strongest weights are then connected to each query item. According to Figure 15, SDCF achieves a perfect clustering of items measuring the same skills.

Figure 15

Graph Depicting the Clustering of Items Measuring the Same Skills by SDCF



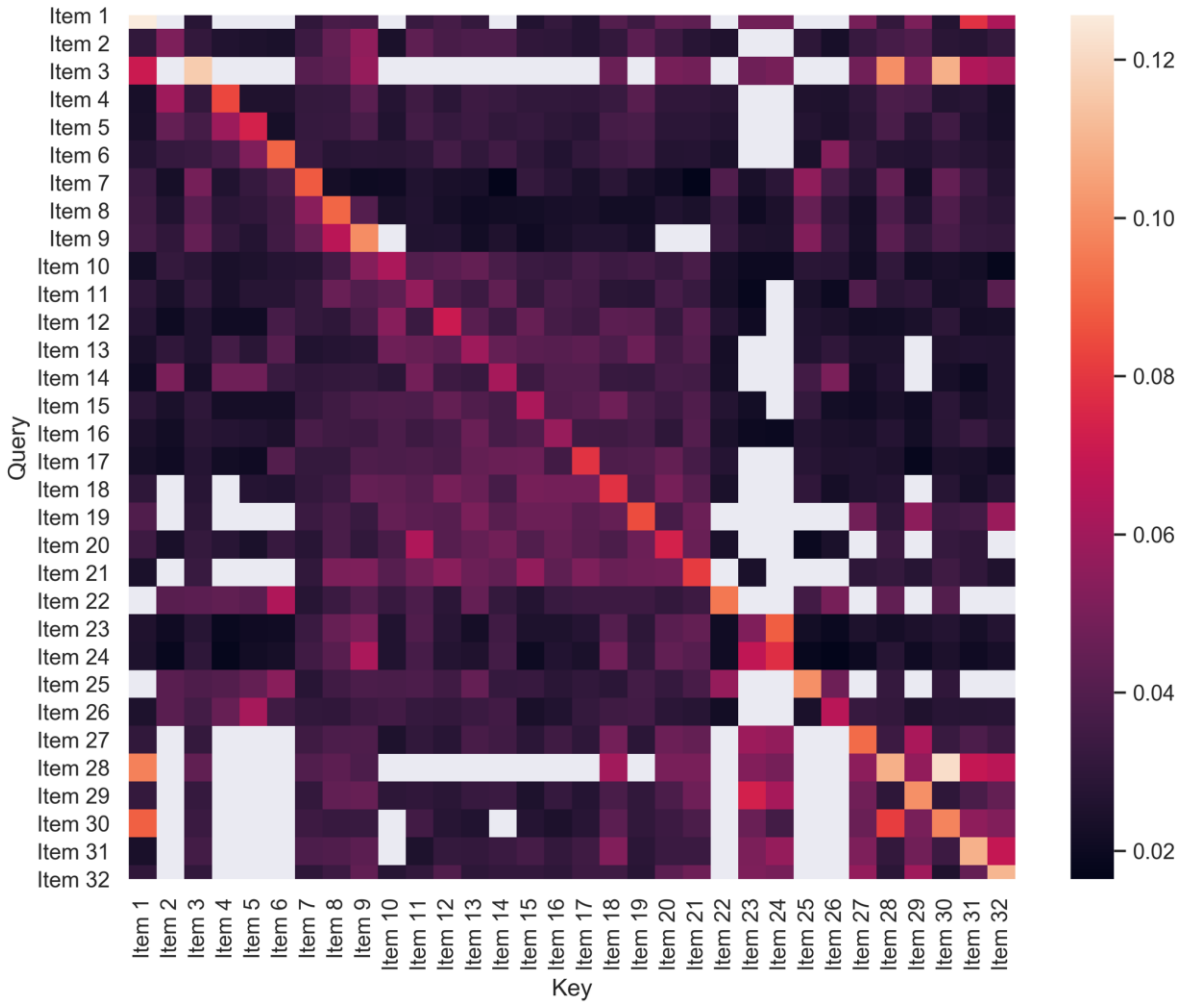
Note. Items 1 to 10, items 11 to 20, and items 21 to 30 measure the hidden skills 1, 2, and 3 respectively.

Figure 16 presents the heatmap of item relevance weights for the real-world dataset. In general, compared with the heatmap for the synthetic dataset, the heatmap for the real-world dataset suggests a less clear-cut clustering of items. However, it is still evident that items 10 to 21 constitute a major cluster given their stronger relevance weights between each other. Table 5 presents the item and skill names for the real-world dataset. It can be seen that items 10 to 21 measure the same skill labelled "equivDragFract", and their associations were

correctly discovered by SDCF. For the original skills measured by only one or two items, the model could not accurately identify their differences. However, this finding was not surprising since skills measured by one or two items could not be adequately exercised by learners and the learned relevance weights were inevitably affected by more randomness. Moreover, it should be noted that the "true" item-skill associations for this dataset are not known, so researchers proposed multiple skill models for this dataset. Therefore, skill labels in Table 5 are not necessarily the ground truth. In general, the capacity of SDCF to discover item-skill associations is to some extent justified given that the major skill measured by most items was successfully identified.

Figure 16

Heatmap of Item Relevance Weights Estimated by SDCF for the Real Data



Note. The item name and skill name for each item ID are presented in Table 5.

Table 5

Item and Skill Names for the Real-World Data

ID	Item Name	Skill Name
1	combo1 UpdateComboBox	equivFractEquivalent
2	combo1_3 UpdateComboBox	compFract
3	combo2 UpdateComboBox	relationEquivMultiplySameNumber
4	combo2_1 UpdateComboBox	compSectSize
5	combo2_2 UpdateComboBox	compNumSect
6	combo2_3 UpdateComboBox	compFract
7	combo3 UpdateComboBox	relationEquivConserveAmount
8	combo4 UpdateComboBox	relationEquivSameAmount
9	combo5 UpdateComboBox	relationEquivDiffNumbers
10	dragTarget1 WasJustHitByA Circle	equivDragFract
11	dragTarget1 WasJustHitByA NL	equivDragFract
12	dragTarget1 WasJustHitByA Rect	equivDragFract
13	dragTarget2 WasJustHitByA Circle	equivDragFract
14	dragTarget2 WasJustHitByA NL	equivDragFract
15	dragTarget2 WasJustHitByA Rect	equivDragFract
16	dragTarget3 WasJustHitByA Circle	equivDragFract
17	dragTarget3 WasJustHitByA NL	equivDragFract
18	dragTarget3 WasJustHitByA Rect	equivDragFract
19	dragTarget4 WasJustHitByA Circle	equivDragFract
20	dragTarget4 WasJustHitByA NL	equivDragFract
21	dragTarget4 WasJustHitByA Rect	equivDragFract
22	fract1_denom1 UpdateTextArea	relationCompTotalSectNumber
23	fract1_denomMultiply1 UpdateTextArea	equivMultiplyDenom
24	fract1_numMultiply1 UpdateTextArea	equivMultiplyNum
25	fract2_denom1 UpdateTextArea	relationCompTotalSectNumber
26	fract2_num1 UpdateTextArea	numSectZeroDot
27	fract3_denom UpdateTextArea	equivNameDenomFract
28	fract3_denomMultiply1 UpdateTextArea	equivMultiplyDenom
29	fract3_num UpdateTextArea	equivNameNumFract
30	fract3_numMultiply1 UpdateTextArea	equivMultiplyNum
31	fract4_denom UpdateTextArea	equivNameDenomFract
32	fract4_num UpdateTextArea	equivNameNumFract

Chapter 4 LogCF: Deep Collaborative Filtering with Process Data[3]

The goal of this chapter is to approach the second research problem by proposing a CF-based general framework of learning outcome modeling with process data. The proposed approach for learning outcome modeling with process data, LogCF, attempts to integrate a deep learning-based CF architecture for learning learner- and item-skill associations and a deep learning architecture for learning process data, for the purpose of enhanced prediction accuracy and interpretability. The following sections start with the problem formulation, followed by the introduction of a general LogCF framework, and the technical details of the deep learning architectures for learning learner- and item-skill associations and process data.

Problem Formulation

Suppose the assessment data involves m learners and n items, measuring k latent skills, which constitute an item response matrix $\mathbf{R} = \{R_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$. In the matrix, $R_{ij} = \langle \mathbf{m}_i, \mathbf{n}_j, r_{ij} \rangle$ indicates that the learner $\mathbf{m}_i$ gives a response r_{ij} on the item $\mathbf{n}_j$. Moreover, for each learner-item interaction R_{ij}, they have an associated problem-solving process $L_{ij} = \langle \mathbf{m}_i, \mathbf{n}_j, l_{ij} \rangle$, indicating that the learner $\mathbf{m}_i$ answers the item $\mathbf{n}_j$ with a problem-solving process l_{ij}. The problem-solving processes L_{ij} constitute a process data matrix $\mathbf{L} = \{L_{ij} | 1 \leq i \leq m, 1 \leq j \leq n\}$. Learners and items are denoted by $\mathbf{m}_i$ and $\mathbf{n}_j$ instead of their subscripts i and j because learners and items can be characterized with various information. In LogCF, $\mathbf{m}_i$ and $\mathbf{n}_j$ are simply learner and item identifications, but they can be extended to other learner and item features.

Having the item response matrix $\mathbf{R}$ and the process data matrix $\mathbf{L}$, our goal is to learn a model $\mathcal{M}$ which is capable of discovering learner-skill associations $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$ and

[3] This chapter was published by the author. See "LogCF: Deep collaborative filtering with process data for enhanced learning outcome modeling", F. Chen and Y. Cui, 2020, Journal of Educational Data Mining, 12, pp. 66–99. https://doi.org/10.5281/zenodo.4399685

item-skill associations $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$ for the prediction of item responses. The interpretations of **U** and **V** are the same as those introduced in the last chapter. Moreover, the item response matrix **R** and the process data matrix **L** are not necessarily complete. This is because a learner might be assigned with a small portion of items in the item pool and correspondingly, an item might be only witnessed by some of all learners. In the case of **R** and **L** with missing entries, the model can simply make predictions of the missing responses R_{ij} based on the learned **U** and **V**.

Specifically, given a binary-valued item response $R_{ij} \in \{0,1\}$ which means that the learner $\mathbf{m}_i$ provided a correct response indicated by 1 or an incorrect response indicated by 0 on the item $\mathbf{n}_j$, the model predicts R_{ij} by:

$$Z_{ij} = \mathbf{h}^T \begin{bmatrix} \phi^{\mathrm{CF}} \\ \phi^{\mathrm{Log}} \end{bmatrix}, R_{ij} \sim Ber(\sigma(Z_{ij})). \tag{34}$$

In the above equation, $Ber(z)$ denotes a Bernoulli distribution with a success probability z followed by the learners' correct responses. Also, $\sigma(z)$ denotes a logistic function converting a real value z to a success probability ranging from 0 to 1, which is given by:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{35}$$

Moreover, Z_{ij} indicates the output produced by a neural network layer that learns the concatenation of the output produced by the deep CF architecture, ϕ^{CF}, and the output produced by the deep learning architecture for learning actions and time durations, ϕ^{Log}. Moreover, **h** denotes the neural network weights for outputting Z_{ij}. More details regarding how to learn ϕ^{CF} and ϕ^{Log} are presented in the general framework of LogCF (Figure 13) and will be discussed in the next section. Briefly, ϕ^{CF} is outputted by a deep neural network architecture which learns the concatenation of **U** and **V** as the input. As mentioned, **U** and **V** indicate the learned individual and item representations, respectively, based on learner and

item identifications produced by CF. Also, ϕ^{Log} is outputted by a deep neural network architecture which learns the concatenation of action and time representations as the input. The action and time representations are produced by deep neural network architectures of multiple LSTM network layers learning the raw action and time sequences as the input.

Having the above definition, we can formularize the problem of learning LogCF, especially estimating $\mathbf{U}$ and $\mathbf{V}$, as a maximum likelihood problem which maximizes the likelihood of the observed item response matrix $\mathbf{R}$,

$$p\left(R_{ij}\middle|\mathbf{u}_i, \mathbf{v}_j\right) = \sigma(Z_{ij})^{R_{ij}}(1-\sigma(Z_{ij}))^{1-R_{ij}}, \tag{36}$$

which is given by:

$$\underset{\mathbf{U},\mathbf{V}}{\text{maximize}} \sum_{i,j} \log p\left(R_{ij}\middle|\mathbf{u}_i, \mathbf{v}_j\right). \tag{37}$$

General Framework

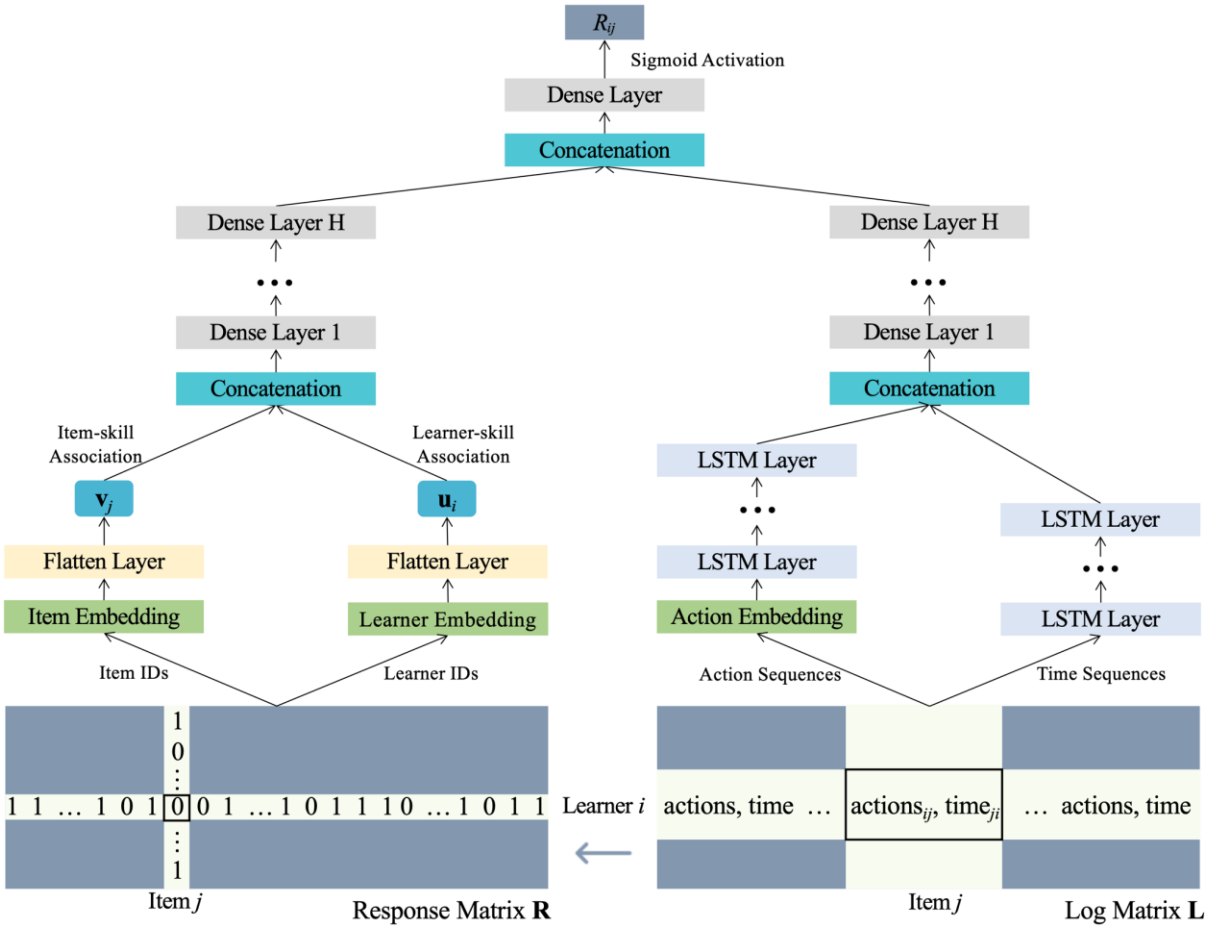
Figure 17 presents the general framework of LogCF, which is of two sub-architectures, a deep CF architecture for learning item- and learner-skill associations (outputting ϕ^{CF}) and a deep learning architecture for learning the process data (outputting ϕ^{Log}).

Deep Collaborative Filtering

How learners interact with items in terms of their affinities with the latent skills is learned by a deep neural network architecture adapted from the neural CF framework (He et al., 2017). In Figure 17, despite the item response matrix shown as the initial inputs, learner and item identifications are used as the raw inputs for the deep CF architecture, which are embedded as learner- and item-skill associations — $\mathbf{u}_i$ and $\mathbf{v}_j$ — respectively. The technical details of embedding layers are the same as those introduced in the last chapter. Above the embedding layers of items and learners, the model uses a flatten layer to reshape the outputs of the embedding layers so that they can be fed into the following deep neural network architecture as inputs.

Figure 17

General Framework of LogCF



The deep neural network architecture on top of the embedding layers is used to further capture the complexity of how learner representations interact with item representations in affecting item responses. This is where LogCF significantly differs from conventional matrix factorization approaches, which simply model the learner-item interaction as the product of $\mathbf{u}_i$ and the item-skill association $\mathbf{v}_j$. The deep neural network architecture of LogCF is capable of learning non-linear interactions between learner and item representations. Concretely, the model first concatenates learner and item representations. Then the concatenated factor is fed into multiple neural network layers for outputting ϕ^{CF}, which is the final representation of learner-item interactions. Formally, the above deep neural network architecture with CF can be formulated as:

$$\phi^{\mathrm{CF}} = f_H(\mathbf{W}_H^T f_{H-1}(\dots f_2(\mathbf{W}_2^T f_1(\mathbf{W}_1^T \begin{bmatrix}\mathbf{U}^T\mathbf{m}_i \\ \mathbf{V}^T\mathbf{n}_j\end{bmatrix})) \dots)), \quad (38)$$

where $\mathbf{U}$ and $\mathbf{V}$ denote learner- and item-skill associations respectively, $\mathbf{W}_1$ to $\mathbf{W}_H$ indicate the neural network weights for the H neural network layers, and f_1 to f_H represent the activation function applied in each neural network layer. Given the above formulation, the number of neural network layers and the number of nodes within each layer are not fixed. Instead, they are two hyper-parameters to be tuned in training.

Deep Learning of the Problem-Solving Process

The right part of the general framework demonstrates a deep learning architecture of multiple LSTM network layers for learning the process data, which shares similarities with the deep neural network architecture introduced above. The reason why LSTM network layers rather than neural network layers are used in process data learning is that learners' actions and time durations are logged as sequential data by the system.

The inputs for the process data learning architecture are learners' raw action and time sequences. Given the action sequence $a_{ij} = \{e_1, e_2, \dots e_Q\}$ indicating that learner $\mathbf{m}_i$ has Q action steps on item $\mathbf{n}_j$, the model first converts each action e_q in a_{ij} to a dense vector of d_0 dimensions through an embedding layer on actions, which is then fed into an LSTM network layer for learning the time-series dependencies between actions (see Equation 26 and description on LSTM in Chapter 3 for more details). As shown in the general framework, the model allows for multiple LSTM layers to better capture the temporal dependencies between actions and time durations in sequences. The multiple LSTM layers finally output learned representations of actions and time durations, which are denoted as A_{ij} and T_{ij} respectively.

Subsequently, the model concatenates A_{ij} and T_{ij} and feeds them into a deep neural network architecture for learning the interactions between actions and time durations. This deep neural network architecture finally outputs the representation of the whole problem-solving process ϕ^{Log}, which is given by:

$$\phi^{\text{Log}} = f_H(\mathbf{W}_H^T f_{H-1}(\dots f_2(\mathbf{W}_2^T f_1(\mathbf{W}_1^T \begin{bmatrix} A_{ij} \\ T_{ij} \end{bmatrix})) \dots)), \quad (39)$$

where the mathematical notations are interpreted as in Equation 35.

Prediction

The prediction module of LogCF is shown by the topmost layer in the general framework. Specifically, for predicting item responses, the model concatenates the outputs produced by the deep CF architecture and the deep learning architecture of process data, ϕ^{CF} and ϕ^{Log}, which are then fed into a neural network layer with one-dimensional output. For producing the probabilities of correctly answering items, the one-dimensional output is converted to a value ranging from 0 to 1 with the *Sigmoid* activation.

Variants of LogCF

The above sections focus on the general framework of LogCF. However, to better evaluate the capacity of LogCF to discover item-skill associations and predict item responses, the dissertation proposes three variants of LogCF sharing the same topology shown in Figure 17. The LogCF variants mainly differ in how they initialize the weights of the deep CF architecture. The three LogCF variants can be categorized as variants with or without expert information. Variants with expert information are as follows.

expert-Q. This variant makes item-skill associations fixed as the ones pre-specified by experts, and all the other model weights are adjustable and learnable. In other words, given this variant, the model relies, entirely, on expert-specified item-skill associations for prediction.

expert-Q-init. This variant initializes item-skill associations with the expert-specified ones. However, item-skill associations in this variant can still be adjusted and learned in training.

Variant without expert information is random-int.

random-init. In this variant, all the model weights are initialized with a uniform distribution ranging in ($\sqrt{-6/(N_i + N_o)}$, $\sqrt{6/(N_i + N_o)}$) where N_i and N_o represent the input size and the output size of different embedding layers, neural network layers, and LSTM network layers (Glorot & Bengio, 2010).

The same pre-training of the model weights in the deep learning architecture for learning process data applies to all three variants of LogCF. That said, the deep learning architecture for learning process data is first trained and learned as a separate and standalone model for learning outcome modeling. Thereafter, for each of the three variants, the model initializes the weights of the deep learning architecture for learning process data with the ones learned by pre-training.

LogCF Learning

The following model parameters are to be updated in training: the embedding weights for items, learners and actions, the weights of the two architectures of multiple LSTM network layers for learning the temporal dependencies between actions and time durations, the weights of the two deep neural network architectures for learning the representations of learner-item interactions and action-time interactions, and the weights of the topmost neural network layer for prediction. The objective function and the optimization method for learning the model weights are the same as those used for SDCF in Chapter 3.

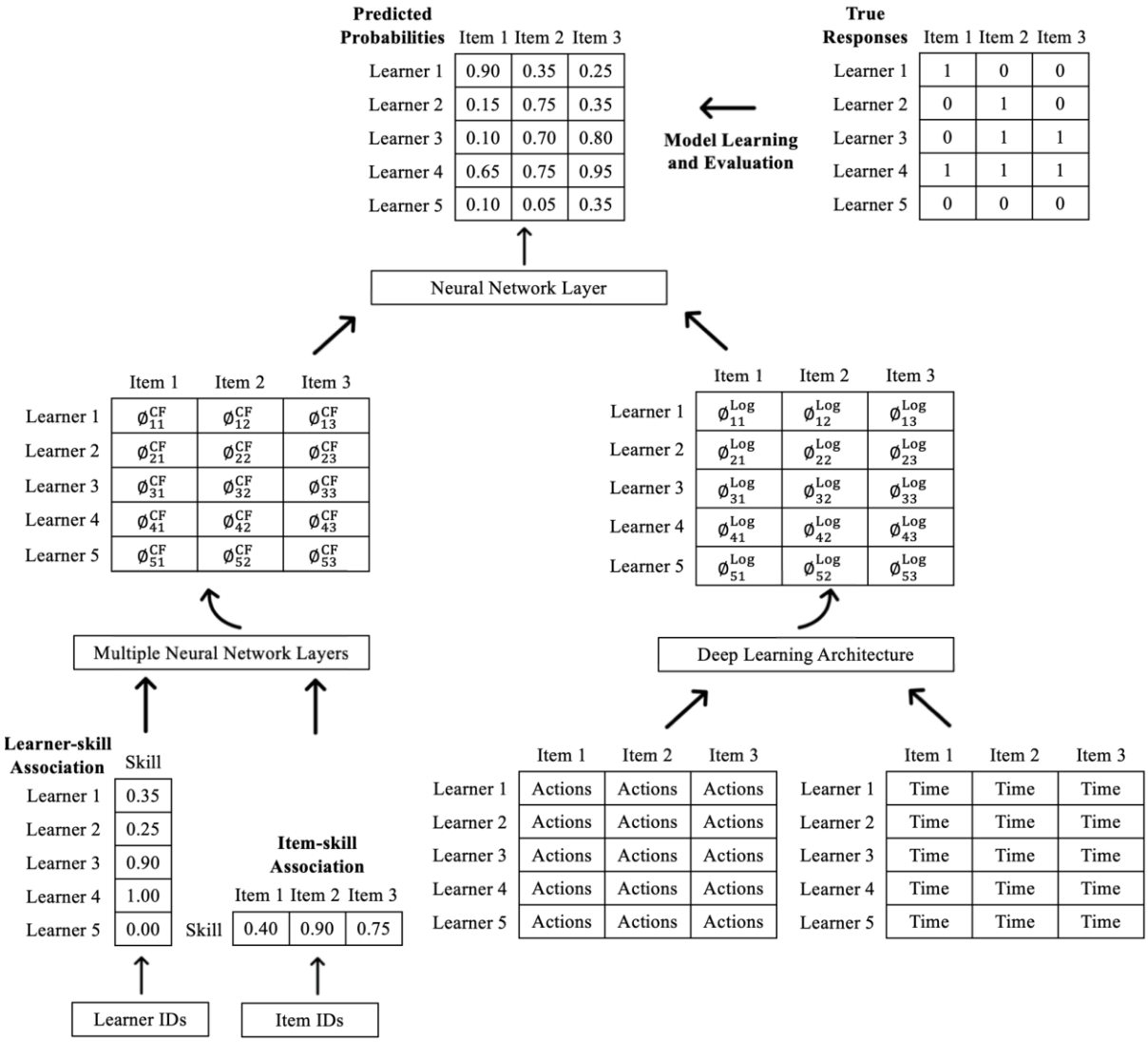
A Hypothetical Example

To further clarify how LogCF works for learning outcome modeling, a hypothetical example is presented in Figure 18. The example is a computer-based assessment evaluating learners' problem-solving competency. Suppose there are only five learners and three assessment items, a 5×3 item response matrix is presented at the top right of Figure 18. Specifically, learners 1 and 2 correctly answered one item, learners 3 correctly answered two items, learner 4 got all items correct, and learner 5 got all items wrong. The goal of LogCF is

to predict the five learners' probabilities of getting the three items correct, which constitute a predicted item response matrix produced by LogCF (the top middle matrix in the Figure).

Figure 18

A Hypothetical Example of LogCF



Aside from predicted probabilities of item responses, the model also estimates item- and learner-skill associations (see the bottom left of Figure 18) for the purposes of learner and domain modeling. Given the example, the learner-skill associations for learners 3 and 4 are stronger than those for the other learners, indicating that learners 3 and 4 have higher problem-solving proficiency. Likewise, item 2 has a stronger association with the latent skill than the other two items, indicating that it measures the latent skill more closely. These item-

and learner-skill associations are directly learned from the identifications of the five learners and the three items through embedding layers, which are then processed by a deep neural network architecture for outputting ϕ^{CF}. In addition, for each learner-item interaction, there is a sequence of problem-solving actions and a sequence of time durations associated with each action logged by the system (see the bottom right of Figure 18). The action and time sequences are then processed by deep learning architectures of multiple neural and LSTM network layers for outputting ϕ^{Log}. Then, ϕ^{CF} and ϕ^{Log} (see the middle of Figure 18) are concatenated and processed by a neural network layer for producing the predicted probabilities of correct item responses.

Experiments

In the following sections, extensive experiments are conducted to evaluate the effectiveness of LogCF with two distinctive datasets of unique characteristics. The experiments address the following five specific research questions.

- Does LogCF show higher predictive capacity than the baselines?
- Does expert information on item-skill associations contribute to higher prediction performance for LogCF than learning item-skill associations from scratch?
- Does the number of latent skills affect the prediction performance of LogCF?
- Does LogCF show good prediction performance at different levels of missing responses or different percentages of learner fist item responses going for training?
- Can we interpret the estimated learner- and item-skill associations by LogCF in the context of computer-based assessment?

To address the research questions, the prediction performance of LogCF is evaluated under the condition of different training/test partition rates. Moreover, the two datasets are analyzed for different research questions. The educational data mining dataset is used to

investigate the effects of the number of latent skills on prediction performance and demonstrate how well LogCF is capable of retrieving or refining item-skill associations specified by experts. The educational assessment dataset is mainly used to demonstrate the interpretability of learner- and item-skill associations compared with the model parameters of IRT models.

Dataset Description

The first dataset was introduced in Chapter 3 and it was used to evaluate SDCF, whereas the second dataset was introduced in Chapter 2 and was obtained from PISA 2012 (https://www.oecd.org/pisa). In 2012, PISA assessed the mathematics literacy of approximately 470,000 students from 65 countries and economies (Organisation for Economic Co-operation and Development, 2014). Moreover, it also evaluated students' creative problem-solving competency. An example problem-solving assessment question was introduced in the section “A Computer-Based Assessment Example on Problem Solving” of Chapter 2. The data of problem-solving assessment includes both students’ question scores and the process data on problem solving. The process data is publicly available at https://www.dropbox.com/s/b8kb4jmqnha6jom/CPRO_logdata_released.zip?dl=0. Given that students were assigned with different subsets of assessment items, given the requirement of a large sample size for deep learning, the data of four items is a subset from the publicly available datasets (10,070 students). Although the problem-solving assessment included a few different items, not all the items were suitable for process data learning because the information on correct item responses might be stored in the process data. According to the PISA 2012 context framework, the four items measure a single competency—complex problem solving. Moreover, when solving the problems, students could not use any explicit hints and provide a solution with multiple attempts. According to the scoring rubric, a solution would be given a credit of 2 when it was fully correct, a partial credit of 1 when it

showed some correct actions but was not fully correct, and a credit of 0 when it did not show any correct actions and was fully incorrect. Students' scores are recoded so that a correct response is valued at 1 and a partial or incorrect response is valued at 0. Moreover, students' action events (e.g., drawing lines between controls and humidity/temperature) and corresponding event types (e.g., the type of "Diagram") are concatenated as actions. The time durations are calculated as the difference between two consecutive timestamps. In addition, all the actions showing how learners gave final solutions on solving an item were blinded because they were directly associated with item scores.

The two datasets differ in several aspects. First, compared with the PISA 2012 dataset which is a structured standardized assessment dataset, the "Lab study 2012" dataset is much more unstructured. Second, compared with the PISA 2012 dataset, the "Lab study 2012" dataset has much fewer learners but much more items. Third, the competencies measured by the two assessments are largely different, and the "Lab study 2012" items on fractions are much more straightforward than PISA 2012 items on complex problem solving. Therefore, it is reasonable to posit that the process data might be less useful for improving prediction for the "Lab study 2012" assessment than for the PISA 2012 assessment. Finally, compared with the PISA 2012 dataset measuring only one latent skill, the "Lab study 2012" dataset has many more latent skills.

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.

Baselines

To evaluate the effectiveness and predictive capacities of LogCF, LogCF is compared with the following models.

NeuralCF. The first baseline approach is the deep neural network architecture with CF, called NeuralCF, shown in Figure 17. NeuralCF, unlike LogCF, makes the final prediction solely based on the output of the deep neural network architecture as its output is not concatenated with the process data learning output but directly fed into the prediction module. As a sub-architecture of LogCF, NeuralCF has the same three variants as LogCF does. In addition, given that NeuralCF is obtained by dropping the process data learning architecture of LogCF, it is reasonable to posit that LogCF outperforms NeuralCF because it includes much more learner information for training.

Log. The Log method is the process data learning architecture shown in Figure 17. 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.

DKT. It was used as a baseline in Chapter 3 and was introduced in Chapter 2.

Given the differences between the two datasets, the above baselines apply to the "Lab study 2012" dataset only. For the PISA 2012 dataset, because it has a limited number of items and is unidimensional, its predictive capacities and interpretability are evaluated in comparison with three IRT models, the Rasch model, the 2PL model, and the 3PL model, which were introduced in Chapter 2. Notably, by using IRT models, the item-skill associations estimated by LogCF can be interpreted against both item difficulty and item discrimination estimated by IRT models. Because IRT models are psychometric models which are strongly interpretable and used widely in education, the comparison shed light on the potential of deep learning in the psychometric analysis.

Table 6

Model Performance 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
	NeuralCF	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
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
	NeuralCF	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
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
	NeuralCF	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
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
	NeuralCF	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

Note. DKT = Deep Knowledge Tracing; LogCF refers to the full model proposed in this study; NeuralCF refers to the the deep learning-based CF architecture of LogCF; Log refers to the deep learning architecture of LogCF for learning the process data. ACC = Accuracy; AUC = Area under the ROC Curve; MAE = Mean Absolute Error; RMSE = Root Mean Square Error.

Evaluation

LogCF is evaluated with different training/test partition ratios. Specifically, for each variant of LogCF, 40%, 30%, 20%, and 10% of all item response entries are used as the test dataset and the remaining entries are used as the training dataset. Moreover, 20% of the

training item response entries are used as the validation dataset in training. The data is partitioned at the entry level, implying that each training or test sample is one independent item response associated with its actions and time information. In addition to data partition at the entry level, to further evaluate the effectiveness of LogCF, the "Lab study 2012" dataset is also partitioned sequentially as the way used for evaluating SDCF. The evaluation metrics are the same as the ones used in Chapter 3.

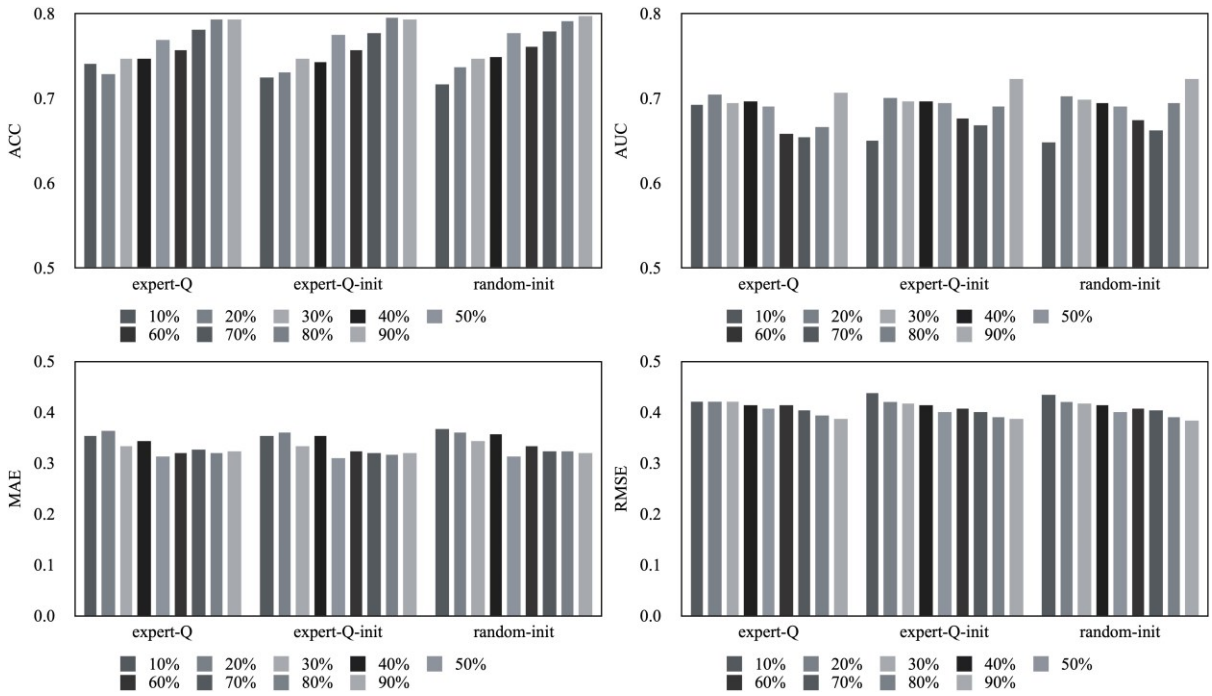
Experimental Results

Main Prediction Results

Table 6 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 LogCF slightly outperforms other models in terms of predictive power.

Figure 19

Model Performance of LogCF for Sequential Training/Test Partition



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.

Regarding the model performance given different levels of learners' first item responses going for training, Figure 19 shows 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 sequentially. This is however an unsurprising finding given that the class weights for training and test datasets might be largely different in this case.

Figure 20

Model Performance of LogCF for the PISA Dataset

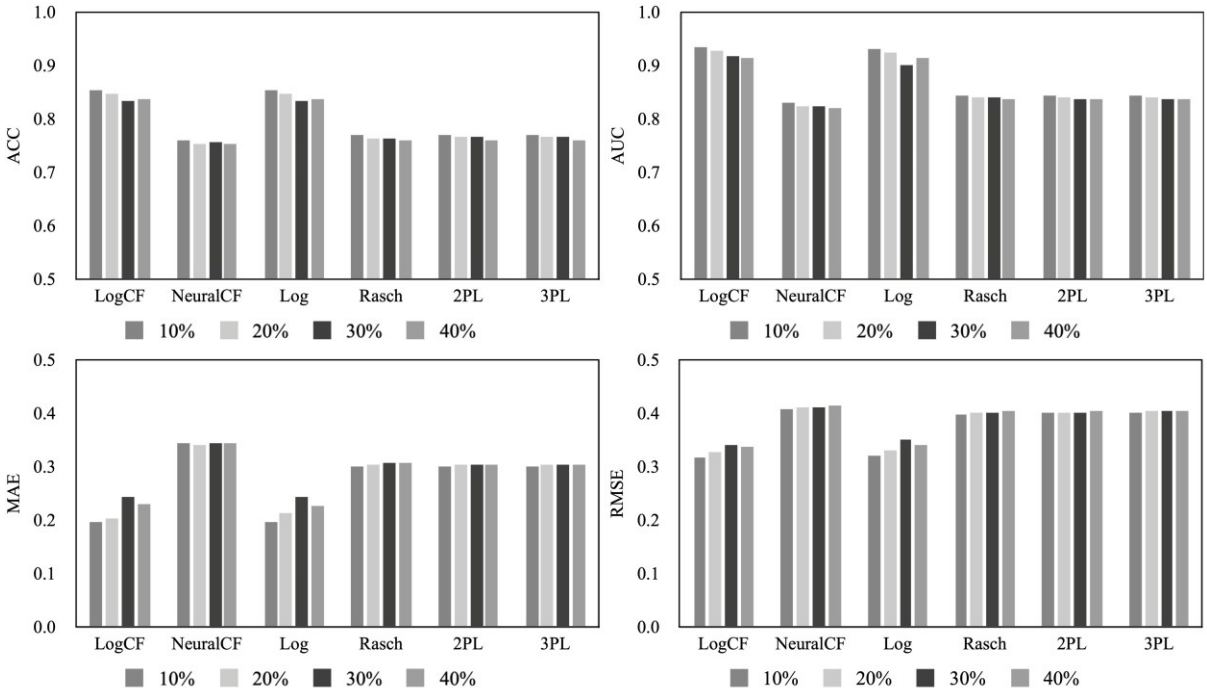


Figure 20 shows the model performance for the PISA 2012 dataset with respect to each evaluation metric at different levels of missing responses. It can be seen that, compared with the IRT models, LogCF performs the best under all the experimental conditions. Concretely, in terms of ACC and AUC, LogCF 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, 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, although 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, LogCF demonstrates a substantially higher prediction performance than the baselines. Moreover, its prediction performance would not be greatly affected by the missing response rates, indicating the robustness of LogCF.

Performance of Learning or Refining Item-Skill Associations

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.

Table 7

Model Performance of LogCF for 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

Note. ACC = Accuracy; AUC = Area under the ROC Curve; MAE = Mean Absolute Error; RMSE = Root Mean Square Error.

Effects of the Number of Latent Skills

To answer the third research question, the LogCF variants without expert information were further evaluated with different latent skill dimensions. Table 7 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.

Interpretability of LogCF

The interpretability of a model is especially beneficial to educational practitioners. In this 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 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). However, LogCF, models a single item parameter, which therefore can be considered similar to 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 answer it correctly.

Table 8

Item-Skill Associations and Item Parameters Estimated by LogCF and Baselines

Item	LogCF	NeuralCF	Rasch	2PL		3PL	
			d_j	a_j	d_j	a_j	d_j
1	-0.07	-0.02	-0.17	1.53	-0.18	2.49	0.87
2	-1.06	-1.2	-2.23	1.84	-2.5	1.67	2.38
3	1.42	1.51	2.23	1.59	2.32	1.72	2.41
4	0.59	0.65	1.4	1.1	1.26	1.11	1.26

Note. d and *a* refers to item intercept and item discrimination respectively.

Item-Skill Association. Given the above theoretical clarification, this study further compared 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 8, 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, both the item discrimination a_j and the item intercept $d_j = -a_j \times b_j$ are presented in Table 8 (given a 30% missing rate). 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, a line chart is presented to show the pattern of item parameters of each method in Figure 21. 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.

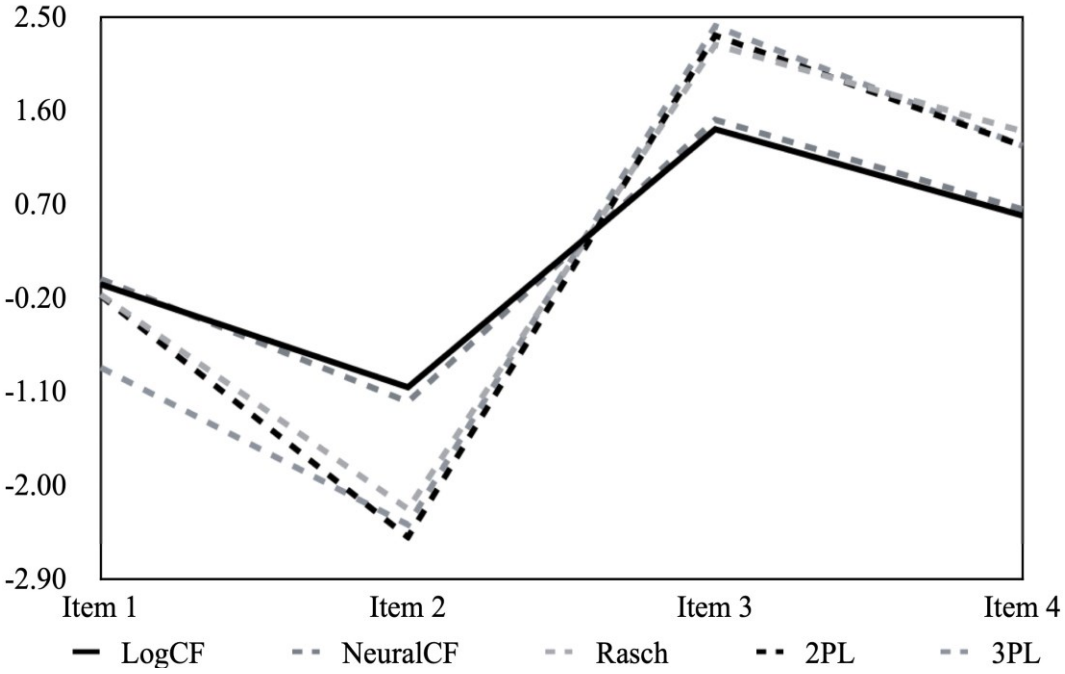
In addition to the comparison of item parameters across methods, this study also referred to the PISA 2012 assessment framework and results (OECD, 2014a) for more evidence validating the results. According to the PISA 2012 assessment framework, the four items involved in the analyses map onto different 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, 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 item difficulties were not presented in the Table, they can be calculated as the negative values of item intercepts divided by item discriminations). However, it should be noted 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 8, 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 this 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, the process data learning of LogCF can be considered as a regularization technique for training. As Figure 21 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.

Figure 21

A plot of Item-Skill Associations and Item Intercepts for LogCF and Baselines



Note. Lower values indicate higher proficiency levels required by items.

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, this study calculated the correlation coefficients of learner parameters between LogCF and the baselines, which are presented in Table 9 (given a missing rate of 30%). 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.

Table 9

Correlation Coefficients of Learner Parameters between LogCF and Baselines

	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

Chapter 5 LogSDCF: Sequential Deep Collaborative Filtering with Process Data

The goal of this chapter is to approach the third research problem by integrating the first two proposed models. The proposed model, LogSDCF, is used for sequentially modeling both product and process data. In general, LogSDCF is a variant of SDCF with an additional architecture for process data learning. The following sections start with the problem formulation, followed by the introduction to a general framework of LogSDCF and its technical details. Given that LogSDCF is a hybrid of LogCF and SDCF, to save space, some duplicate technical details are omitted in this chapter.

Problem Formulation

The problem formulation for LogSDCF is similar to that for SDCF, with the distinction that learner process data is used as auxiliary inputs. Formally, suppose that a hypothetical assessment of n items measures k latent skills, and there are m independent learners who take the assessment. Each learner's item responding process can be denoted as $\mathbf{R}_i = \{(\mathbf{m}_i, \mathbf{n}_1^i, R_1^i, L_1^i), (\mathbf{m}_i, \mathbf{n}_2^i, R_2^i, L_2^i), \dots, (\mathbf{m}_i, \mathbf{n}_T^i, R_T^i, L_T^i)\}$, where $\mathbf{m}_i$ denotes the identification of the learner, $\mathbf{n}_t^i$ denotes the item $\mathbf{n}_t$ responded by learner $\mathbf{m}_i$ at the tth timestep, R_t^i denotes the corresponding item response result (correct/incorrect), and $L_t^i = \{\boldsymbol{a}_t^i, \boldsymbol{t}_t^i\}$ denotes the problem-solving process associated with R_t^i, which includes both action sequence $\boldsymbol{a}_t^i$ and time sequence $\boldsymbol{t}_t^i$. If learner $\mathbf{m}_i$ correctly solves item $\mathbf{n}_t$, $R_t^i = 1$, otherwise $R_t^i = 0$. In LogSDCF, learner and item identifications, $\mathbf{m}_i$ and $\mathbf{n}_t$, are embedded as learner and item latent representations.

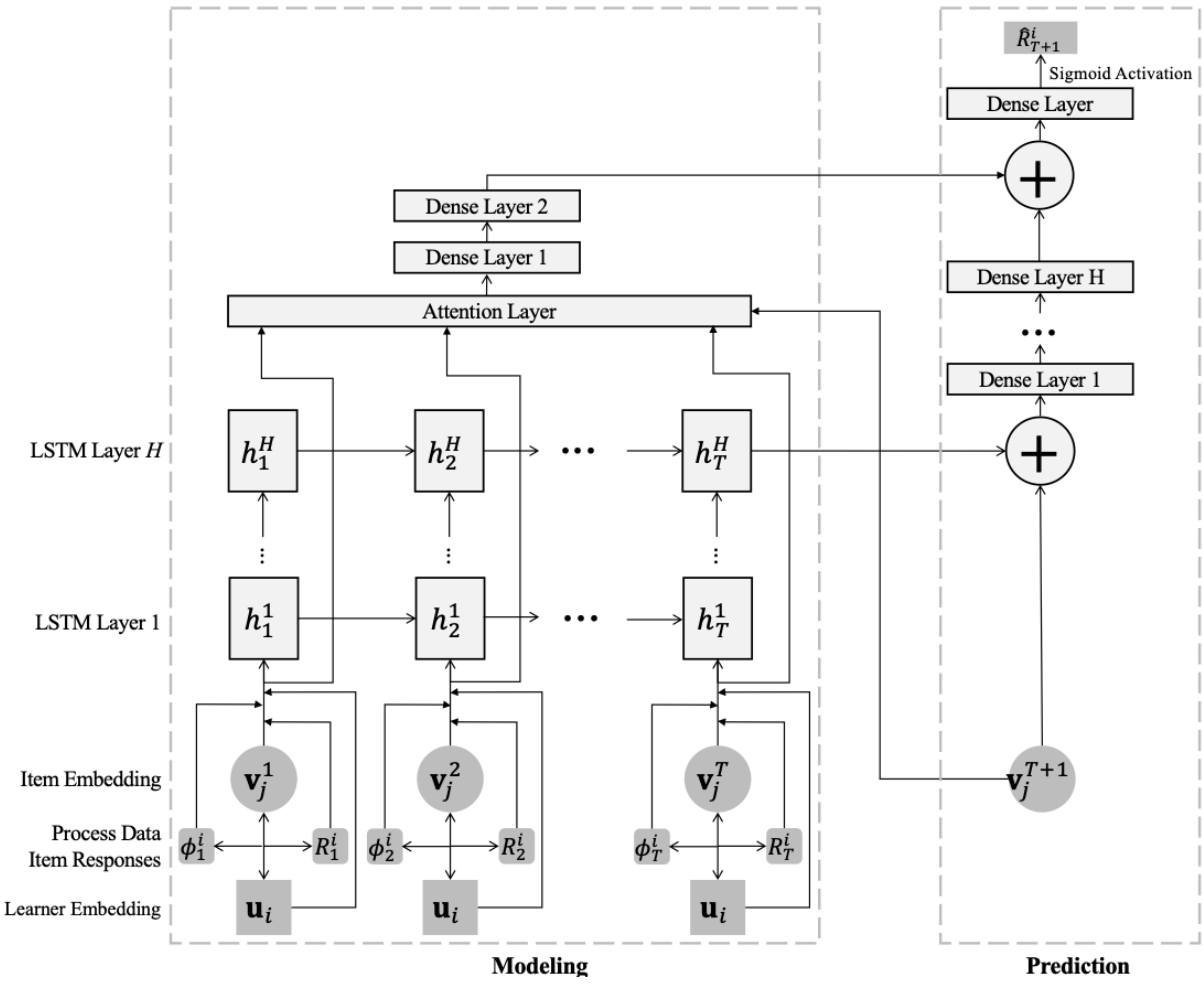
Having the item responding process $\mathbf{R}_i$ of each learner across the first T item response opportunities, our goal is to learn a model $\mathcal{M}$ which is capable of predicting $\hat{R}_{T+1}^i$ on the next item $\mathbf{n}_{T+1}^i$ at the timestep $T+1$. In the meantime, the model is capable of discovering item-skill associations based on the relevance between items.

Modeling Process of LogSDCF

Figure 22 presents the graphical representation of LogSDCF, which is of two sub-architectures: an architecture for modeling item responses, and the other for predicting future item responses. Compared to SDCF shown in Figure 13, LogSDCF mainly differs in that the inputs fed into LSTM networks for sequential modeling additionally include the learned representations of process data (i.e., actions and time) ϕ_t^i, which are produced by the process data learning architecture of LogCF (see Figure 17). When process data are available and process data learning is desirable, LogSDCF rather than SDCF should be used for sequential learning outcome modeling.

Figure 22

Graphical Representation of LogSDCF



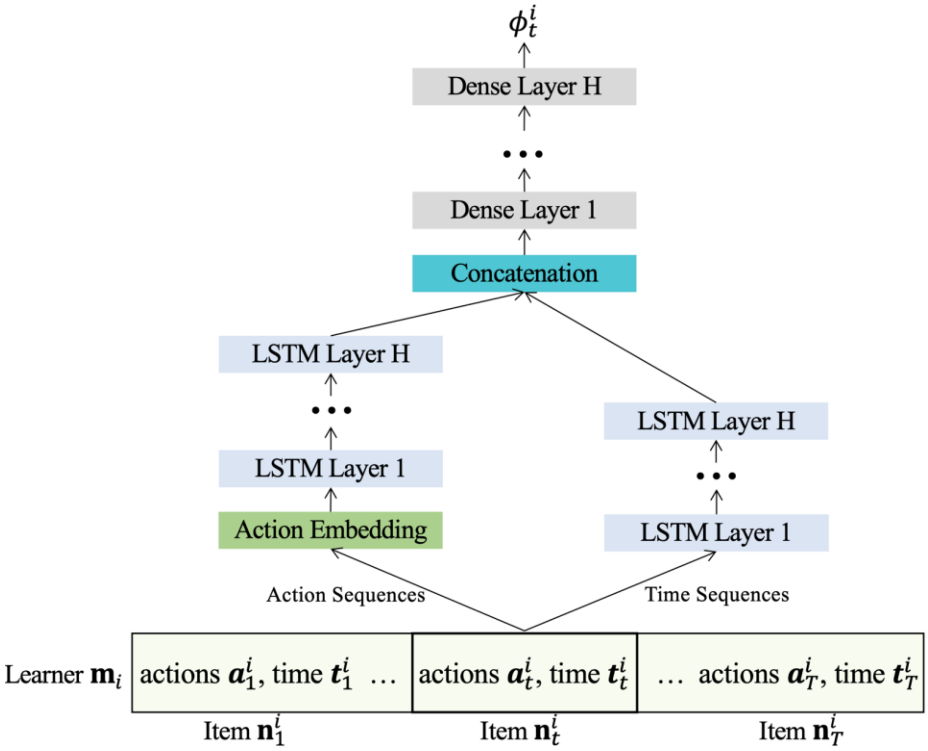
Item and Learner Embedding

Item and learning embedding in LogSDCF is identical to that of SDCF (see Chapter 3).

Deep Learning of Problem-Solving Process

Figure 23

Architecture for the Learning Process Data in LogSDCF



In addition to learner and item embeddings, the process data needs to be processed and learned for sequential modeling in LogSDCF, which borrows the deep learning architecture of LogCF (see Figure 23 and Chapter 4 for more details). At the tth timestep, learner $\mathbf{m}_i$ responding item $\mathbf{n}_t^i$ produces a sequence of problem-solving actions $\boldsymbol{a}_t^i = \{e_1, e_2, \ldots e_Q\}$ and a sequence of action-associated time durations $\boldsymbol{t}_t^i = \{t_1, t_2, \ldots t_Q\}$, where e_q and t_q indicate the qth problem-solving step and associated time duration. Given that $\boldsymbol{a}_t^i$ is a vector with categorical values, the model converts each action e_q to a dense vector of d_0 dimensions through embedding, which is then fed into an LSTM network layer for learning

the time-series dependencies between actions (see Equation 26). Similar to LogCF, LogSDCF allows for multiple LSTM layers to better capture the temporal dependencies between actions and time durations in sequences, which finally output learned representations of actions and time durations. Subsequently, the model concatenates learned representations of actions and time durations and feeds them into a deep neural network architecture for learning the interactions between actions and time durations (see Equation 36), producing a final learned representation of process data at the tth timestep ϕ_t^i.

Concatenation of Embeddings and Item Responses

Like SDCF, in LogSDCF, the item embedding $\mathbf{v}_j$, the learner embedding $\mathbf{u}_i$, the process data representation ϕ_t^i, and the item responses R_t^i need to be concatenated as inputs for sequential modeling. The model first concatenates learner and item embeddings and the process data representation. Since both learner and item embeddings have k dimensions and suppose action embedding has d_a dimensions, after concatenation, the three embeddings are combined as a $(2k + d_a)$-dimensional embedding vector, $\mathbf{e}_{ij}$. Subsequently, the model combines the concatenated vector $\mathbf{e}_{ij}$ with the item response R_t^i at timestep t with the approach shown in Equation 25.

For LogSDCF, the deep LSTM network architecture for learning the temporal dependencies between history problem-solving activities and the prediction architecture for predicting the probabilities of getting the next item correct or incorrect are the same as those of SDCF.

LogSDCF Learning

The following model parameters are to be updated in training: the embedding weights for items, learners, and actions, the weights of the deep LSTM network architecture for sequential learning, and the weights of the deep neural network architecture for prediction.

The objective function and the optimization method for learning the model weights are the same as those used for SDCF in Chapter 3.

Experimental Setup

In the following sections, experiments are conducted to evaluate the effectiveness of LogSDCF with a real-world dataset. The experiments address the following specific research questions.

- Does LogSDCF show higher predictive capacity than DKT?
- Does LogSDCF show good prediction performance at different percentages of learner fist item responses going for training?
- How interpretable are the item-skill associations estimated by LogSDCF?

Dataset Description

The real-world dataset used for evaluating LogSDCF is the same as that used for evaluating SDCF, preprocessed with the same procedure. However, unlike SDCF, LogSDCF additionally deals with learners' action and time sequences for solving each item in training and testing. The maximum action and time sequence length is fixed at six as more than 90% of items were solved with six or fewer actions.

LogSDCF Training Setting

Hyperparameter tuning is conducted as follows. For item, learner, and action embedding weights, a hyperparameter search was conducted on the following four candidate regularization weights: 0, 0.001, 0.01, and 0.1. Of these, 0.001 was selected as the finalized regularization weight. In addition, prior to each neural network layer, a dropout layer with a dropout rate of 0.5 (selected from candidate rates of 0, 0.2, and 0.5) was used to prevent overfitting. The deep LSTM network architecture contained one layer with an output dimension of five; the deep neural network architecture for prediction contained one layer with an output dimension of two. Moreover, a latent dimension of 120 was selected for

embedding items, learners, and actions. Regarding the learning rate for *Adam*, a hyperparameter search was conducted on the following four candidate learning rates: 0.0001, 0.001, 0.01, and 0.1, with 0.0001 being selected as the finalized rate. Regarding batch sizes, a hyperparameter search was conducted on the following values: 5, 32, 64, 128, 256, and the model was finally trained for 150 epochs with a batch size of 256.

Baseline

As in the previous two chapters, LogSDCF is compared with DKT to evaluate its effectiveness and predictive capacities. Similar to the study for SDCF, this study compares LogSDCF with its two variants, LogSDCF-Attention and LogSDCF-LSTM, which are two sub-architecture of LogSDCF.

Evaluation

The training/test partition rates and evaluation metrics for evaluating LogSDCF are the same as the ones used for SDCF.

Experimental Results

Main Prediction Results

Table 10 shows the testing performance of each model across different training/test partition ratios. Generally, disregarding the training/test partition ratios, LogSDCF demonstrates higher ACC and AUC rates and lower MAE and RMSE rates than DKT and SDCF. Moreover, using more history items for training slightly improves the prediction accuracy of LogSDCF, shown by slightly higher ACC and AUC rates and slightly lower MAE and RMSE rates.

Regarding the comparison between LogSDCF and its two variants, it is evident that LogSDCF has a similar or higher prediction performance than its two variants. However, the attention sub-architecture slightly outperforms the LSTM sub-architecture.

Table 10

Model Prediction Performance of LogSDCF for the Real-World Dataset

Model	ACC	AUC	MAE	RMSE
Training ratio: 0.7				
DKT	0.7037	0.7157	0.3786	0.4339
SDCF	0.7143	0.7347	0.3583	0.4298
LogSDCF	**0.7225**	**0.7400**	**0.3580**	**0.4254**
LogSDCF-Attention	0.7219	0.7395	0.3583	0.4258
LogSDCF-LSTM	0.6909	0.6928	0.4065	0.4419
Training ratio: 0.5				
DKT	0.6890	0.6974	0.3739	0.4422
SDCF	0.7076	0.7266	0.3587	0.4342
LogSDCF	**0.7160**	**0.7323**	**0.3578**	**0.4300**
LogSDCF-Attention	0.7160	0.7309	0.3589	0.4305
LogSDCF-LSTM	0.6904	0.6946	0.4028	0.4408
Training ratio: 0.3				
DKT	0.6672	0.6439	0.3764	0.4748
SDCF	0.7065	0.7182	0.3595	0.4382
LogSDCF	**0.7126**	0.7259	**0.3616**	**0.4330**
LogSDCF-Attention	0.7118	**0.7261**	0.3617	0.4335
LogSDCF-LSTM	0.6847	0.6882	0.4093	0.4453

Note. ACC = Accuracy; AUC = Area under the ROC Curve; MAE = Mean Absolute Error; RMSE = Root Mean Square Error. Values in bold represent the metric of the optimum model of the ones compared.

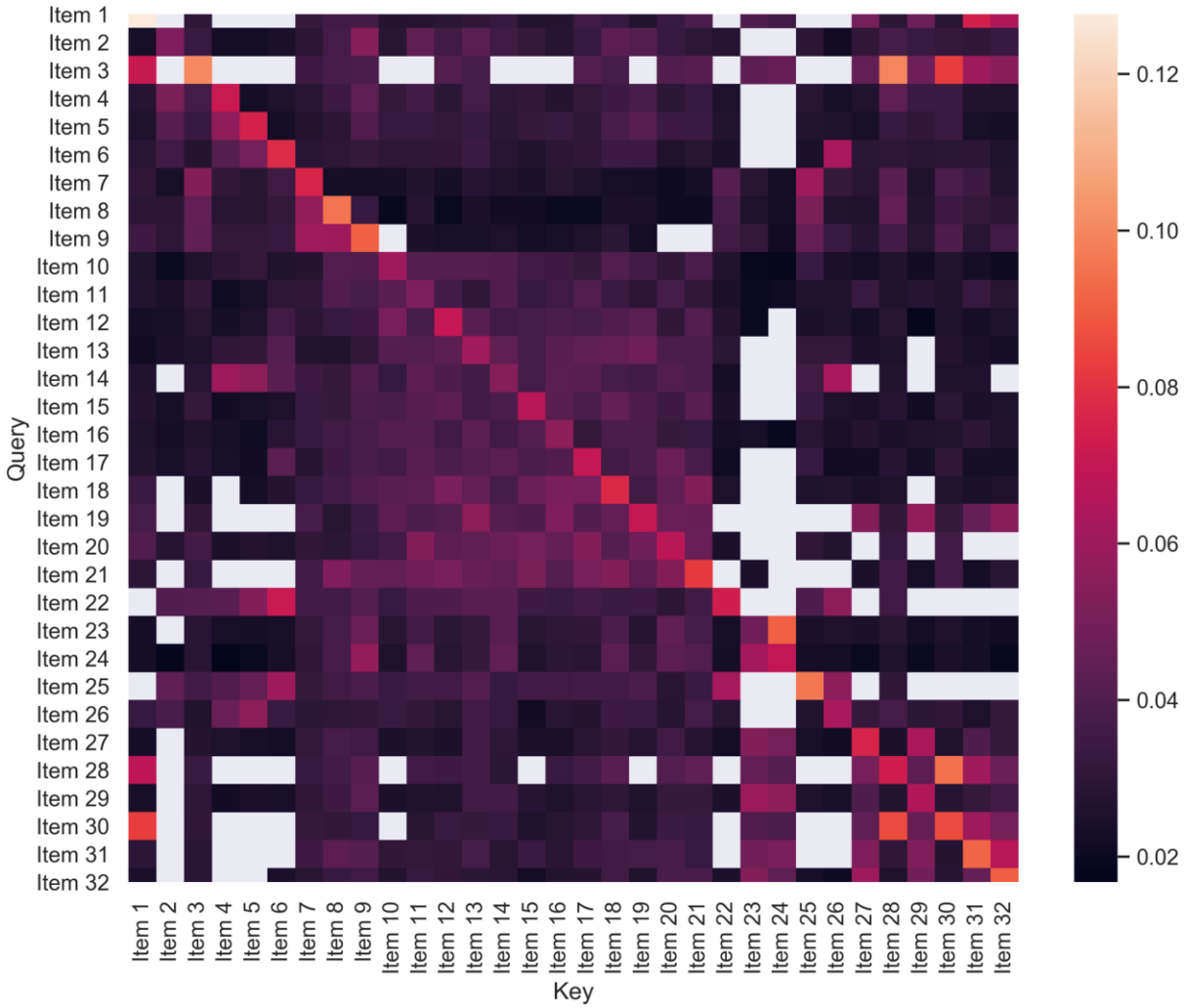
Item-Skill Associations Discovered by LogSDCF

LogSDCF discovers item-skill associations in the same way as SDCF did. It is evident that items 10 to 21 constitute a major cluster given their stronger relevance weights between each other. According to Table 5 in Chapter 3, items 10 to 21 measure the same skill labelled "equivDragFract". Therefore, their associations were correctly discovered by SDCF. For the original skills measured by only one or two items, similar to SDCF, LogSDCF could not accurately identify their differences. As discussed earlier, this might be due to skills measured by one or two items being not adequately exercised by learners, which resulted in more randomness for calculating the relevance weights. In addition, the "true" item-skill associations for this dataset are unknown so the clustering of skills by LogSDCF cannot be fully validated. In general, similar to SDCF, the capacity of LogSDCF to discover item-skill

associations is to some extent justified given that the major skill measured by most items was successfully identified.

Figure 24

Heatmap of Item Relevance Weights Estimated by LogSDCF



Note. The item and skill names are presented in Table 5.

Chapter 6 Discussion[4]

In this work, based on deep learning and CF, I developed three models for learning outcome modeling with a specific focus on process data learning. Particularly, in addition to predicting learner performance, the three generic models were capable of discovering item-skill associations without expert input. In practice, SDCF is suitable for the scenario where only product data is available and item responses are in a sequential form; LogCF is suitable for the scenario where both product and process data are available and item responses are not in a sequential form, and LogSDCF is suitable for the scenario where both product and process data are available and item responses are in a sequential form. Moreover, it is worth noting that the proposed models, especially LogCF, are not only techniques for educational data mining, but also have the potential for psychometric analysis given their interpretability.

In practice, successful applications of the proposed approaches require the following assumptions to be met. First, enough data for model training is needed. To make the models more effective and generalizable, they are expected to see a large variety of learners and items during training. Only thus can the models perform well on unseen learners and items. Second, model parameters need to be well tuned in training. An overfitting model usually performs inadequately on the testing data, and an under-fitting model suffers from high bias in training and does not generalize well on other data as well. Third, assessment items are expected to be well designed. Although the estimated item-skill associations by the proposed approaches can be used to partially inform the quality of the assessment design (e.g., items without connections to any skills or skills measured by very few items indicates possible design issues), a large number of low-quality items might result in unreliable models for learning evaluation. That said, the majority of assessment items are supposed to strongly

[4] Part of this chapter was published by the author. See "LogCF: Deep collaborative filtering with process data for enhanced learning outcome modeling", F. Chen and Y. Cui, 2020, Journal of Educational Data Mining, 12, pp. 66–99. https://doi.org/10.5281/zenodo.4399685

measure the learning topics and accurately differentiate learners of different skill levels. Finally, the models should be applied in an appropriate context. For example, the models should be used with a diagnostic purpose for promoting learning rather than with a summative purpose for comparing learning outcomes with standards or benchmarks. Moreover, in the context where learners' problem-solving actions are treated as their final explicit learning outcomes, the models should not be used because, in this case, learning outcomes are used as both targets and features in training. For example, if an item is scored by directly comparing learners' action sequences against the pre-specified correct action sequence, the model would always achieve a perfect prediction, which is useless in reality. It should be noted that violations of the above-mentioned assumptions are problematic in applying the proposed approaches to analyze real-world assessment data.

Next, the implications, limitations, and future directions of the proposed approaches will be discussed.

Theoretical Implications

Theoretically, LogCF demonstrates the potential of deep CF for recovering psychometric measurement models as a special case with model regularizations. Although the dissertation does not focus on how to fully recover IRT models from deep CF with process data learning as did in previous studies (e.g., Bergner et al., 2012), the interpretability of estimated item-skill associations evidenced by the comparison with IRT model parameters supports that deep CF can recover features of psychometric models. Admittedly, item-skill associations estimated by the proposed models do not share the same interpretation as item difficulties and item discriminations by IRT given that their estimations are also affected by process data learning and thus might reflect some information about learner problem-solving processes. Although the learned representations of process data cannot be explicitly interpreted, process data learning of the proposed models should be considered as a

regularization technique because it brings much more extra information for model training. Therefore, learning deep CF models is much less likely to be an ill-posed problem.

Regarding SDCF and LogSDCF, essentially, they are analogous to ensemble learning techniques which aggregate the advantages of many different algorithms for improved prediction performance. Specifically, the effectiveness of SDCF and LogSDCF relies on both recurrent neural networks and the attention mechanism. Attentive models have been successfully applied in the areas of machine translation and knowledge tracing (Vaswani et al., 2017; Pandey & Karypis, 2019) and they showed higher prediction accuracy than conventional deep learning approaches. This work differs from previous work on attentive modeling in that LSTM networks are integrated with the attentive model for improved prediction performance. This can be validated by the findings that SDCF and LogSDCF outperformed their LSTM or attention sub-architectures. In summary, this work demonstrates the potential of ensemble learning for enhanced learning outcome modeling.

Practical Implications

Significance of Process Data Learning

LogCF and LogSDCF are both generic systems for modeling and predicting learning outcomes based on deep CF with process data learning. To demonstrate the usefulness of LogCF and LogSDCF, I compared the effectiveness in missing response prediction between them and conventional data mining and psychometric models using data sets from an international large-scale complex problem-solving assessment and a web-tutoring system. The experimental results with the real-world datasets validated the effectiveness and interpretability of LogCF and LogSDCF.

As mentioned earlier, despite the existing machine learning-based approaches, this work argues the importance of incorporating learner process data in learning outcome modeling. As the results suggest, the variants of LogCF outperform NeuralCF (i.e., the model

without process data learning) and LogSDCF outperforms SDCF, indicating that process data learning helps refine and improve training and prediction. For example, by modeling learners' problem-solving processes, LogCF and LogSDCF might be more capable of differentiating a correct response produced by a learner with 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 and LogSDCF is especially beneficial for personalized learning. For example, in intelligent tutoring systems, it is often the case that feedback or a hint is given when learners give incorrect responses while solving a problem (Psotka et al., 1988). However, the same incorrect responses might be associated with very different underlying problem-solving processes, which affect the diagnosis of learners' mastery of latent skills. In this sense, the proposed models are more efficient and accurate for cognitive diagnosis by exploiting the process data.

Significance of Item-Skill Association Discovery

All three models are capable of learning item-skill associations from scratch without expert input. Specifically, it was found that item-skill associations discovered by LogCF were not worse than the expert-specified ones. The comparison between the three variants of LogCF suggests that LogCF performed well regardless of whether the expert-specified item-skill associations are available or not. Regarding SDCF and LogSDCF, they were both capable of detecting the major item clusters in terms of item-skill associations. Particularly, given the synthetic data, item-skill associations discovered by SDCF were almost the same as the ground truth. The capacities of the proposed approaches to discover item-skill associations were particularly promising for large-scale assessments. In the scenarios where numerous items are automatically generated by machine for computer-based assessments, using the proposed approaches, experts' efforts in specifying which items map onto which skills might be minimized given that item-skill associations can be automatically learned by

each model. Consequently, the development of large-scale assessments will be much more cost-effective.

Generalizability for Extensive Applications

The proposed approaches were developed to promote learning as a formative assessment tool but not evaluating learning as a summative assessment tool. The proposed approaches have great potential for a wide range of applications across different domains. Although LogCF, SDCF, and LogSDCF were evaluated from perspectives of educational data mining and psychometric measurement, they are generic and flexible frameworks 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., 2019). The proposed approaches 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. The proposed approaches are capable of modeling learners' performance using past learning opportunities to predict their performance in the future, which facilitates individualized learning path recommendations with their estimated mastery levels of latent skills and learning outcomes. However, it should be noted that experimental studies might be needed to validate the advantages of machine-recommended learning paths over conventional pre-planned learning paths for improved learning outcomes.

Limitations and Future Directions

Although the proposed approaches do not see any action sequences from the test sample in training, in the test stage, actions and durations on new items are needed for predicting unseen responses, which could be a limitation in practice. However, the models could still have important practical applications in the following circumstances. First, in the psychometric 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" exploiting process data for modeling. Second, in the circumstance that predictions of future item responses are desirable, the models 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 involves actions and associated time duration over a long period (i.e., from registering the course to finishing the course). In that case, with partial process data, the models 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 models could still predict unseen and future item responses only based on the deep learning-based CF architecture without process data learning.

Although process data can be used for enhanced learning outcome modeling, the process data learning of LogCF and LogSDCF is not very informative for deciphering how learners attempt problems. The process data analytics for characterizing learning behaviors was not conducted in this work mainly because the proposed approaches do not model actions and time durations separately for each item but embed them to latent representations simultaneously. 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

might confound the unique contribution of actions to item responses. Nevertheless, some recent studies focus on analyzing action sequences. 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 has demonstrated the possibility of using process data to decipher how learners attempt a problem. Instead, as mentioned earlier, the process data learning proposed in this work is more of a regularization technique for enhanced learner modeling.

This work used two different approaches to discover item-skill associations. For LogCF, the learned latent representations embedded from item IDs were directly used to indicate the strength of item-skill associations. However, the estimated item-skill associations were not in a binary or categorical scale which might not be well accepted by educational practitioners. Moreover, its interpretability was elucidated in reference to IRT item parameters, which may not very intuitive for those who are not familiar with psychometric models. For SDCF and LogSDCF, attention weights were used to calculate the relevance weights between items, and item clusters were used to indicate the item-skill mapping. In this way, latent skills were not parametrized in modeling. Therefore, the item-skill associations might be discovered with more randomness.

In addition, the proposed approaches were developed for binary item responses only. In many educational settings, non-binary scoring is used more often. How to adapt the proposed approaches for non-binary scores should be investigated in the future.

Furthermore, like some recommendation system algorithms, the proposed approaches bear 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 mitigates the cold-start problem.

Lastly, it should be noted that the proposed approaches cannot be used to completely replace human raters. Although the three models showed satisfactory prediction performance, their effectiveness still suffers from a variety of random factors such as low item quality and small samples for training which result in inevitable prediction errors. In addition, it is undoubtful that human raters such as teachers are more likely to have a more in-depth understanding of student learning, and therefore they are more likely to provide better-tailored learning interventions for improved 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, this work also stresses the importance of multi-modal learning for enhanced learning outcome modeling. For example, how to model the data from non-conventional modalities, such as video, audio, sensors, eye-trackers, and wearables, within a generic framework is worth future investigations. Moreover, how to extract interpretable learning strategies or psychological traits from multi-modal data in the context of large-scale assessments is another interesting topic in this area.

References

Abele, S. (2018). Diagnostic problem-solving process in professional contexts: Theory and empirical investigation in the context of car mechatronics using computer-generated log-files. *Vocations and Learning*, *11*(1), 133-159. https://doi.org/10.1007/s12186-017-9183-x

Aggarwal, C. C. (2016). Model-based collaborative filtering. In *Recommender systems* (pp. 71-138). Springer, Cham. https://doi.org/10.1007/978-3-319-29659-3_3

Aggarwal, C. C. (2018). *Neural networks and deep learning*. Springer.

Almutairi, F. M., Sidiropoulos, N. D., & 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. https://doi.org/10.1109/JSTSP.2017.2705581

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... & Chen, J. (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML'16: Proceedings of the 33rd international conference on international conference on machine learning* (pp. 173-182).

Azevedo, R., Johnson, A., Chauncey, A., & Burkett, C. (2010). Self-regulated learning with MetaTutor: Advancing the science of learning with MetaCognitive tools. In M. Khine & I. Saleh (Eds.), *New science of learning* (pp. 225-247). Springer.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Baker, E. L., & Mayer, R. E. (1999). Computer-based assessment of problem solving. *Computers in Human Behavior*, *15*(3-4), 269-282. https://doi.org/10.1016/S0747-5632(99)00023-0

Baker, R. S., & Yacef, K. (2009). The state of educational data mining in 2009: A review and future visions. *JEDM| Journal of Educational Data Mining*, *1*(1), 3-17. https://doi.org/10.5281/zenodo.3554657

Bell, S. (2010). Project-based learning for the 21st century: Skills for the future. *The Clearing House*, *83*(2), 39-43. https://doi.org/10.1080/00098650903505415

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157-166. https://doi.org/10.1109/72.279181

Bergner, Y., Droschler, S., Kortemeyer, G., Rayyan, S., Seaton, D., & 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* (pp. 95-102). International Educational Data Mining Society.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Black, P., & Wiliam, D. (2009). Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability*, *21*(1), 5-31. https://doi.org/10.1007/s11092-008-9068-5

Blanchard, E. G., Wiseman, J., Naismith, L., & Lajoie, S. P. (2012). A realistic digital deteriorating patient to foster emergency decision-making skills in medical students. In *12th IEEE International Conference on Advanced Learning Technologies* (pp. 74–76). Rome, Italy: IEEE Communications Society. https://doi.org/10.1109/ICALT.2012.44

Botelho, A. F., Baker, R. S., & Heffernan, N. T. (2017). Improving sensor-free affect detection using deep learning. In *International Conference on Artificial Intelligence in Education* (pp. 40-51). Springer. https://doi.org/10.1007/978-3-319-61425-0_4

Braun, H., Bejar, I. I., & Williamson, D. M. (2006). Rule-based methods for automatic scoring: Application in a licensing context. In D. M. Williamson, R. J. Mislevy, & I. I. Bejar (Eds.), *Automated scoring for complex constructed response tasks in computer-based testing* (pp. 83-122). Mahwah, NJ: Lawrence Erlbaum.

Brennan, R. L. (Ed.). (2006). *Educational measurement* (4th ed.). Westport, CT: Praeger.

Carbonell, J. R. (1970). AI in CAI: An artificial-intelligence approach to computer-assisted instruction. *IEEE Transactions on Man-machine Systems*, *11*(4), 190-202. https://doi.org/10.1109/TMMS.1970.299942

Carrillo-de-la-Pena, M. T., Bailles, E., Caseras, X., Martínez, À., Ortet, G., & Pérez, J. (2009). Formative assessment and academic achievement in pre-graduate students of health sciences. *Advances in Health Sciences Education*, *14*(1), 61-67. https://doi.org/10.1007/s10459-007-9086-y

Carroll, R. J., Primo, D. M., & Richter, B. K. (2016). Using item response theory to improve measurement in strategic management research: An application to corporate social responsibility. *Strategic Management Journal*, *37*(1), 66-85. https://doi.org/10.1002/smj.2463

Cartwright, G. F., & Derevensky, J. L. (1975). An attitudinal study of computer-assisted testing as a learning method. *Psychology in the Schools, 13*(3), 317–321.

Cen, H., Koedinger, K., & Junker, B. (2005). Automating cognitive model improvement by A* search and logistic regression. In *Proceedings of AAAI 2005 Educational Data Mining Workshop*. https://www.aaai.org/Papers/Workshops/2005/WS-05-02/WS05-02-007.pdf

Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis – a general method for cognitive model evaluation and improvement. In *Intelligent tutoring systems. ITS*

2006. Lecture notes in computer science, vol 4053 (pp. 164-175). Springer. https://doi.org/10.1007/11774303_17

Chaplot, D. S., MacLellan, C., Salakhutdinov, R., & Koedinger, K. (2018). Learning cognitive models using neural networks. In *International Conference on Artificial Intelligence in Education* (pp. 43-56). Springer. https://doi.org/10.1007/978-3-319-93843-1_4

Chatzopoulou, D. I., & Economides, A. A. (2010). Adaptive assessment of student's knowledge in programming courses. *Journal of Computer Assisted Learning*, *26*(4), 258-269. https://doi.org/10.1111/j.1365-2729.2010.00363.x

Chen, F., & Cui, Y. (2020). LogCF: Deep collaborative filtering with process data for enhanced learning outcome modeling. *Journal of Educational Data Mining, 12*(4), 66–99. https://doi.org/10.5281/zenodo.4399685

Chen, Y., Li, X., Liu, J., & Ying, Z. (2019). Statistical analysis of complex problem-solving process data: An event history analysis approach. *Frontiers in psychology*. https://doi.org/10.3389/fpsyg.2019.00486

Cheng, S., Liu, Q., Chen, E., Huang, Z., Huang, Z., Chen, Y., ... & 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* (pp. 2397-2400). Association for Computing Machinery. https://doi.org/10.1145/3357384.3358070

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., ... & Xie, W. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, *15*(141), 20170387. https://doi.org/10.1098/rsif.2017.0387

Chiu, C. Y. (2013). Statistical refinement of the Q-matrix in cognitive diagnosis. *Applied Psychological Measurement*, *37*(8), 598-618. https://doi.org/10.1177/0146621613488436

Chollet, F. (2015). *Keras: Deep learning library for theano and tensorflow*. https://keras.io

Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3642-3649). IEEE. https://doi.org/10.1109/CVPR.2012.6248110

Coelho, O. B., & Silveira, I. (2017). Deep learning applied to learning analytics and educational data mining: A systematic literature review. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (pp. 143-152). http://dx.doi.org/10.5753/cbie.sbie.2017.143

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, *4*(4), 253-278. https://doi.org/10.1007/BF01099821

Cui, Y., Chu, M. W., & Chen, F. (2019). Analyzing student process data in game-based assessments with Bayesian knowledge tracing and dynamic Bayesian networks. *Journal of Educational Data Mining*, *11*(1), 80-100. https://doi.org/10.5281/zenodo.3554751

Daniel, J., Cano, E. V., & Cervera, M. G. (2015). The future of MOOCs: Adaptive learning or business model?. *International Journal of Educational Technology in Higher Education*, *12*(1), 64-73. https://doi.org/10.7238/rusc.v12i1.2475

Davies, J., & Ecclestone, K. (2008). 'Straitjacket'or 'springboard for sustainable learning'? The implications of formative assessment practices in vocational learning cultures. *The Curriculum Journal*, *19*(2), 71-86. https://doi.org/10.1080/09585170802079447

De Champlain, A. F. (2010). A primer on classical test theory and item response theory for assessments in medical education. *Medical education, 44*(1), 109-117. https://doi.org/10.1111/j.1365-2923.2009.03425.x

de Klerk, S., Veldkamp, B. P., & 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. https://doi.org/10.1016/j.compedu.2014.12.020

de la Torre, J., van der Ark, L. A., & Rossi, G. (2018). Analysis of clinical data from a cognitive diagnosis modeling framework. *Measurement and Evaluation in Counseling and Development*, *51*(4), 281-296. https://doi.org/10.1080/07481756.2017.1327286

Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, *7*(3-4), 197-387. http://dx.doi.org/10.1561/2000000039

Desmarais, M. (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* (pp. 41-50). International Educational Data Mining Society.

Desmarais, M. C. (2012). Mapping question items to skills with non-negative matrix factorization. *ACM SIGKDD Explorations Newsletter*, *13*(2), 30-36. https://doi.org/10.1145/2207243.2207248

Desmarais, M. C., & Baker, R. S. d. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, *22*(1-2), 9-38. https://doi.org/10.1007/s11257-011-9106-8

Desmarais, M. C., & Naceur, R. (2013). A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In *International Conference on Artificial Intelligence in Education* (pp. 441-450). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-39112-5_45

DiCerbo, K. E. (2014). Game-based assessment of persistence. *Educational Technology & Society, 17* (1), 17–28.

Ding, C., Li, T., Peng, W., & 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* (pp. 126-135). Association for Computing Machinery. https://doi.org/10.1145/1150402.1150420

Divjak, B., & Tomić, D. (2011). The impact of game-based learning on the achievement of learning goals and motivation for learning mathematics-literature review. *Journal of Information and Organizational Sciences*, *35*(1), 15-30.

Dong, X., Yu, L., Wu, Z., Sun, Y., Yuan, L., & Zhang, F. (2017). A hybrid collaborative filtering model with deep structure for recommender systems. In *Proceedings of the thirty-first AAAI conference on artificial intelligence* (pp. 1309-1315).

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, *12*(2011), 2121-2159.

Durand, G., Belacel, N., & Goutte, C. (2015). Evaluation of expert-based Q-matrices predictive quality in matrix factorization models. In *Design for teaching and learning in a networked world* (pp. 56-69). Springer, Cham. https://doi.org/10.1007/978-3-319-24258-3_5

Elkahky, A. M., Song, Y., & 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* (pp. 278-288). International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/2736277.2741667

Erhel, S., & Jamet, E. (2013). Digital game-based learning: Impact of instructions and feedback on motivation and learning effectiveness. *Computers & Education*, *67*, 156-167. https://doi.org/10.1016/j.compedu.2013.02.019

Feng, M., Heffernan, N., & Koedinger, K. (2009). Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, *19*(3), 243-266. https://doi.org/10.1007/s11257-009-9063-7

Gaydos, M. J. (2016). Developing a geography game for Singapore clasrooms. In C.-K. Looi, J. L. Polman, U. Cress, & P. Reimann (Eds.), *Transforming learning, empowering learners: The International Conference of the Learning Sciences (ICLS)* (Vol. 2, pp. 729–736). Singapore: International Society of the Learning Sciences.

Gikandi, J. W., Morrow, D., & Davis, N. E. (2011). Online formative assessment in higher education: A review of the literature. *Computers & Education*, *57*(4), 2333-2351. https://doi.org/10.1016/j.compedu.2011.06.004

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256).

Gobert, J. D., Sao Pedro, M. A., Baker, R. S., Toto, E., & Montalvo, O. (2012). Leveraging educational data mining for real-time performance assessment of scientific inquiry skills within microworlds. *Journal of Educational Data Mining*, *4*(1), 104-143. https://doi.org/10.5281/zenodo.3554645

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, *4*(2), 133-151.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Greiff, S., Kretzschmar, A., Müller, J. C., Spinath, B., & Martin, R. (2014). The computer-based assessment of complex problem solving and how it is influenced by students' information and communication technology literacy. *Journal of Educational Psychology*, *106*(3), 666-680.

Greiff, S., Niepel, C., Scherer, R., & Martin, R. (2016). Understanding students' performance in a computer-based assessment of complex problem solving: An analysis of behavioral data from computer-generated log files. *Computers in Human Behavior*, *61*, 36-46. https://doi.org/10.1016/j.chb.2016.02.095

Greiff, S., Wüstenberg, S., & 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. https://doi.org/10.1016/j.compedu.2015.10.018

Greiff, S., Wüstenberg, S., & 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. https://doi.org/10.1016/j.compedu.2015.10.018

Haertel, E. H. (1989). Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement, 26,* 301-323. https://doi.org/10.1111/j.1745-3984.1989.tb00336.x

Halverson, R., & Owen, V. E. (2014). Game-based assessment: an integrated model for capturing evidence of learning in play. *International Journal of Learning Technology*, *9*(2), 111-138. https://doi.org/10.1504/IJLT.2014.064489

Hansen, M., Cai, L., Monroe, S., & 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. https://doi.org/10.1111/bmsp.12074

Hartz, S. (2002). *A Bayesian framework for the unified model for assessing cognitive abilities: blending theory with practicality* [Unpublished doctoral dissertation]. University of Illinois at Urbana-Champaign.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182). International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3038912.3052569

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173-182). International World Wide Web Conferences Steering Committee. https://doi.org/10.1145/3038912.3052569

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*(3), 197-243. https://doi.org/10.1023/A:1022623210503

Henson, R. A., Templin, J. L., & Willse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, *74*(2), 191-210. https://doi.org/10.1007/s11336-008-9089-5

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, *5*, 1457-1469.

Hsiao, H. S., & Chen, J. C. (2016). Using a gesture interactive game-based learning approach to improve preschool children's learning performance and motor skills. *Computers & Education*, *95*, 151-162. https://doi.org/10.1016/j.compedu.2016.01.005

Hwang, G. J., & Wu, P. H. (2012). Advancements and trends in digital game-based learning research: a review of publications in selected journals from 2001 to 2010. *British Journal of Educational Technology*, *43*(1), E6-E10. https://doi.org/10.1111/j.1467-8535.2011.01242.x

Iseli, M. R., Koenig, A. D., Lee, J. J., & Wainess, R. (2010). Automatic Assessment of Complex Task Performance in Games and Simulations. CRESST Report 775. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*.

Jackson, G. T., & McNamara, D. S. (2013). Motivation and performance in a game-based intelligent tutoring system. *Journal of Educational Psychology, 105*(4), 1036–1049. https://doi.org/10.1037/a0032580

Jang, E. E. (2009). Cognitive diagnostic assessment of L2 reading comprehension ability: Validity arguments for Fusion Model application to LanguEdge assessment. *Language Testing*, *26*, 31-73. https://doi.org/10.1177/0265532208097336

Joosten-ten Brinke, D., Van Bruggen, J., Hermans, H., Burgers, J., Giesbers, B., Koper, R., & Latour, I. (2007). Modeling assessment for re-use of traditional and new types of assessment. *Computers in Human Behavior*, *23*(6), 2721-2741. https://doi.org/10.1016/j.chb.2006.08.009

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement, 25*(3), 258-272. https://doi.org/10.1177/01466210122032064

Kazanidis, I., Palaigeorgiou, G., Chintiadis, P., & Tsinakos, A. (2018). Á Pilot Evaluation of a Virtual Reality Educational Game for History Learning. In *European Conference on e-Learning* (pp. 245-253). Academic Conferences International Limited.

Kerr, D., & Chung, G. K. (2012). Identifying key features of student performance in educational video games and simulations through cluster analysis. *Journal of Educational Data Mining*, *4*(1), 144-182. https://doi.org/10.5281/zenodo.3554647

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. *The Internet and Higher Education*, *8*(1), 13-24. https://doi.org/10.1016/j.iheduc.2004.12.001

Kiili, K., & Ketamo, H. (2017). Evaluating cognitive and affective outcomes of a digital game-based math test. *IEEE Transactions on Learning Technologies*, *11*(2), 255-263. https://doi.org/10.1109/TLT.2017.2687458

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://arxiv.org/abs/1412.6980

Kleitman, S., & Costa, D. S. (2014). The role of a novel formative assessment tool (Stats-mIQ) and individual differences in real-life academic performance. *Learning and Individual Differences*, *29*, 150-161. https://doi.org/10.1016/j.lindif.2012.12.001

Kline, T. J. B. (2005). Classical test theory: Assumptions, equations, limitations, and item analyses. In T. J. B. Kline (Eds.), *Psychological testing: A practical approach to design and evaluation* (pp. 91–105). Thousand Oaks, CA: Sage. https://dx.doi.org/10.4135/9781483385693

Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, *57*(2), 1813-1824. https://doi.org/10.1016/j.compedu.2011.02.003

Koedinger, K. R., Baker, R. S. J. d., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the EDM community: The PSLC dataShop. In C. Romero, S. Ventura, M. Pechenizkiy, & R. S. J. D. Baker (Eds.), *Handbook of Educational Data Mining* (pp. 43-55). Boca Raton, FL: CRC Press.

Koenig, A. D., Lee, J. J., Iseli, M., & Wainess, R. (2010). A Conceptual Framework for Assessing Performance in Games and Simulations. CRESST Report 771. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques.* The MIT Press.

Kong, S. C., & Song, Y. (2015). An experience of personalized learning hub initiative embedding BYOD for reflective engagement in higher education. *Computers & Education*, *88*, 227-240. https://doi.org/10.1016/j.compedu.2015.06.003

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30-37. https://doi.org/10.1109/MC.2009.263

Kovanović, V., Joksimović, S., Gašević, D., Hatala, M., & Siemens, G. (2017). Content analytics: The definition, scope, and an overview of published research. In C. Lang, G. Siemens, A. Wise, & D. Gašević (Eds.), *Handbook of learning analytics* (1st ed., pp. 77–92). Edmonton: SoLAR.

Kumar, N. P., & Fan, Z. (2015). Hybrid user-item based collaborative filtering. *Procedia Computer Science*, *60*, 1453-1461. https://doi.org/10.1016/j.procs.2015.08.222

Lan, A. S., Waters, A. E., Studer, C., & 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., & Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems* (pp. 556-562). MIT Press. http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf

Leighton, J. P., Gierl, M. J., & Hunka, S. M. (2004). The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka's rule-space approach. *Journal of Educational Measurement, 41*(3), 205-237. https://doi.org/10.1111/j.1745-3984.2004.tb01163.x

Levy, R. (2013). Psychometric and evidentiary advances, opportunities, and challenges for simulation-based assessment. *Educational Assessment*, *18*(3), 182-207. https://doi.org/10.1080/10627197.2013.814517

Levy, R. (2014). Dynamic Bayesian Network Modeling of Game Based Diagnostic Assessments. CRESST Report 837. *National Center for Research on Evaluation, Standards, and Student Testing (CRESST)*.

Levy, R., & Mislevy, R. J. (2004). Specifying and refining a measurement model for a computer-based interactive assessment. *International Journal of Testing*, *4*(4), 333-369.

Li, S., Kawale, J., & 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* (pp. 811-820). https://doi.org/10.1145/2806416.2806527

Linden, G., Smith, B., & York, J. (2003). Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76-80. https://doi.org/10.1109/MIC.2003.1167344

Lindsey, R. V., Khajah, M., & Mozer, M. C. (2014). Automatic discovery of cognitive skills to improve the prediction of student learning. In Z. Ghahramani, M. Welling, C.

Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds), *Advances in neural information processing systems 27* (pp. 1386-1394). Curran Associates, Inc. http://papers.nips.cc/paper/5554-automatic-discovery-of-cognitive-skills-to-improve-the-prediction-of-student-learning.pdf

Ling, C. X., Huang, J., & Zhang, H. (2003). AUC: A better measure than accuracy in comparing learning algorithms. In *Conference of the canadian society for computational studies of intelligence* (pp. 329-341). Springer. https://doi.org/10.1007/3-540-44886-1_25

Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE Assp Magazine, 4*(2), 4-22. https://doi.org/10.1109/MASSP.1987.1165576

Liu, H., Liu, Y., & 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*. https://doi.org/10.3389/fpsyg.2018.01372

Liu, J., Xu, G., & Ying, Z. (2012). Data-driven learning of Q-matrix. *Applied Psychological Measurement*, *36*(7), 548-564. https://doi.org/10.1177/0146621612456591

Liu, Y., Tian, W., & 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. https://doi.org/10.3102/1076998615621293

Long, P., Siemens, G., Conole, G., & Gašević, D. (Eds.). (2011). *Proceedings of the 1stInternational Conference on Learning Analytics and Knowledge* (LAK '11), 27 February–1 March 2011, Banff, AB, Canada. New York: ACM.

Lord, F. M. (1952). *A theory of test scores* (Psychometric Monograph, No. 7). Psychometric Society.

López-Pastor, V. M., Pintor, P., Muros, B., & Webb, G. (2013). Formative assessment strategies and their effect on student performance and on student and tutor workload:

the results of research projects undertaken in preparation for greater convergence of universities in Spain within the European Higher Education Area (EHEA). *Journal of Further and Higher Education*, *37*(2), 163-180. https://doi.org/10.1080/0309877X.2011.644780

Ma, W., Adesope, O. O., Nesbit, J. C., & Liu, Q. (2014). Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of Educational Psychology*, *106*(4), 901-918. http://dx.doi.org/10.1037/a0037123

Macfadyen, L. P., Dawson, S., Pardo, A., & Gaševic, D. (2014). Embracing big data in complex educational systems: The learning analytics imperative and the policy challenge. *Research & Practice in Assessment*, *9*, 17-28.

Martínez-Plumed, F., Prudêncio, R. B., Martínez-Usó, A., & Hernández-Orallo, J. (2016). Making sense of item response theory in machine learning. In *Proceedings of the Twenty-second European Conference on Artificial Intelligence* (pp. 1140-1148). IOS Press. https://doi.org/10.3233/978-1-61499-672-9-1140

Mathrani, A., Christian, S., & Ponder-Sutton, A. (2016). PlayIT: Game based learning approach for teaching programming concepts. *Journal of Educational Technology & Society*, *19*(2), 5-17.

Matsuda, N., Furukawa, T., Bier, N., & Faloutsos, C. (2015). Machine beats experts: Automatic discovery of skill models for data-driven online course refinement. In *Proceedings of the 8th International Conference on Educational Data Mining* (pp. 101-108). International Educational Data Mining Society.

Meek, S. E., Blakemore, L., & Marks, L. (2017). Is peer review an appropriate form of assessment in a MOOC? Student participation and performance in formative peer review. *Assessment & Evaluation in Higher Education*, *42*(6), 1000-1013. https://doi.org/10.1080/02602938.2016.1221052

Mehta, R., & Rana, K. (2017). A review on matrix factorization techniques in recommender systems. In *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)* (pp. 269-274). IEEE. https://doi.org/10.1109/CSCITA.2017.8066567

Min, W., Frankosky, M., Mott, B. W., Rowe, J., Smith, P. A. M., Wiebe, E., ... & Lester, J. (2019). DeepStealth: game-based learning stealth assessment with deep neural networks. *IEEE Transactions on Learning Technologies*, *13*(2), 312-325. https://doi.org/10.1109/TLT.2019.2922356

Mislevy, R. J., Almond, R. G., & Lukas, J. (2004). *A brief introduction to evidence-centered design*. The National Center for Research on Evaluation, Standards, and Student Testing (CRESST). http://www.cse.ucla.edu/products/reports/r632.pdf

Mislevy, R. J., Oranje, A., Bauer, M. I., von Davier, A. A., & Hao, J. (2014). *Psychometric considerations in game-based assessment*. GlassLab Games. http://www.instituteofplay.org/work/projects/glasslab-research/

Mislevy, R. J., Steinberg, L. S., Almond, R. G. (2003). Focus article: On the structure of educational assessments. *Measurement: Interdisciplinary Research and Perspectives, 1*(1), 3–62. https://doi.org/10.1207/S15366359MEA0101_02

Mislevy, R. J., Steinberg, L. S., Breyer, F. J., Almond, R. G., & Johnson, L. (2002). Making sense of data from complex assessments. *Applied Measurement in Education*, *15*(4), 363-389. https://doi.org/10.1207/S15324818AME1504_03

Molnár, G., & Csapó, B. (2018). The efficacy and development of students' problem-solving strategies during compulsory schooling: Logfile analyses. *Frontiers in psychology*. https://doi.org/10.3389/fpsyg.2018.00302

Mullis, I. V. S., Martin, M. O., Foy, P., & Hooper, M. (2016). *TIMSS 2015 International Results in Mathematics*. Boston College, TIMSS & PIRLS International Study Center. http://timssandpirls.bc.edu/timss2015/international-results/

Mullis, I. V. S., Martin, M. O., Foy, P., & Hooper, M. (2017). *PIRLS 2016 International Results in Reading*. Boston College, TIMSS & PIRLS International Study Center. http://timssandpirls.bc.edu/pirls2016/international-results/

Nguyen, D. M., Tsiligianni, E., & Deligiannis, N. (2018). Extendable neural matrix completion. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6328-6332). IEEE. https://doi.org/10.1109/ICASSP.2018.8462164

Organisation for Economic Co-operation and Development. (2014). *PISA 2012 results: Creative problem solving*. OECD, Paris, France. https://www.oecd.org/pisa/keyfindings/pisa-2012-results-volume-V.pdf

Organisation for Economic Co-operation and Development. (2015). *PISA 2015 technical report*. OECD, Paris, France. https://www.oecd.org/pisa/data/2015-technical-report/

Pandey, S., & Karypis, G. (2019). A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*.

Pardos, Z. A., & Dadu, A. (2018). dAFM: Fusing psychometric and connectionist modeling for Q-matrix refinement. *Journal of Educational Data Mining*, *10*(2), 1-27. https://doi.org/10.5281/zenodo.3554689

Pastor, V. M. L. (2011). Best practices in academic assessment in higher education: A Case in formative and shared assessment. *Journal of Technology and Science Education*, *1*(2), 25-39. http://dx.doi.org/10.3926/jotse.20

Pásztor, A., Molnár, G., & Csapó, B. (2015). Technology-based assessment of creativity in educational context: the case of divergent thinking and its relation to mathematical

achievement. *Thinking Skills and Creativity*, *18*, 32-42. https://doi.org/10.1016/j.tsc.2015.05.004

Peat, M., & Franklin, S. (2002). Supporting student learning: the use of computer–based formative assessment modules. *British Journal of Educational Technology*, *33*(5), 515-523. https://doi.org/10.1111/1467-8535.00288

Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, *27*(3-5), 313-350. https://doi.org/10.1007/s11257-017-9193-2

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. In *Advances in neural information processing systems 28* (pp. 505-513). Curran Associates, Inc. http://papers.nips.cc/paper/5654-deep-knowledge-tracing.pdf

Plass, J., Homer, B. D., Kinzer, C. K., Chang, Y. K., Frye, J., Kaczetow, W., et al. (2013). Metrics in simulations and games for learning. In M. Seif El-Nasr, A. Drachen, & A. Canossa (Eds.), *Game analytics: Maximizing the value of player data* (pp. 694–730). London: Springer.

Poropudas, J., & Virtanen, K. (2007). Analyzing air combat simulation results with dynamic Bayesian networks. In *Proceedings of the 2007 winter simulation conference* (pp. 1370-1377). IEEE. https://doi.org/10.1109/WSC.2007.4419745

Psotka, J., Massey, L. D., & Mutter, S. A. (1988). *Intelligent tutoring systems: Lessons learned*. Psychology Press.

Python Software Foundation (2019). *Python Language Reference (Version 3.8)* [Computer Software]. Available online at: http://www.python.org (accessed November 15, 2019).

Quellmalz, E. S., Davenport, J. L., Timms, M. J., DeBoer, G. E., Jordan, K. A., Huang, C. W., & Buckley, B. C. (2013). Next-generation environments for assessing and promoting complex science learning. *Journal of Educational Psychology*, *105*(4), 1100-1114. https://doi.org/10.1037/a0032220

Quellmalz, E. S., Timms, M. J., & Buckley, B. (2010). The promise of simulation-based science assessment: The Calipers project. *International Journal of Learning Technology*, *5*(3), 243-263. https://doi.org/10.1504/IJLT.2010.037306

Quellmalz, E. S., Timms, M. J., Silberglitt, M. D., & Buckley, B. C. (2012). Science assessments for all: Integrating science simulations into balanced state science assessment systems. *Journal of Research in Science Teaching*, *49*(3), 363-393. https://doi.org/10.1002/tea.21005

Rasch, G. (1960). *Probabilistic models for some intelligence and attainment tests*. Copenhagen, Denmark: Danish Institute for Educational Research. (expanded edition, 1980. Chicago: University of Chicago Press.)

Reckase, M. D. (1997). The past and future of multidimensional item response theory. *Applied Psychological Measurement, 21*(1), 25-36. https://doi.org/10.1177/0146621697211002

Reise, S. P., & Waller, N. G. (2009). Item response theory and clinical measurement. *Annual Review of Clinical Psychology*, *5*, 27-48. https://doi.org/10.1146/annurev.clinpsy.032408.153553

Reise, S. P., Ainsworth, A. T., & Haviland, M. G. (2005). Item response theory: Fundamentals, applications, and promise in psychological research. *Current Directions in Psychological Science*, *14*(2), 95-101. https://doi.org/10.1111/j.0963-7214.2005.00342.x

Roberts-Mahoney, H., Means, A. J., & Garrison, M. J. (2016). Netflixing human capital development: Personalized learning technology and the corporatization of K-12 education. *Journal of Education Policy*, *31*(4), 405-420. https://doi.org/10.1080/02680939.2015.1132774

Rosen, Y., & Tager, M. (2014). Making student thinking visible through a concept map in computer-based assessment of critical thinking. *Journal of Educational Computing Research*, *50*(2), 249-270. https://doi.org/10.2190/EC.50.2.f

Rowe, E., Asbell-Clarke, J., Baker, R. S. (2015). Serious games analytics to measure implicit science learning. In C. S. Loh, Y. Sheng, & D. Ifenthaler (Eds.), *Serious games analytics.* Springer, Cham.

Rupp, A. A., Nugent, R., & Nelson, B. (2012). Evidence-centered design for diagnostic assessment within digital learning environments: Integrating modern psychometrics and educational data mining. *Journal of Educational Data Mining*, *4*(1), 1-10. https://doi.org/10.5281/zenodo.3554639

Rupp, A. A., Templin, J., & Henson, R. (2010). *Diagnostic measurement: Theory, methods, and applications*. New York, NY: Guilford Press.

Sahebi, S., Lin, Y. R., & 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* (pp. 502-506). International Educational Data Mining Society.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web* (pp. 285-295). Association for Computing Machinery. https://doi.org/10.1145/371920.372071

Shen, F., Liu, S., Wang, Y., Wang, L., Afzal, N., & Liu, H. (2017). Leveraging collaborative filtering to accelerate rare disease diagnosis. In *AMIA Annual Symposium Proceedings* (Vol. 2017, p. 1554). American Medical Informatics Association.

Shu, Z., Bergner, Y., Zhu, M., Hao, J., & 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.

Shu, Z., Bergner, Y., Zhu, M., Hao, J., & 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.

Shute, V. J. (2011). Stealth assessment in computer-based games to support learning. In S. Tobias & J. D. Fletcher (Eds), *Computer games and instruction Vol. 55* (pp. 503-524). Information Age Publishers.

Shute, V. J., & Becker, B. J. (2010). Prelude: issues and assessment for the 21st century. In V. J. Shute, & B. J. Becker (Eds.), *Innovative assessment for the 21st century: Supporting educational needs* (pp. 1e11). New York, NY: Springer-Verlag.

Shute, V. J., & Rahimi, S. (2017). Review of computer-based assessment for learning in elementary and secondary education. *Journal of Computer Assisted Learning*, *33*(1), 1-19. https://doi.org/10.1111/jcal.12172

Shute, V. J., & Ventura, M. (2013). *Measuring and supporting learning in games: Stealth assessment*. The MIT Press.

Shute, V. J., Hansen, E. G., & Almond, R. G. (2008). You can't fatten A hog by weighing It–Or can you? evaluating an assessment for learning system called ACED. *International Journal of Artificial Intelligence in Education*, *18*(4), 289-316.

Shute, V. J., Leighton, J. P., Jang, E. E., & Chu, M. W. (2016). Advances in the science of assessment. *Educational Assessment*, *21*(1), 34-59. https://doi.org/10.1080/10627197.2015.1127752

Shute, V. J., Masduki, I., Donmez, O., Dennen, V. P., Kim, Y. J., Jeong, A. C., & Wang, C. Y. (2010). Modeling, assessing, and supporting key competencies within game environments. In D. Ifenthaler, P. Pirnay-Dummer, & N.M. Seel (Eds.), *Computer-based diagnostics and systematic analysis of knowledge* (pp. 281-309). Springer, Boston, MA.

Shute, V. J., Ventura, M., Bauer, M., & Zapata-Rivera, D. (2009). Melding the power of serious games and embedded assessment to monitor and foster learning. In U. Ritterfeld, M. Cody, & P. Vorderer (Eds.), *Serious games: Mechanisms and effects* (pp. 295-321). Mahwah, NJ: Routledge, Taylor and Francis.

Sottilare, R., Graesser, A., Hu, X., & Holden, H. (Eds.). (2013). *Design recommendations for Intelligent Tutoring Systems*. Orlando, FL: U.S. Army Research Laboratory.

Spearman, C. (1904). General intelligence: Objectively determined and measured. *The American Journal of Psychology, 15,* 202-259.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

Stadler, M., Fischer, F., & Greiff, S. (2019). Taking a closer look: An exploratory analysis of successful and unsuccessful strategy use in complex problems. *Frontiers in psychology*. https://doi.org/10.3389/fpsyg.2019.00777

Starbird, K., Muzny, G., & Palen, L. (2012). Learning from the crowd: Collaborative filtering techniques for identifying on-the-ground Twitterers during mass disruptions.

In *Proceedings of 9th International Conference on Information Systems for Crisis Response and Management, ISCRAM* (pp. 1-10).

Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, *2009*, 421425. https://doi.org/10.1155/2009/421425

Su, Y., Liu, Q., Liu, Q., Huang, Z., Yin, Y., Chen, E., ... & Hu, G. (2018). Exercise-enhanced sequential modeling for student performance prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence* (pp. 2435-2443).

Sun, Y., Ye, S., Inoue, S., & Sun, Y. (2014). Alternating recursive method for Q-matrix learning. In *Proceedings of the 7th international conference on educational data mining* (pp. 14-20). International Educational Data Mining Society.

Suzuki, K. (2017). Overview of deep learning in medical imaging. *Radiological Physics and Technology*, *10*(3), 257-273. https://doi.org/10.1007/s12194-017-0406-5

Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson.

Tang, X., Wang, Z., Liu, J., & Ying, Z. (2019). An exploratory analysis of the latent structure of process data via action sequence autoencoder. *arXiv preprint arXiv:1908.06075*. https://arxiv.org/abs/1908.06075

Tatsuoka, K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement, 20*, 345-354. https://doi.org/10.1111/j.1745-3984.1983.tb00212.x

Tatsuoka, K. K. (1990). Toward an integration of item response theory and cognitive error diagnosis. In N. Frederiksen, R. Glaser, A. Lesgold, & M. G. Shafto (Eds.), *Diagnostic monitoring of skill and knowledge acquisition* (pp. 453–488). Lawrence Erlbaum Associates, Inc.

Templin, J. L., & Henson, R. A. (2006). Measurement of psychological disorders using cognitive diagnosis models. *Psychological Methods*, *11*, 287-305. https://doi.org/10.1037/1082-989X.11.3.287

Terzis, V., & Economides, A. A. (2011). The acceptance and use of computer based assessment. *Computers & Education*, *56*(4), 1032-1044. https://doi.org/10.1016/j.compedu.2010.11.017

Thai-Nghe, N., Drumond, L., Horváth, T., Krohn-Grimberghe, A., Nanopoulos, A., & Schmidt-Thieme, L. (2012). Factorization techniques for predicting student performance. In *Educational recommender systems and technologies: Practices and challenges* (pp. 129-153). IGI Global.

Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*(2), 26-31.

Tsai, F. H., Tsai, C. C., & Lin, K. Y. (2015). The evaluation of different gaming modes and feedback types on game-based formative assessment in an online learning environment. *Computers & Education*, *81*, 259-269. https://doi.org/10.1016/j.compedu.2014.10.013

van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 26* (pp. 2643-2651). Curran Associates, Inc. http://papers.nips.cc/paper/5004-deep-content-based-music-recommendation.pdf

van der Linden, W. J., & Glas, C. A. (2000). *Computerized adaptive testing: Theory and practice*. Boston, MA: Kluwer-Nijhoff.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

von Davier, M. (2005). *A general diagnostic model applied to language testing data* (RR-05-16). Princeton, NJ: Educational Testing Service. https://doi.org/10.1002/j.2333-8504.2005.tb01993.x

Wang, H., Wang, N., & 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* (pp. 1235-1244). Association for Computing Machinery. https://doi.org /10.1145/2783258.2783273

Wang, L., Sy, A., Liu, L., & Piech, C. (2017). Deep knowledge tracing on programming exercises. In *Proceedings of the fourth annual ACM conference on learning at scale* (pp. 201-204). https://doi.org/10.1145/3051457.3053985

Wang, P. Y., & Yang, H. C. (2012). Using collaborative filtering to support college students' use of online forum for English learning. *Computers & Education*, *59*(2), 628-637. https://doi.org/10.1016/j.compedu.2012.02.007

Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing, 25*(6), 667-675. https://doi.org/10.1016/j.jvlc.2014.09.011

West, P., Rutstein, D. W., Mislevy, R. J., Liu, J., Levy, R., Dicerbo, K. E., ... & Behrens, J. T. (2012). A Bayesian network approach to modeling learning progressions. In *Learning progressions in science* (pp. 255-292). Brill Sense.

Willmott, C. J., & 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. https://doi.org/10.3354/cr030079

Winne, P. H. (2005). A perspective on state-of-the-art research on self-regulated learning. *Instructional Science*, *33*(5), 559-565. https://doi.org/10.1007/s11251-005-1280-9

Xiong, X., Zhao, S., Van Inwegen, E. G., & Beck, J. E. (2016). Going deeper with deep knowledge tracing. In *Proceedings of the 9th international conference on educational data mining* (pp. 545-550). International Educational Data Mining Society.

Yao, L., & Boughton, K. A. (2007). A multidimensional item response modeling approach for improving subscale proficiency estimation and classification. *Applied Psychological Measurement*, *31*(2), 83-105. https://doi.org/10.1177/0146621606291559

Yeung, C. K. (2019). Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. *arXiv preprint arXiv:1904.11738*. https://arxiv.org/abs/1904.11738

Yeung, C. K., & Yeung, D. Y. (2018). Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the fifth annual ACM conference on learning at scale* (pp. 1-10). https://doi.org/10.1145/3231644.3231647

Yukselturk, E., Altıok, S., & Başer, Z. (2018). Using game-based learning with kinect technology in foreign language education course. *Journal of Educational Technology & Society*, *21*(3), 159-173.

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & 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* (pp. 353-362). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939673

Zhang, S., Yao, L., & Xu, X. (2017). AutoSVD++ An Efficient Hybrid Collaborative Filtering Model via Contractive Auto-encoders. In *Proceedings of the 40th*

International ACM SIGIR conference on Research and Development in Information Retrieval (pp. 957-960).

Zhang, Z., Zhang, Y., & Ren, Y. (2020). Employing neighborhood reduction for alleviating sparsity and cold start problems in user-based collaborative filtering. *Information Retrieval Journal*, *23*(4), 449-472. https://doi.org/10.1007/s10791-020-09378-w