Submission to the NIPS 2016 Workshop on Machine Learning for Education

Estimating student proficiency: Deep learning is not the panacea

Kevin H. Wilson
Knewton Inc.

Xiaolu Xiong
Worcester Polytechnic Inst.

Mohammad Khajah
U. of Colorado, Boulder

Robert V. Lindsey
Imagen Technologies

Siyuan Zhao
Worcester Polytechnic Inst.

Yan Karklin
Knewton Inc.

Eric G. Van Inwegen
Worcester Polytechnic Inst.

Bojian Han
Carnegie Mellon U.

Chaitanya Ekanadham
Knewton Inc.

Joseph E. Beck
Worcester Polytechnic Inst.

Neil Heffernan
Worcester Polytechnic Inst.

Michael C. Mozer
U. of Colorado, Boulder

In theoretical cognitive science, there is a tension between highly structured models whose parameters have a direct psychological interpretation and highly complex, general-purpose models whose parameters and representations are difficult to interpret. The former typically provide more insight into cognition but the latter often perform better. This tension has recently surfaced in the realm of educational data mining, where a deep learning approach to estimating student proficiency, termed *deep knowledge tracing* or *DKT* [17], has demonstrated a stunning performance advantage over the mainstay of the field, *Bayesian knowledge tracing* or *BKT* [3].

DKT [17] is a standard LSTM recurrent network architecture whose input sequence consists of the series of exercises given to a student along with a binary flag indicating whether or not not the student completed the exercise correctly. DKT's charge is to estimate the success rate for any exercise that could follow next. DKT achieves substantial improvements in prediction performance over BKT on two real-world data sets (ASSISTments, Khan Academy) and one synthetic data set which was generated under assumptions that are not tailored to either DKT or BKT. DKT achieves a reported 25% gain in prediction quality, as assessed by the signal detection discriminability measure AUC, over the best previous result on the ASSISTments benchmark.

DKT, which appeared at NIPS in 2015, made a splash in the popular press, including an article in *New Scientist* entitled, “Hate exams? Now a computer can grade you by watching you learn,” and descriptions of the work in the blogosphere (e.g., [5]). DKT also shook up the educational data mining community, which is entrenched in traditional probabilistic and statistical models, some of which—like BKT—date back over twenty years.

At the 2016 Educational Data Mining Conference, three papers were presented [11, 18, 19] that examine DKT and its relationship to traditional probabilistic and statistical models. The papers all argue that while DKT is a powerful, useful, general-purpose framework for modeling student learning, its gains do not come from the discovery of novel representations—the fundamental advantage of deep learning. Assessing student proficiency, or *knowledge tracing*, does not appear to be a domain that benefits from ‘depth’; ‘shallow’ models perform just as well and offer greater interpretability and explanatory power.

In this extended abstract, we collect the results from the three interrelated papers, with the goal of encouraging additional scrutiny in evaluating and comparing models.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

1 Modeling Student Learning

The domain we're concerned with is electronic tutoring systems which employ cognitive models to track and assess student knowledge. Beliefs about what a student knows and doesn't know allow a tutoring system to dynamically adapt its feedback and instruction to optimize the depth and efficiency of learning.

Ultimately, the measure of learning is how well students are able to apply skills that they have been taught. Consequently, student modeling is often formulated as time series prediction: given the series of exercises a student has attempted previously and the student's success or failure on each exercise, predict how the student will fare on a new exercise. Formally, the data consist of a set of binary random variables indicating whether student s produces a correct response on trial t, $\{X_{st}\}$. The data also include the labels, $\{Y_{st}\}$, which characterize the exercise. These labels might index the specific exercise, e.g., $3 + 4$ versus $2 + 6$, or they might provide a more general characterization of the exercise, e.g., *single digit addition*. In the latter case, the label denotes a *skill* that must be applied to obtain a solution. (The term skill is sometimes referred to as a *knowledge component* or *concept*.) In this article, we use the terms *exercise indexed* and *skill indexed* to refer to labels that indicate either a particular exercise or a general skill required to perform the exercise.

2 Knowledge Tracing

BKT models skill-specific performance, i.e., performance on a series of exercises that all tap the same skill. A separate instantiation of BKT is made for each skill, and a student's raw trial sequence is parsed into skill-specific subsequences that preserve the relative ordering of exercises within a skill but discard the ordering relationship of exercises across skills. For a given skill σ, BKT is trained using the data from each student s, $\{X_{st}|Y_{st} = \sigma\}$, where the relative trial order is preserved. To distinguish between absolute trial index and the relative trial index within a skill, we use t to denote the former and use i to denote the latter.

BKT is based on a theory of all-or-none human learning [2] which postulates that the knowledge state of student s following the i'th exercise requiring a certain skill, K_{si}, is binary: 1 if the skill has been mastered, 0 otherwise. BKT, formalized as a hidden Markov model, infers K_{si} from the sequence of observed responses on trials $1 \ldots i$, $\{X_{s1}, X_{s2}, \ldots, X_{si}\}$. BKT is typically specified by four parameters: $P(K_{s0} = 1)$, the probability that the student has mastered the skill prior to solving the first exercise; $P(K_{s,i+1} = 1 \mid K_{si} = 0)$, the transition probability from the not-mastered to mastered state; $P(X_{si} = 1 \mid K_{si} = 0)$, the probability of correctly *guessing* the answer prior to skill mastery; and $P(X_{si} = 0 \mid K_{si} = 1)$, the probability of answering incorrectly due to a *slip* following skill mastery. Typically, the model assumes no forgetting, i.e., K cannot transition from 1 to 0.

BKT is a highly constrained, structured model. It assumes that the student's knowledge state is binary, that predicting performance on an exercise requiring a given skill depends only on the student's binary knowledge state, and that the skill associated with each exercise is known in advance. If correct, these assumptions allow the model to make strong inferences. If incorrect, they limit the model's performance. The only way to determine if model assumptions are correct is to construct an alternative model that makes different assumptions and to determine whether the alternative outperforms BKT.

DKT is exactly such an alternative model. Rather than separating the skills, DKT models all skills jointly. The input to the model is the complete sequence of exercise-performance pairs, $\{(X_{s1}, Y_{s1})... (X_{st}, Y_{st})... (X_{sT}, Y_{sT})\}$, presented one trial at a time. As depicted in Figure 1, DKT is a recurrent neural net which takes (X_{st}, Y_{st}) as input and predicts $X_{s,t+1}$ for each possible exercise label. The model is trained and evaluated based on the match between the actual and predicted $X_{s,t+1}$ for the tested exercise ($Y_{s,t+1}$). In addition to the input and output layers representing the current trial and the next trial, respectively, the network has a hidden layer with fully recurrent connections (i.e., each hidden unit connects back to all other hidden units). The hidden layer thus serves to retain relevant aspects of the input history as they are useful for predicting future performance. The hidden state of the network can be conceived of as embodying the student's knowledge state. Piech et al. [17] used a particular type of hidden unit, called an LSTM (long short-term memory) [8], which is interesting because these hidden units behave very much like the BKT latent knowledge state, K_{si}. To briefly explain LSTM, each hidden unit acts like a memory element that can hold a bit of

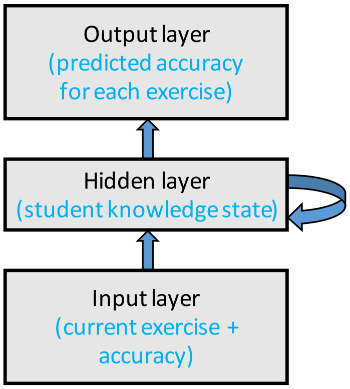

Figure 1: Deep knowledge tracing (DKT) architecture. Each rectangle depicts a set of processing units; each arrow depicts complete connectivity between each unit in the source layer and each unit in the destination layer.

information. The unit is triggered to turn on or off by events in the input or the state of other hidden units, but when there is no specific trigger, the unit preserves its state, very similar to the way that the latent state in BKT is sticky—once a skill is learned it stays learned. With 200 LSTM hidden units—the number used in simulations reported in [17]—and 50 skills, DKT has roughly 250,000 free parameters (connection strengths). Contrast this number with the 200 free parameters required for embodying 50 different skills in BKT.

With its thousand-fold increase in flexibility, DKT is a very general architecture. One can implement BKT-like dynamics in DKT with a particular, restricted set of connection strengths. However, DKT clearly has the capacity to encode learning dynamics that are outside the scope of BKT. This capacity is what allows DKT to discover structure in the data that BKT misses.

3 Data Sets and Analyses

As we explained, a separate instantiation of BKT is made for each skill, whereas DKT models all skills simultaneously. This difference leads to several subtle issues with any analysis that compares the models. As it turns out, these issues, when not properly addressed, yield results favoring DKT.

3.1 Exercises tagged with multiple skills

Xiong et al. [19] and Wilson et al. [18] re-examined one of the key data sets used to compare BKT and DKT, called ASSISTments 2009-2010 [1]. They noted that some exercises were tagged with multiple skill labels. Multiple skills were handled by replicating a record in the data base. For BKT, the data were partitioned by skill so the replicated records ended up in distinct data sets. However, for DKT and any model that processes all skills simultaneously, the model will see the same student interaction several times in a row, essentially providing the model access to ground truth when making a prediction. These duplicated rows account for approximately 25% of the data set.[1]

Xiong et al. created a new version of the data set in which multi-skill exercises were assigned a single skill label that denotes the combination of skills. DKT still significantly outperforms BKT with the corrected data set, but the magnitude of the difference shrinks, as shown in Table 1. Performance is

[1] Original and cleaned versions of the data can be found at http://tiny.cc/assistments09data.

data set	**DKT AUC score**	**BKT AUC score**
ASSISTMENTS 2009-2010 (original)	0.81	0.60
ASSISTMENTS 2009-2010 (corrected)	0.75	0.63

Table 1: Correcting data set for duplicate records

measured with the AUC score which ranges from 0.5 (no ability to discriminate correct from incorrect trials) to 1.0 (perfect ability).

3.2 Computing AUC

Khajah et al. [11] investigated two alternative methods for obtaining an AUC score. First, AUC can be calculated on a *per-skill* basis and the per-skill AUCs can be averaged to yield an overall AUC. Second, predictions may be combined *across all skills* to compute a global AUC. It appears that in the original DKT paper [17], AUC for BKT is computed on a per-skill basis whereas AUC for DKT is computed across skills. (This difference in methodology arose because BKT operates on a per-skill basis and DKT operates across skills.)

Per-skill and across-skill AUCs differ in two regards. First, per-skill AUC weighs all *skills* equally in the final computation, whereas across-skill AUC weighs all *trials* equally. Because some skills have far more trials than others, and because one would expect any model to perform more poorly on skills for which there are less data, the per-skill AUC tends to be lower than the across-skill AUC. Second, when skills are separated via the per-skill AUC, the averaged AUC score does not reflect a model's ability to predict relative accuracy of one skill versus another; in contrast, the across-skill AUC improves to the degree that model predictions capture the relative accuracy across skills. Thus, the per-skill AUC will tend to be lower, as long as a model can predict the relative difficulty of skills—the sort of base rate statistic that should be readily learned.

For these two reasons, it is unadvisable to use per-skill AUC to evaluate one model and across-skill AUC to evaluate another. Table 2 shows that when BKT is re-evaluated with across-skill AUC on two data sets, about 1/3 of the gap between DKT and BKT performance vanishes.

4 Comparisons between DKT and state-of-the-art statistical models

In the previous section, we argued that the comparison between DKT and BKT in [17] was biased in favor of DKT. Because the two source of bias were discovered by different researchers, we have not yet measured the combined effect of the two biases. Nonetheless, we believe that DKT will still outperform off-the-shelf BKT. The thrust of our research was not focused on comparing DKT to off-the-shelf BKT, but rather, to elaborations of BKT that have been proposed in the literature as well as other statistical models.

4.1 Comparison to regression models

Xiong et al. [19] compared DKT to a logistic regression model popular in the educational data mining community, *performance factors analysis* or *PFA* [16]. PFA is formulated as a skill-specific model whose regressors consist of the number of previous correct and incorrect responses on exercises that student s has previously performed requiring skill j, denoted c_{sj} and e_{sj}:

$$P(X_{si} = 1) = \text{logistic}\left(\sum_{j \in \mathbb{Q}_i} \alpha_j c_{sj} + \beta_j e_{sj} + \delta_j \right),$$

where i is the exercise label, $\mathbb{Q}_i$ is the set of skills required for exercise i, and α_j, β_j, and δ_j are model coefficients. As Table 3 summarizes, DKT is matched by PFA on two data sets, but is notably superior on one data set.

data set	**DKT** **across-skill AUC**	**BKT** **across-skill AUC**	**BKT** **per-skill AUC**
ASSISTMENTS 2009-2010 (original)	0.86*	0.73	0.67*
Piech synthetic	0.75	0.62	0.54

Table 2: Comparison of alternative methods of computing AUC score. (*These scores are based on different BKT and DKT implementations than the corresponding scores reported in Table 1.)

data set	**DKT AUC**	**PFA AUC**
ASSISTMENTS 2009-2010 (corrected)	0.75	0.73
ASSISTMENTS 2014-2015	0.70	0.69
KDD	0.79	0.71

Table 3: Comparison of DKT and PFA

4.2 Comparison to additive-factor models

Item Response Theory (*IRT*) is a standard framework for modeling student responses dating back to the 1950s [4, 14]. A single number, called the *proficiency* or *ability* represents a student's knowledge state during the course of completing several assessment. It is assumed that this proficiency is not changing during this examination.

The model assumes that many students have completed overlapping sets of exercises and assigns each student s a proficiency $\rho_s \in \mathbb{R}$. A key innovation of IRT is to model variation across different exercises. In its simplest form, each exercise i is assign a parameter δ_i representing the difficulty of the specific exercise. The probability that student s answers exercise i correctly is given by:

$$P(X_{si} = 1) = f(\rho_s - \delta_i),$$

where f is a sigmoidal function. The cumulative distribution function of the standard normal is used in the work we describe by Wilson et al. [18].

IRT has been extended in many different directions. Wilson et al. explore two extensions, *hierarchical IRT* (*HIRT*) and *temporal IRT* (*TIRT*). HIRT exploits structure among the exercises by assuming that related exercises—those tapping the same skill—will have difficulty parameters drawn from the same distribution. Essentially, individual exercises vary in difficulty, but exercises for easy skills tend to be easier and exercises for harder skills tend to be harder. HIRT corresponds to a hierarchical Bayesian model in which the mean of the δ_i distribution is in turn distributed according to a normal mean-zero hyperprior. TIRT models each ρ_s as a time-varying stochastic process. Wilson et al. selected a Wiener process.

IRT, in contrast to BKT, is fundamentally based on the notion of exercise-specific difficulties. (BKT has been extended in IRT-like directions to individuate the difficulty of exercises that tap the same skill, e.g., [6, 10, 12, 15, 20].) Nonetheless, the input label, i, used for IRT could be either exercise indexed or skill indexed. Similarly, DKT can use either exercise- or skill-indexed labels as input. Wilson et al. [18] employ a variety of representations and compare the three variants of IRT to a version of DKT with standard processing units in the recurrent layer. (Piech et al. used a recurrent net with LSTM units in the hidden layer, but found similar performance with a standard recurrent net.)

Figure 2 shows key comparisons between DKT and the three variants of IRT. ASSISTments refers to a corrected version of the 2009-2010 data set;[2] KDD is the KDD cup set also used in the comparison to PFA (Table 3); and Knewton is a proprietary data set. All variants of IRT outperform DKT, with the following caveats. For the IRT models, both exercise-indexed and skill-indexed labels were tested, and exercise-indexed labels were superior. For DKT, it was not computationally feasible to

[2] Wilson et al. corrected ASSISTments by selecting only a single skill for all exercises labeled with multiple skills, in contrast to Xiong et al., who merged multiple skills into a single meta-skill.

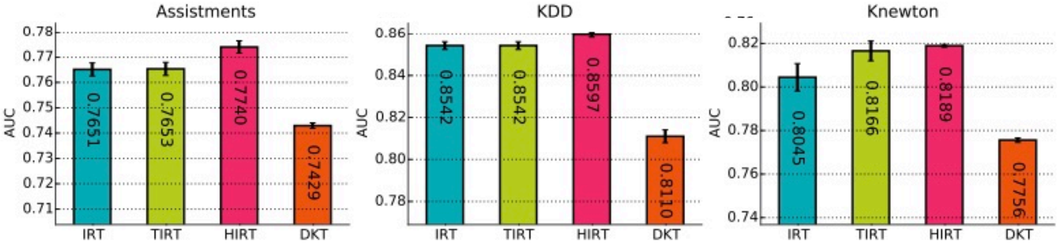

Figure 2: Comparison of DKT and variants of Item-Response Theory (IRT) including temporal IRT (TIRT) and hierarchical IRT (HIRT).

evaluate exercise-indexed labels for KDD and Knewton. Thus, for these data sets the comparison is mismatched in the information provided to the models, but one might consider them matched in terms of computational tractability. Interestingly, for ASSISTments, DKT did no better with exercise-indexed labels than with skill-indexed labels. This result may be due to the fact that exercises are presented in a fixed order for all students.

4.3 Comparison to state-of-the-art BKT

The original BKT model [3] spawned many variants and extensions that have improved the model's predictive performance. Khajah et al. [11] compared DKT to three extensions of BKT that have shown promise. These three extensions were selected based on an analysis of information that DKT can in principle exploit but which off-the-shelf BKT cannot. The three extensions are:

- *Incorporating latent student abilities.* Individuating predictions based on a student's ability is a core element of IRT, and has been incorporated into BKT [10, 12]. The latent ability is one means by which information can be shared across skills.
- *Forgetting.* The two-state learning model that underlies BKT allows for transitions both from a state of not knowing to a state of knowing as well as from a state of knowing to not knowing. The latter transition, sometimes referred to as forgetting, is dropped from most implementations of BKT, but it is a simple means by which recency effects can be modeled. DKT, like all recurrent neural nets, is very effective in acting on recent information in a sequence.
- *Skill discovery.* Automated procedures have been proposed to discover the mapping of exercises to skills from performance data [7, 13]. The approach in [13] couples BKT with a technique that searches over partitions of the exercise labels to simultaneously (1) determine which skill is required to correctly answer each exercise, and (2) model a student's dynamical knowledge state for each skill. Formally, the technique assigns each exercise label to a latent skill such that a student's expected accuracy on a sequence of same-skill exercises improves monotonically with practice according to BKT. The technique incorporates a nonparametric prior over the exercise-skill assignments that is based on expert-provided skills, when available, and a weighted Chinese restaurant process [9].

Figure 3 shows a comparison of DKT with five versions of BKT: the original, one version for each of the three extensions, and one version containing all three extensions, termed BKT+FSA. Evaluating the models on four data sets, Khajah et al. [11] found that BKT+FSA slightly outperforms DKT on two, and DKT slightly outperforms BKT+FSA on two.

5 Discussion

DKT is a powerful, general-purpose framework for modeling student proficiency. Our goal was to determine whether this powerful, general-purpose framework is *necessary* to obtain the best predictive performance on a data set. We found that while DKT does outperform the classic, twenty year old

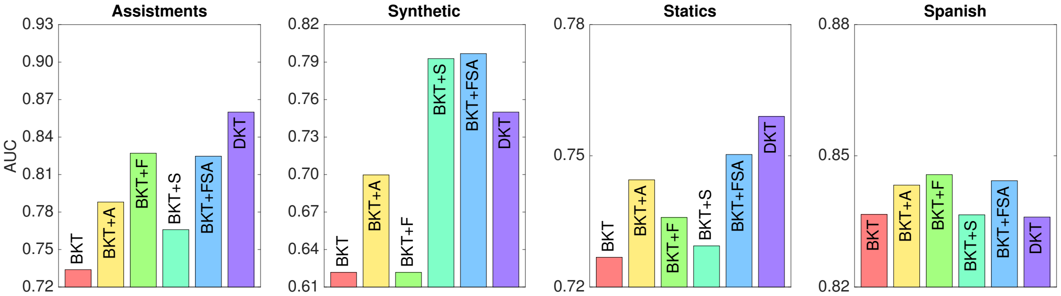

Figure 3: Comparison of DKT and extensions of BKT. BKT+A = BKT with inference of latent student abilities; BKT+F = BKT with forgetting; BKT+S = BKT with symbolic skill discovery; BKT+FSA = BKT with all three extensions.

BKT—even after several methodological issues with the original research are addressed—other models achieve a level of performance comparable to that of DKT. In particular, we investigated (1) PFA, a logistic regression model which predicts based on counts of previous successful and unsuccessful attempts to perform skill-specific exercises; (2) IRT, a model that infers latent student and exercise characteristics and is readily extended to handle structure among exercises and temporal variation in student knowledge state; (3) BKT+FSA, a state-of-the-art variant of BKT that incorporates forgetting, skill discovery, and inference of skill-invariant student ability.

Neural net methods like DKT have their strength in being able to discover novel representations through deep networks that transform representation at each stage. If this 'depth' was fully exploited by DKT, DKT should perform better than methods like PFA, IRT, and BKT+FSA which operate on predefined representations. (One might consider the skill-induction component of BKT+FSA to be a form of representation discovery, but it is an extremely simple all-or-none partitioning of exercises into skills; in contrast, neural net representations would allow each exercise to have continuously varying degree of dependence to each skill in a completely unconstrained manner.)

PFA, IRT, and BKT+FSA vary in the information they use for prediction, but they share an underlying probabilistic foundation and the fact that each model has parameters and inferred states are psychologically meaningful. DKT's flexibility comes at a price: interpretability. DKT is massive neural network model with tens of thousands of parameters which are individually impossible to interpret. Although the creators of DKT did not have to invest much up-front time analyzing their domain, they did have to invest substantive effort to understand what the model had actually learned.

For estimation of student proficiency, deep learning does not appear to be the panacea, particularly when an explicit underlying theory, explanatory power, and interpretability matter. Nonetheless, we anticipate that deep learning has a promising future in educational data mining, but that future depends on data sets that have a much richer encoding of the exercises and learning context. For example, with a deep learning approach, the input could be extended from exercise labels to the raw text of the exercise along with associated images, and even metadata concerning the placement of the exercise in the curriculum. Input could also include information about the students—their gaze pattern, state of altertness, reading and response latencies—as well as contextual information such as time of day, recent activities the student had engaged in, etc. As data sets grow in size, the potential to exploit richer data sources increases, and no explicit theory is powerful enough to predict the impact of and interactions among these many factors. When we reach this point, black-box neural net models will be needed.

References

[1] ASSISTments.org. Skill-builder data 2009-2010, 2010. [Online; accessed 24-May-2016].

[2] R. Atkinson and J. A. Paulson. An approach to the psychology of instruction. *Psychology Bulletin*, 78:49–61, 1972.

[3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Model. User-Adapt. Interact.*, 4(4):253–278, 1995.

[4] P. De Boeck and M. Wilson. *Explanatory Item Response Models: a Generalized Linear and Nonlinear Approach*. Springer-Verlag, New York, NY, 2004.

[5] R. Golden. How to optimize student learning using recurrent neural networks (educational technology). Web page, 2016. http://tinyurl.com/GoldenDKT, retrieved February 29, 2016.

[6] J. González-Brenes, Y. Huang, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *To appear in Proceedings of the Seventh International Conference on Educational Data Mining*, pages 84–91, 2014.

[7] J. P. González-Brenes. Modeling skill acquisition over time with sequence and topic modeling. In S. V. N. V. G. Lebanon, editor, *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*. JMLR, 2015.

[8] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[9] H. Ishwaran and L. James. Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13:1211–1235, 2003.

[10] M. Khajah, Y. Huang, J. P. Gonzáles-Brenes, M. C. Mozer, and P. Brusilovsky. Integrating knowledge tracing and item response theory: A tale of two frameworks. In M. Kravcik, O. C. Santos, and J. G. Boticario, editors, *Proceedings of the 4th International Workshop on Personalization Approaches in Learning Environments*, pages 7–15. CEUR Workshop Proceedings, 2014.

[11] M. Khajah, R. V. Lindsey, and M. Mozer. How deep is knowledge tracing? In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016)*, pages 94–101. International Educational Data Mining Society, 2016.

[12] M. Khajah, R. M. Wing, R. V. Lindsey, and M. C. Mozer. Incorporating latent factors into knowledge tracing to predict individual differences in learning. In J. Stamper, Z. Pardos, M. Mavrikis, and B. M. McLaren, editors, *Proceedings of the 7th International Conference on Educational Data Mining*, pages 99–106. International Educational Data Mining Society, 2014.

[13] R. V. Lindsey, M. Khajah, and M. C. Mozer. Automatic discovery of cognitive skills to improve the prediction of student learning. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1386–1394. Curran Associates, Inc., 2014.

[14] F. M. Lord. *A theory of test scores (number 7 in Psychometric Monograph)*. Psychometric Corporation, Richmond, VA, 1952.

[15] Z. A. Pardos and N. T. Heffernan. KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *User Modeling, Adaption and Personalization*, pages 243–254. Springer, 2011.

[16] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*, pages 531–538, Amsterdam, The Netherlands, The Netherlands, 2009. IOS Press.

[17] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 505–513. Curran Associates, Inc., 2015.

[18] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016)*, pages 539–544. International Educational Data Mining Society (IEDMS), 2016.

[19] X. Xiong, S. Zhao, E. V. Inwegen, and J. Beck. Going deeper with deep knowledge tracing. In T. Barnes, M. Chi, and M. Feng, editors, *Proceedings of the 9th International Conference on Educational Data Mining (EDM 2016)*, pages 545–550. International Educational Data Mining Society, 2016.

[20] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian knowledge tracing models. In *Artificial Intelligence in Education*, pages 171–180. Springer, 2013.