Dynamic Models for Knowledge Tracing & Prediction of Future Performance

Androniki Sapountzi[1] Sandjai Bhulai[2] Ilja Cornelisz[1] Chris van Klaveren[1]
[1]Vrije Universiteit Amsterdam, Faculty of Behavioral and Movement Sciences, Amsterdam Center for Learning Analytics
[2]Vrije Universiteit Amsterdam, Faculty of Science, Department of Mathematics
Email addresses: a.sapountzi@vu.nl, s.bhulai@vu.nl, i.cornelisz@vu.nl, c.p.b.j.van.klaveren@vu.nl

Abstract— **Large-scale data about learners' behavior are being generated at high speed on various online learning platforms. Knowledge Tracing (KT) is a family of machine learning sequence models that are capable of using these data efficiently with the objective to identify the likelihood of future learning performance. This study provides an overview of KT models from a technical and an educational point of view. It focuses on data representation, evaluation, and optimization, and discusses the underlying model assumptions such that the strengths and weaknesses with regard to a specific application become visible. Based on the need for advanced analytical methods suited for large and diverse data, we briefly review big data analytics along with KT learning algorithms' efficiency, learnability and scalability. Challenges and future research directions are also outlined. In general, the overview can serve as a guide for researchers and developers, linking the dynamic knowledge tracing models and properties to the learner's knowledge acquisition process that should be accurately modeled over time. Applied KT models to online learning environments hold great potential for the online education industry because it enables the development of personalized adaptive learning systems.**

Keywords- big data applications; educational data mining; knowledge tracing; sequential supervised machine learning.

I. INTRODUCTION

Big Data Analytics (BDA) is becoming increasingly important in the field of online education. Massive Open Online Courses (i.e. Coursera), Learning Management Systems (i.e. Moodle), social networks (i.e. LinkedIn Learning), online personalized learning platforms (i.e. Knewton), skill-based training platforms (i.e. Pluralsight), educational games (i.e. Quizlet), and mobile apps (i.e. Duolingo) are generating various types of large-scale data about learner's behaviors and their knowledge acquisition [1]–[3]. To illustrate this with an example, the 290 courses offered by MIT and Harvard in the first four years of edX produced 2.3 billion logged events from 4.5 million learners. The emerging scientific fields of educational neuroscience [4] and smart-Education [5][6], which hopefully are going to provide new insights about how people acquire skills and knowledge, indicate new big data sources in education.

Artificial Intelligence (AI), Learning Analytics (LA), and Educational Data Mining (EDM) are three areas under development oriented towards the inclusion and exploration of big data analytics in education [2][7]–[9]. EDM considers a wide variety of types of data, practices, algorithms, and methods for modeling and analysis of student data, as categorized by [1][2][10][11]. EDM, LA, AI and Big Data technologies are well-established and have progressed rapidly, however advanced analytic methods suited for large, diverse, streaming or real-time data are still being under development. A critical question in this area is whether more advanced learning algorithms or data of higher quality [12] and well pre-processed [1], or bigger datasets [8][13]–[15] are more important for achieving better analysis results. For all the above reasons, the implementation of BDA in education is considered to be both a major challenge and an opportunity in education [2][3][7]–[11][13][16][17].

Knowledge Tracing (KT) is widely applied in intelligent tutoring systems, and to other modal sources of big data [11] such as online standardized tests, Massive Open Online Courses (MOOC's) data, and educational apps. KT is an EDM framework for modeling the acquisition of student knowledge over time, as the student is observed to interact with a series of learning resources. The objective of the model is either to infer the knowledge state for the specific skill being tutored or to predict the performance on either the next learning resource in the sequence or all the learning resources. KT can be considered as a sequence machine learning model that estimates a hidden state, that is the probability that a certain concept of knowledge is acquired, based on a sequence of noisy observations, that are the interaction-performance pairs on different learning resources at consecutive trials. The estimated probability is then considered a proxy for knowledge mastery which is leveraged in recommendation engines to dynamically adapt the feedback, instruction or learning resource returned to the learner. Furthermore, KT models are applied in mastery learning frameworks which are used to estimate the moment that a certain skill is acquired by the learner [18]. These components empower the development of adaptive learning systems.

The literature distinguishes two representations of KT models: the probabilistic and the deep learning. The former models the knowledge of a learner as a binary hidden state with a level of uncertainty attached to it. The latter models the knowledge of a learner with distributed continuous hidden states that are updated in non-linear, deterministic ways. Graphical probabilistic models of Hidden Markov Models and Dynamic Bayesian Networks can be considered as the baseline models, while deep Recurrent Neural Networks models with Long Short-Term Memory (LSTM) units have only recently been employed. Throughout the paper, the differences between these modeling approaches and their impact on the educational purposes are discussed.

This study provides an overview of currently existing representations of KT models from both an educational and a technical angle. It discusses the underlying model assumptions such that the strengths and weaknesses of the

 ISBN: 978-1-61208-681-1

reviewed models are revealed. The review can serve as a guide for researchers and developers, in that when the objective is to predict future performance in online learning environments, the review is informative for which dynamic KT models should be chosen. In addition to that, we hope that by highlighting their strengths and similarities, inspiration for more sophisticated algorithms or richer data sources would be created, capable of accurately capturing the process of knowledge acquisition.

This study proceeds as follows. Section II describes the representation for the knowledge tracing along with a brief introduction behind the probabilistic and recurrent neural network sequence models. Section III introduces the baseline KT model and the other three models, after which the strengths, weaknesses, differences, and similarities are highlighted together with their intrinsic behaviors. Section IV discusses the Item Response Theory (IRT), as it is the alternative family of models for modeling and predicting knowledge acquisition. Section V discusses the prospects and challenges, and Section VI provides the conclusions.

II. DATA REPRESENTATION FOR KNOWLEDGE TRACING

Data representation refers to the choice of a mathematical structure with which to model the data or, relatedly, to the implementation of that structure. It turns a theoretical model to a learning algorithm and embodies assumptions required for the generalization [19]. If the assumptions of the representation or the explanatory factors accompanied the data are not sufficient to capture the reality and determine the right model, the algorithm will fail to generalize to new examples. Instances of assumptions could be the linear relationships of factor dependencies or a hierarchical representation of explanatory factors. A good representation is one that can express the available kind of knowledge and hence can be a useful input to the predictor [15], meaning that a reasonably-sized learned representation can capture the structure of a huge number of possible input configurations. Other elements contributing to a good representation are outlined in [19]. An interesting point to note is that predictive analytics that lies in distributed or parallel systems, a common case in BDA, the choice of representation will affect how well the data set can be decomposed into smaller components so that analysis can be performed independently on each component.

A. The Knowledge Tracing Task

In KT, two AI frameworks have been utilized to represent the different kinds of available knowledge and disentangle the underlying explanatory factors: the Bayesian (inspired by Bayesian probability theory and statistical inference) and the connectionist deep learning framework (inspired by neuroscience). Bayesian Knowledge Tracing (BKT) is the oldest and still dominant approach for modeling cognitive knowledge over time while the deep learning approach to knowledge tracing (DKT) is a state-of-the-art model.

KT in its general form is formulated as a supervised learning problem of time-series prediction. Suppose a data set *D* consisting of ordered sequences of length *T,* to be exercise-performance observation pairs $X = \{(x_{m,1}, y_{m,1}) \dots (x_{m,T}, y_{m,T})\}$ with $y_{m,t} \in \{0,1\}$ from the m - th student on trial $t \in \{1, .., T\}$. The goal is to compute the posterior probability distribution for the parameters $p(y|x;\theta)$ for student *m*.

The objective in the Bayesian approach is to estimate the probability that a student has mastered a skill S_1 based on the sequence of observed answers that tap S_1, as determined by the concept map. The prediction task in the deep learning approach is the probability that the student will answer the next exercise correctly in their next interaction while the network is presented with the whole trial sequence for all the skills practiced.

A distinction between the two approaches is located on the existence of the concept map. In BKT, the sequences of X are passed through a pre-determined concept map which is assumed to be accurately labelled by experts. The concept map represents a mapping of an exercise or a step of a learning resource to the related skills. The domain knowledge is divided into a hierarchy of relatively fine-grained component skills, also known as Knowledge Components (KC). This may include skills, concepts, or facts. The concept map is used to ensure that students master prerequisite skills before tackling higher level skills in the hierarchy [18]. In the Bayesian approach, a different model is initiated for each new skill while the prediction serves for drawing inferences about the knowledge state of a student for the skill. In the BKT, 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.

Rather than constructing a separate model for each skill, DKT model all skills jointly. In deep learning though, the sequences are not passed through a concept map, but through featurization, that is the distributed hidden units in the layers that relate the input sequences to the output sequences. This distributed featurization, which is the core of the RNN's generalizing principle, is used to induce features and hence discover the concept map and skill dependencies.

B. Probabilistic Sequence Models

The problem of knowledge tracing was first posed as a special case of Hidden Markov Models (HMM) with a straightforward application of Bayesian inference. DBN employed afterward to solve for the assumption of independence of latent states among the different skills. A DBN is a Bayesian network repeated among multiple time steps.

HMM and DBN are Probabilistic Graphical Models (PGM). In PGMs, two concepts are always present: *i)* the data or random variables are represented as nodes in a graph and *ii)* a probabilistic distribution is attached over the nodes via the edges of the graph [20]. HMM is an undirected PGM while Bayesian networks are Directed Acyclical Graphs (DAG) describing probabilistic influences between the nodes of the graph. To briefly explain the benefits of each representation, DAG are useful for expressing causal relationships between random variables, whereas undirected

 ISBN: 978-1-61208-681-1

graphs are better suited to expressing soft constraints between the latent and observed random variables [20].

HMM is used to model sequences of possible events in which the probability of each event depends only on the state attained in the previous event, i.e., *Markov processes*, with unobserved states, also called *hidden* or *latent* states. The latent variables are the discrete variables h_n describing which component of the mixture distribution is responsible for generating the corresponding observation. They can take only one value of all the possible hidden states *K* where each hidden state has got its own internal dynamics described by a transition matrix *A* describing stochastic transitions between states. The inference of the probability distribution over the hidden states allow us to predict the next output. The outputs produced by a state are stochastic and hidden, in the sense that there is no direct observation about which state produced an output, much like a student's mental process. However, the hidden states produce as observables the emission probabilities Φ that govern the distribution (i.e., actions of a learner).

The parameters that need to be evaluated and learned in HMM are $\lambda = \{\Pi, \mathrm{A}, \Phi\}$, where Π is the initial latent variable z_1 which doesn't depend on some other variable. In HMM, including DBN and all generative models, the inference problem is firstly solved: given the parameters θ and a sequence of observations *(practice attempts)* $\mathrm{X} = \{X_t\}$, $t \in \{1, .., T\}$, what is the probability that the observations are generated given the model $P(X|\,\lambda)$; and secondly the learning problem $P(\,\lambda|X)$ is solved.

In the DBN, this is equivalent to $P(X, h|\,\lambda)$, where we marginalize over the hidden states h of the latent variables. Since this is a directed graph and edges carry arrows that have directional significance, the joint distribution is given by the product over all of the nodes of the graph, whose distribution is conditioned on the variables corresponding to the parents of each node. A detailed explanation of the computations in the DBN is provided by [20][21].

C. Recurrent Neural Network Sequence Models

Deep Recurrent Neural Networks and specifically the Long Short-Term Memory (LSTM) unit was only recently employed to the KT task to solve for the binary, highly structured representation of the hidden knowledge state. Recurrent Neural Networks (RNN) are a family of Artificial Neural Networks (ANN) used for modeling sequential data that hold a temporal pattern. ANN relate the input units to the output units through a series of hidden layers, each comprising a set of hidden units. The latter is triggered to obtain a specific value by events found in the input and previous hidden states, a process implemented by a non-linear activation function.

RNNs are layered ANNs that share the same parameters, also called *weights*, through the activation function. This property is illustrated in Fig.3 with the formation of a directed circle between the hidden units. RNNs are very powerful because they combine the two following properties:

i. The distributed hidden state allows them to forget and store a lot of information about the past such that they can predict efficiently.

ii. The non-linear activation functions allow them to update their hidden state in complicated ways which can yield high-level structures found in the data.

LSTM is a type of hidden units in RNN that includes '*gates'* which let the hidden state to act as a memory able to hold bits of information for long periods of time and thus can adjust the flow of information across time. When there is no specific trigger, the unit preserves its state, very similar to the way that the latent state in HMM is sticky—once a skill is learned it stays learned [22].

III. SEQUENCE MODELS APPLIED IN KNOWLEDGE TRACING

A. Standard Bayesian KT: skill-specific discrete states

The BKT [18] includes four binary parameters that are defined in a skill-specific way. The two performance-related variables that are emitted from the model are the following:

i. S-slip, the probability that a student will make an error when the skill has been learned, and

ii. G -guess, the probability that a student will guess correctly if the skill is not learned;

The two latent and learning-related variables are the following:

i. $P(\theta_{t-1}) = P(\theta_0)$ which is the initial probability of knowing the skill a priori, and

ii. $P(T)$ which represents the transition probability of learning after practicing a specific skill on learning activities. The estimated acquired skill-knowledge, which is the probability of $P(\theta_t)$, is updated according to (1c) using the $P(T) = P(\theta_{t+1} = 1|\,\theta_t = 0)$ and observations from correct or incorrect attempts X computed either by (1a) or (1b), respectively. Equation (1d) computes the probability of a student applying the skill correctly on an upcoming practicing opportunity. The equations are as follows:

$$P(\theta_{t+1}|y_t = 1)\ \frac{P(\theta_t) * (1 - P(S))}{P(\theta_t) * \big(1 - P(S)\big) + (1 - P(\theta_t)) * (P(G))} \quad (1a)$$

$$P(\theta_{t+1}|y_t = 0) = \frac{P(\theta_t) * P(S)}{P(\theta_t) * P(S) + (1 - P(\theta_t)) * (1 - P(G))} \quad (1b)$$

$$P(\theta_{t+1}) = P(\theta_{t+1}|y_t) + \big(1 - P(\theta_{t+1}|y_t)\big) * P(T) \quad (1c)$$

$$P(C_{t+1}) = P(\theta_t) * \big(1 - P(S)\big) + \big(1 - P(\theta_t)\big) * P(G) \quad (1d)$$

At each t, a student m is practicing a step of a learning opportunity that tap a skill S. The step-by step process of a student trying to acquire knowledge about S is illustrated in Fig.1. Given a series of y_t, and t for the student m, the learning task is the likelihood maximization of the given data $P(y|\,\lambda)$, where $\lambda = \{P(S), P(G), P(T), P(\theta_t)\}$. In the original paper, this is done through Curve Fitting and evaluated via Mean Absolute Error, which is considered an insufficient measure [26]. The key idea of BKT, is that it considers guessing and slipping in a probabilistic manner to infer the current state and update the learning parameter during the practicing process.

 ISBN: 978-1-61208-681-1

Even though BKT updates the parameter estimates based on dynamic student responses, it is assumed that all of the four parameters are the same for each student. In essence, the actual probability of a correct response is averaged across students, and the models predicted probability of a correct response is averaged across skills. Because the data of all students practicing a specific skill are used to fit the BKT parameters for that skill, without conditioning on certain student's characteristics, a big part of the research is focused on adding learner-specific variables by assuming variability among students' process of learning. Yudelson [23] found that the inclusion of student-specific parameters has a significant positive effect on prediction accuracy and interpretability, as well as in dealing with over-fitting. [24] added Dirichlet priors for the initial mastery θ_{t-1}, while [23] extended their work and found that adding variables of learning rates $P(T)$ for individual learners, provides higher model accuracy.

B. *BKT with student-specific features of learning rates: Personalized predictions*

The Individualized BKT (IBKT) model [23] includes apart from skill-specific, student-specific parameters as well. The model is developed by splitting the skill-specific BKT parameters, substituted by w, into two parameters components *(i)* w^k-the skill-specific and *(ii)* w^u-the student-specific component; and combining them by summing their logit function $\mathrm{l}(p) = \log\left(\frac{p}{1-p}\right)$, and sigmoid function $\sigma(\mathrm{x}) = \frac{1}{(1+e^{-x})}$. These two procedures are illustrated in (2a)

$$w = \sigma(l(w^k) + l(w^u)) \qquad (2a)$$

Updating the gradients of the parameters is possible using the chain rule, as illustrated in (2b) for the student-specific component of the parameter.

$$\frac{\partial J}{\partial w^u} = \frac{\partial J}{\partial w}\frac{\partial w}{\partial w^u} \qquad (2b)$$

The IBKT models are built in an incremental manner by adding w^u in batches and where the effects of each addition are examined on Cross Validation (CV) performance. It is also possible to improve the overall accuracy by incrementally updating the w^k once a new group of students finishes a course or a course unit.

Figure 1 depicts the structure for the HMM model of both BKT and IBKT. Although the underlying HMM model and hence the process of a student practicing exercises remains the same, the fitting process is different. The parameters λ are spitted into two components and the model is fitted for each student separately by computing the gradients of these parameters.

The blue circular nodes capture the hidden students' knowledge state per skill, while the orange rectangles denote the exercise-performance observations associated to each skill correspondingly. We note that, in the upcoming figures the blue circular nodes and orange rectangles are also used to describe the same meaning. The nodes in the probabilistic models denote stochastic computations whereas in the RNN deterministic ones.

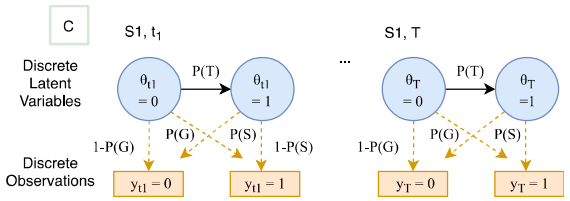

Figure 1. *Baseline and Individualized Bayesian Knowledge Tracing represented as a Hidden Markov Model.* In IBKT, the four parameters {G,S,θ_t,T} are splitted to include student-specific parametrs.

Both the BKT and IBKT assume independent skills sets and cannot deal with hierarchical structures since they are undirected graphs. This assumption is restrictive because it imposes that different skill sets cannot be related and, as a result, observing an outcome for one skill set is not informative for the knowledge level of another skill set. However, the expert model in educational domains, that is the decomposition of the subject matter or set of skills into a set of concepts (KCs) that need to be acquired by a learner, is frequently hierarchical. DAG is the optimal data representation for describing the expert model in traditional and adaptive learning systems that incorporate parallel scalable architectures and BDA [3].

C. *Dynamic Bayesian Network: Discrete Skill-Specific Dependencies in KT*

DBN is a DAG model allowing for the joint representation of dependencies among skills within the same model. [21] applied DBN in knowledge acquisition modeling in a KT setting.

Again, at each trial t, a student m receives a quiz-like assessment that contains learning opportunities, but this time these belong to different skills S. The Bayesian network is repeated itself at each time step t with additional edges connecting the knowledge state on a skill at t to $t+1$. The set of variables X contains all skill nodes S as well as all observation nodes Y of the model while H denote the domain of the unobserved variables, *i.e.*, learning opportunities that have not yet been attempted by students and hence their corresponding binary skill variables S are also latent. The objective is then again to estimate the parameters θ that maximize the likelihood of joint probability $p(y_m, h_m|\theta)$, where y_m and h_m denote the observed and hidden variables respectively.

The enhancement of the model is that even without having observed certain outcomes for a skill, say y_3 in time step t_2, is still possible to infer the knowledge state regarding $S3$. To illustrate that, consider the example model depicted in Figure 2. It depicts that, the probability of skill $S3$ being mastered at t_2 depends not only on the state of $S3$ at the previous time-step t_1, but also on the states of S1 and S2 at t_2. Suppose now that a student solves a learning opportunity associated with $S2$ at step t_2; then the hidden variables at t_2 will be $h_m = \{S1, S2, S3, y3, y1\}$ while the observed variables will be $y_m = y_2$.

In KT, the objective function is the log (likelihood) loss. In DBN, it is reformulated using a log-linear model to obtain a linear combination of a lower dimensional representation of features F. Equation (3) shows the log likelihood function of DBN:

$$L(w) = \sum_{m} \ln\left(\sum_{h_m} exp(w^{\mathrm{T}}\varphi(y_m, h_m) - \ln(Z))\right) \qquad (3)$$

, where φ: Y x H $\rightarrow \mathbb{R}^F$ denotes a mapping from the latent space H and the observed space Y to an F-dimensional feature vector. Z is a normalizing constant and w denotes the weights that can be directly linked to the parameters of the model θ.

DBNs grapple with the same limitations as HMM: the representation of student understanding is binary and reported per skill, given that each skill can be associated with exactly one observable, and there is the requirement for accurate concept labeling. RNN have only recently tried to model student understanding in order to break the beforementioned assumptions.

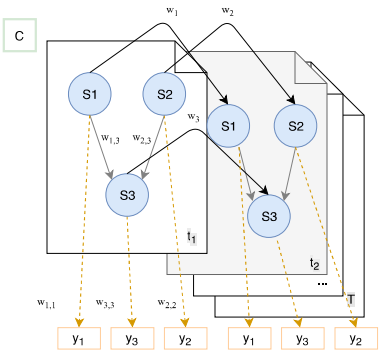

Figure 2. Bayesian Knowledge Tracing represented as a Dynamic Bayesian Network unrolled over T time steps. The hierarchical relationships between the skills are depicted and incorporated to the estimation of knowledge growth, shown by the arrow lines.

D. Recurrent Neural Networks: Continuous Knowledge States & Skill Dependencies

In an RNN, the hidden layer provides a continuous and high dimensional representation of the latent knowledge state h_t, which learns the properties of sequences of observations of student responses $x_t = (x_0, .., x_T)$, denoted as a_t in (4c), on learning activities, denoted as q_t in (4c). DKT [25] exploits the utility of LSTM whose fully and recurrent connection allow them to retain information of x_t for many time steps; and use it in a prediction at a much later point in time. The distributed representation allows an RNN to induce features related to skill dependencies or concepts (KCs) associated with exercises. The hidden-to-hidden connections encode the degree of overlapping between skills and exercises. According to (4a), the hidden units are activated via the hyperbolic tangent, which employs information on both the input x_t and on the previous activation h_{t-1} ,

$$h_t = \tanh(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \qquad (4a)$$

where b_h is the bias term and W_{hx}, W_{hh} are the weights of units corresponding to the input and hidden layer. The non-linear and deterministic output h_t will be passed to the sigmoid function σ to give the probability of getting each of the learning activities correct $y_t = (y_0, .., y_T)$ in the students' next interaction $t + 1$, as shown in (4b):

$$y_t = \sigma(W_{yh}h_t + b_y) \qquad (4b)$$

Finally, the loss for a single student will be the negative log-likelihood, as shown in (4c):

$$L = \sum_{t} l(y^T \delta(q_{t+1}), a_{t+1}) \qquad (4c)$$

where l is the binary cross entropy and δ denotes the one hot encoding transformation of the input, a necessary step for ANN and a suitable preprocessing step for small data-sets. Compressing Sensing is suitable for big data sets.

Figure 3 depicts an example architecture of RNN, where X represents the entire sequence of exercises a student receives in the order the student receives them. After feeding X to the network, each time the student answers an exercise, a prediction is made as to whether or not she/he would answer an exercise of each concept (KC) correctly on her next interaction. It's important to note that, in the DKT a deep RNN is employed, whose architecture include many hidden layers.

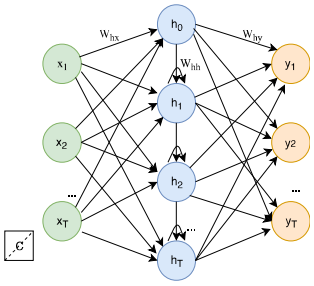

Figure 3. Deep Knowledge Tracing represented as a Recurrent Neural Network unrolled over T time steps. The arrow lines represent the change in the hidden knowledge state which is updated in non-linear

E. Comparison of the sequence models for KT

Table I outlines and compares important features of the models described above. The modeling of knowledge acquisition and the prediction of future performance in the BKT are binary and skill-specific without conditioning in individual learning or skill interdependencies. The predictions in the other models allow for individualization, skill dependencies, or continuous latent states and the discovery of the concept map. All models share the

 ISBN: 978-1-61208-681-1

assumption that each learning activity is a learning opportunity rather than an opportunity to assess the acquired knowledge.

The criterion of choosing the right algorithm is a combination of the efficiency of the available data along with the learning algorithm's components; these are the representation, evaluation, and optimization [15]. The rows of representation, optimization, evaluation, learnability, and efficiency in Table I comprise the technical-oriented elements whereas the remaining ones reflect the impact on educational settings. In the below paragraphs, we briefly describe each of
these components considering the KT task. The data representation has been already introduced in Section II.

1) Evaluation of the predictions

Model evaluation metrics analyze the performance of the predictive model and are widely discussed in the context of general machine learning applications including educational ones [26]–[28]. In KT, these include the Root Mean Square Error (RMSE), classification accuracy, and Area Under the Curve (AUC). RMSE is the standard performance metric, and it has demonstrated a high correlation to the log-likelihood function and the *'moment of knowledge acquisition'* [26]. AUC should be used only as an additional metric, in order to assess the model's ability to discriminate incorrect from correct performance, since it has several important disadvantages with regard to KT. DKT was criticized in terms of the employment of AUC, because it computes the accuracy on a per-trial basis instead of per-skill.

TABLE I. COMPARISON OF KNOWLEDGE TRACING MODELS

Model	**BKT**	**IBKT**	**DBN**	**DKT**
Extension	Baseline	Personalization	Detailed Skill Estimation	Continuous Knowledge State
Representation	HMM	HMM	DBN	RNN
Optimization	Curve Fitting, Expectation Maximization	Gradient Descent	Constrained Latent Structure	Stochastic Gradient Descent
Learnability-Fitting	158 observations-55 skills	2 datasets: *i*.8,918,054 observations, 3,310 students 515 or 541 skills *ii*. 20,012,498 observations 6,043 students 800-900 skills	5 datasets: Size range: 100-500 observations 3-9 Skills 77- 7265 students	3 datasets: Total size: 65.000 students, 2.000 observations-answers 230 items
Efficiency	4 /skill,	4 /skill + 2 /student *(a)*	4/skill + 2^{n-1} for n skills	250,000 with 200 hidden units & 50 skills. 4((input size+1) * output size + output size^2)
Evaluation	MAE	RMSE, Accuracy	RMSE, AUC	AUC
Restrictions	Prone to Bias	Independent Skills	Complex & hard-coded	Highly Complex & not interpretable
Variation in learner ability	✗	✓	✗	✓
Inclusion of forgetting	✗	✗	✓	✓
Inter-Skill Similarity	✗	✗	✓	✓
Exercise ordering effect	✗	✗	✗	✓

a. the initial probability and the learning rate is individualized

2) Optimization, Identifiability and Degeneracy

The optimization function derives the optimal possible values for the parameters of the objective function. Unlike in most other optimization problems, the function that generated the data and should be optimized is unknown and hence training error surrogates for test error [15]. The optimization of the log-likelihood function is performed using Curve Fitting (CF), Expectation Maximization (EM), Constrained optimization, and Gradient Descent methods (GD). All of them with appropriate initialization conditions [18][28]–[30] of the parameters, can solve for the identifiability issue which is considered an issue in the probabilistic approaches [20][28]. The identifiability issue is directly linked with the interpretability of the parameters values computed by the probabilistic models [20], where inferences about knowledge states are being made. It arises when there is more than one combination of parameters that optimizes the objective function. Constrained optimization [21] uses log-linear likelihood to ensure the interpretability of the constrained parameters, and it is suitable for DBNs. GD allows IBKT to introduce student-specific parameters to BKT without expanding the structure of the underlying HMM and hence without increasing the computational cost of fitting [23].

Equally important for the optimization methods is to be robust to degeneracy, where it is possible to obtain model parameters which lead to paradoxical behavior [30]. The standard KT model is susceptible to converging to erroneous degenerate states depending on the initial values of the parameters [28], and many research has focused on this property of the models [29]–[31]. An example in the BKT is the probability that the student acquired the instructed knowledge dropping after three correct answers in a row [29]. An instance in DKT includes the alternation between known and not-yet-known instead of transiting gradually over time [31].

3) Computational & Statistical Efficiency

Learnability comprises statistical efficiency, that is the number of student interaction examples required for good generalization, namely to correctly classify the unseen examples of interactions. The number of examples required to establish convergence, which is related to the number of training data that is used to learn the parameters of the model, is depicted in the table as Learnability-Fitting. The training of BKT model is faster, while deep neural networks and IBKT is relatively slow due to the requirement of large datasets for effective training. Generalization in DBN is inferred by using different learning domains datasets. According to the authors [21], the performance differences between DBN and BKT, especially the influence of the different parameters, need to be investigated further.

Computational efficiency refers to the number of computations during training and during prediction. These

include the number of iterations of the optimization algorithm and the number of resources (i.e., the number of hidden units). In the table, we note only the number of model parameters which is not necessarily the most appropriate measure of model complexity. Nonlinear functions and large datasets increase the model complexity which offers flexibility in fitting the data [20]. In DKT, there are high demands regarding computational resources. Nowadays, there are many parallel and distributed computing infrastructures that can be used to boost the efficiency of such data-intensive tasks. IBKT models take the advantage of parallel computing infrastructures. Comparing HMM and DBN, the latter needed 21-86 parameters for the datasets used in the paper. DBN is more computationally expensive due to their complex loopy structure and the skill-dependencies [21].

It is important to note that, the parameter estimates and the behavior of KT models should be researched in scalability cases in either the number of students or the increased number of observations per student [30].

4) Features related to performance and learning

The DKT model allows for differences in learning ability of the student by conditioning on recent performance of the student. By giving the complete sequence trial of performance-exercise pairs to the model, it can condition on the average accuracy of previous trials. DBN and DKT allow for skill-dependencies and can also infer the effect of exercise ordering on learning, which is considered an important element in learning and retention. The probabilistic KT tends to predict practice performance over brief intervals where forgetting the acquired knowledge, *the probability of transitioning from a state of knowing to not knowing a skill*, is almost irrelevant; whereas DKT incorporates recency effects and allows for long-term learning.

The complex representation in DKT is chosen based on the grounds that learning is a complex process [25] that shouldn't rely only on simple parametric models because they cannot capture enough of the complexity of interest unless provided with the appropriate feature space [22]. The assumption embodied in this approach is that the observed data is generated by the interactions of many different factors on multiple levels. DKT is a complex model, and thereby it should be applied to more complex problems and data. Hence, as long as there are sufficient data more behavioral in nature to constrain the model, a shift to connectionist paradigms of modeling will offer superior results when compared to classical approaches [11].

DKT success is attributed to its flexibility and generality in capturing statistical regularities directly present in the inputs and outputs, instead of representation learning [22]. When the performance of the baseline BKT and DKT models is compared [22], it is found that both models perform equally well, when variations of BKT models allow for more flexibility in modeling statistical regularities, that DKT has already the ability to explore. These are forgetting, variability in abilities among students, and skill discovery that allows for interactions between skills.

IV. ITEM RESPONSE THEORY FOR PREDICTING FUTURE PERFORMANCE

This review focuses specifically on Knowledge Tracing, thereby ignoring the only available alternative which is Item Response Theory (IRT) [32]. Theoretically, IRT models differ from KT models on that the former is developed for assessment purposes *(i.e., the theory focuses on short tests in which no learning occurs)* or for modeling very coarse-grained skills where the overall learning is slow (i.e., summative rather than formative assessments) [33]. Technically, IRT uses the responses on learning opportunities directly to estimate the learner's ability, while KT models go through the concept map or featurization.

The concept of IRT assumes that the probability of a correct response to an assessment item is a mathematical function of student parameters and item parameters. The latter is better estimated when there is a large amount of data to calibrate them. The student parameters can be used to account for variability in student a priori abilities. It's interesting to note that, (2a) of IBKT incorporates the compensatory logic behind the IRT, when summing the logistic functions to incorporate skill and student-specific parameters [23]. The prediction task in the baseline model is done by mapping a difference between a student knowledge on a skill θ and an item difficulty β into the probability of a correct answer $r_t = 1$ using a logistic function $\sigma(x)$, as depicted in (5). The estimate of ability is continually re-calibrated based on learner's performance.

$$p(r_t = 1|\theta) = \frac{1}{1+\exp\{-(\theta-\beta)\}} \quad (5)$$

The baseline IRT model is a logistic regression based Rasch model, also known as the One Parameter (1PL) IRT while its descendants include the Performance Factor Analysis (PFA) and the Additive and Conjugate Factor Model. The latter model is better estimated when there is a large amount of data available for calibration. The PFA model is highly predictive, but it's not useful for adaptive environments in the sense that it cannot optimize the subset of items presented to students according to their historical performance. The literature has already compared the models of PFA and BKT, both in theoretical [34] and in practical [35] terms *(i.e., predictive accuracy and parameter plausibility*). Both models are considered difficult to implement in an online environment and are rarely evaluated with respect to online prediction performance [33][36][37].

V. PROSPECTS AND CHALLENGES

Predicting future performance through modeling knowledge acquisition is a complex task; as human learning is grounded in the complexity of both the human brain and knowledge. This raises the opportunity to increase our understanding of knowledge prediction by synthesizing methods from various academic disciplines such as human-machine interaction design, machine learning, psychometrics, educational science, pedagogy, and neuroscience.

 ISBN: 978-1-61208-681-1

From a social science perspective, learning is influenced by complex macro-, meso- and micro-level interactions, including affect [38], motivation [39][40], and even social identity [41]. Predicting student knowledge with the mere observation of correct versus incorrect responses to learning activities provides weak evidence since it's not a sufficient data source.

Currently, there are some KT models augmented with non-performance data such as metacognitive [42], affect [43], and other student traits apart from learning rates [44] [45]. As educational apps and smart learning environments increase in popularity, it may be possible to collect valuable, diverse and vast amounts of student learning data, that will capture the reality of learning, and hence they will create opportunities, as well as new challenges, to deepen our understanding of knowledge acquisition and employ these insights to personalize education better.

VI. CONCLUSIONS

Modeling learner's skill acquisition and predicting future performance is an integral part of online adaptive learning systems that drive personalized instruction. Knowledge Tracing is a data mining framework widely used for that purpose because of its capability to infer a student's dynamic knowledge state as the learner interacts with a sequence of learning activities. Embarking from the baseline Bayesian model and based on the principles desired for adaptive learning systems, we outline three of the model's most recent extensions. These include the individualization of learning pace among students, the incorporation of the relationships among multiple skills, and the continuous representation of the knowledge state, which is able to induce both student and skill-specific features.

We show how probabilistic and deep learning approaches are related to the task of modeling sequences of student interactions by outlining their technical and educational requirements, advantages and restrictions. In particular, we investigate the assumptions in representation, the potential pitfalls in optimization, and the evaluation of the predictions. The general idea is that by investigating these aspects, one can gain an understanding why prediction models work the way they do or why they fail in other cases. A crucial question is how efficient and accurate these learning methods are regarding learning and generalization when they are applied to online adaptive learning environments where scalability and computational speed are important elements. The current study is useful both for researchers and developers allowing for a comparison of the different models. In addition, the corresponding citations throughout the paper can be used to provide further guidance in implementing or extending a model for a specific data source, online learning environment, or educational application.

REFERENCES

[1] C. Romero, J. R. Romero, and S. Ventura, "A Survey on Pre-Processing Educational Data," Springer Cham, pp. 29–64, 2014.

[2] C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art," IEEE Trans. Syst. Man, Cybern. Part C Applications Rev., vol. 40, no. 6, pp. 601–618, Nov. 2010.

[3] A. Essa, "A possible future for next generation adaptive learning systems," Smart Learn. Environ., vol. 3, no. 1, p. 16, Dec. 2016.

[4] K. W. Fischer, U. Goswami, and J. Geake, "The Future of Educational Neuroscience," Mind, Brain, Educ., vol. 4, no. 2, pp. 68–80, Jun. 2010.

[5] Z.-T. Zhu, M.-H. Yu, and P. Riezebos, "A research framework of smart education," Smart Learn. Environ., vol. 3, no. 1, p. 4, Dec. 2016.

[6] S. Kontogiannis et al., "Services and high level architecture of a smart interconnected classroom," in IEEE SEEDA-CECNSM, Sep. 2018, unpublished.

[7] Z. Papamitsiou and A. A. Economides, "Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence," Journal of Educational Technology & Society, vol. 17. International Forum of Educational Technology & Society, pp. 49–64, 2014.

[8] B. K. Daniel, "Big Data and data science: A critical review of issues for educational research," Review, Wiley, Br. J. Educ. Technol., Nov. 2017.

[9] K. Nadu and L. Muthu, "Application of Big Data in Education Data Mining and Learning Analytics -A Literature Review ICTACT J. Soft Comput., vol. 5, no. 4, pp. 1035–1049, Jul. 2015.

[10] C. Romero and S. Ventura, "Educational data science in massive open online courses," Wiley Interdiscip. Rev. Data Min. Knowl. Discov., vol. 7, no. 1, p. e1187, Jan. 2017.

[11] Z. A. Pardos, "Big data in education and the models that love them," Curr. Opin. Behav. Sci., vol. 18, pp. 107–113, Dec. 2017.

[12] I. Taleb, R. Dssouli, and M. A. Serhani, "Big Data Pre-processing: A Quality Framework," in 2015 IEEE International Congress on Big Data, 2015, pp. 191–198.

[13] P. Prinsloo, E. Archer, G. Barnes, Y. Chetty, and D. Van Zyl, "Bigger data as better data in open distance learning" Review, Int. Rev. Res. Open Distrib. Learn., vol. 16, no. 1, Feb. 2015.

[14] D. Gibson, "Big Data in Higher Education: Research Methods and Analytics Supporting the Learning Journey," Technol. Knowl. Learn., vol. 22, no. 3, pp. 237–241, Oct. 2017.

[15] P. Domingos and Pedro, "A few useful things to know about machine learning," Commun. ACM, vol. 55, no. 10, p. 78, Oct. 2012.

[16] K. Colchester, H. Hagras, D. Alghazzawi, and G. Aldabbagh, "A Survey of Artificial Intelligence Techniques Employed for Adaptive Educational Systems within E-Learning Platforms," J. Artif. Intell. Soft Comput. Res., vol. 7, no. 1, pp. 47–64, Jan. 2017.

[17] A. Peña-Ayala, "Educational data mining: A survey and a data mining-based analysis of recent works," Expert Syst. Appl., vol. 41, no. 4, pp. 1432–1462, Mar. 2014.

[18] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," User Model. User-Adapted Interact., vol. 4, no. 4, pp. 253–278, 1995.

[19] Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[20] C. M. Bishop, Pattern recognition and machine learning, Editors: M. Jordan J. Kleinberg B. Scholkopf, Springer, 2006.
[21] T. Kaser, S. Klingler, A. G. Schwing, and M. Gross, "Dynamic Bayesian Networks for Student Modeling," IEEE Trans. Learn. Technol., vol. 10, no. 4, pp. 450–462, Oct. 2017.
[22] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?," arXiv preprint arXiv:1604.02416, Mar. 2016.
[23] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized Bayesian Knowledge Tracing Models," in International Conference on Artificial Intelligence in Education, 2013, pp. 171–180.
[24] Z. A. Pardos and N. T. Heffernan, "Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing," Springer, Berlin, Heidelberg, 2010, pp. 255–266.
[25] C. Piech et al., "Deep Knowledge Tracing," in Advances in Neural Information Processing Systems, NIPS, 2015, pp. 505–513.
[26] R. Pelanek, "Metrics for Evaluation of Student Models.," J. Educ. Data Min., vol. 7, no. 2, pp. 1–19, 2015.
[27] J. P. González-Brenes and Y. Huang, "Your model is predictive-but is it useful? Theoretical and Empirical Considerations of a New Paradigm for Adaptive Tutoring Evaluation," in Proceedings of the 8th International Conference on Educational Data Mining, 2015.
[28] J. E. Beck and K. Chang, "Identifiability: A Fundamental Problem of Student Modeling," pp. 137–146, 2007, Proceedings of the 11th International Conference on User Modeling.
[29] R. S. J. d. Baker, A. T. Corbett, and V. Aleven, "More Accurate Student Modeling through Contextual Estimation of Slip and Guess Probabilities in Bayesian Knowledge Tracing," pp. 406–415, 2008, in International Conference on Intelligent Tutoring Systems.
[30] Z. A. Pardos, Z. A. Pardos, and N. T. Heffernan, "Navigating the parameter space of Bayesian Knowledge Tracing models: Visualizations of the convergence of the Expectation Maximization algorithm.", In Proceedings of the 3rd International Conference on Educational Data Mining pp. 161–170, 2010
[31] C.-K. Yeung and D.-Y. Yeung, "Addressing Two Problems in Deep Knowledge Tracing via Prediction-Consistent Regularization," Jun. 2018, in press, In Proceedings of the 5th ACM Conference on Learning @ Scale,
[32] M. Khajah, Y. Huang, J. P. Gonzales-Brenes, M. C. Mozer, and P. Brusilovsky, "Integrating knowledge tracing and item response theory: A tale of two frameworks", In Proceedings of the 4th International Workshop on Personalization Approaches in Learning Environments, pp. 7–15, 2014
[33] R. Pelánek, "Applications of the Elo rating system in adaptive educational systems," Comput. Educ., vol. 98, pp. 169–179, Jul. 2016.
[34] R. Pelánek, "Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques," User Model. User-adapt. Interact., vol. 27, no. 3–5, pp. 313–350, Dec. 2017.
[35] Y. Gong, J. E. Beck, and N. T. Heffernan, "Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures," Springer, Berlin, Heidelberg, 2010, pp. 35–44.
[36] C. Ekanadham and Y. Karklin, "T-SKIRT: Online Estimation of Student Proficiency in an Adaptive Learning System.", arXiv preprint arXiv:1702.04282, 2017
[37] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, "Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation Acknowledgements." arXiv preprint arXiv:1604.02336, 2016.
[38] E. A. Linnenbrink, P. R. Pintrich, and P. R. Pintrich, "Role of Affect in Cognitive Processing in Academic Contexts," pp. 71–102, Jul. 2004.
[39] A. J. Elliot and C. S. Dweck, Handbook of competence and motivation. Guilford Press, 2007.
[40] B. Fogg and BJ, "A behavior model for persuasive design," in Proceedings of the 4th International Conference on Persuasive Technology - Persuasive ', p. 1., Sep. 2009
[41] G. L. Cohen and J. Garcia, "Identity, Belonging, and Achievement: A Model, Interventions, Implications," Current Directions in Psychological Science, vol. 17. Sage Publications, Inc. Association for Psychological Science, pp. 365–369, 2008.
[42] I. Roll, R. S. Baker, V. Aleven, B. M. Mclaren, and K. R. Koedinger, "Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems 1 Metacognition in Intelligent Tutoring Systems." In: Proceedings of User Modeling, pp. 379–388, 2005
[43] S. Spaulding and C. Breazeal, "Affect and Inference in Bayesian Knowledge Tracing with a Robot Tutor." Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, pp. 219-220, USA 2015
[44] M. Khajah, R. M. Wing, R. V Lindsey, and M. C. Mozer, "Incorporating Latent Factors Into Knowledge Tracing To Predict Individual Differences In Learning." Proceedings of the 7th International Conference on Educational Data Mining, Educational Data Mining Society Press, pp. 99–106, 2014.
[45] J. I. E. Lee, "The Impact on Individualizing Student Models on Necessary Practice Opportunities.," Int. Educ. Data Min. Soc., In Proceedings of the 5th International Conference on Educational Data Mining, pp. 118–125, Jun. 2012

 ISBN: 978-1-61208-681-1