Interpretable Knowledge Tracing:
Simple and Efficient Student Modeling with Causal Relations

Sein Minn[1], Jill-Jênn Vie[2], Koh Takeuchi[3], Hisashi Kashima[3] and Feida Zhu[4]

[1] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France
[2] Universit Paris-Saclay, Inria, CEA, 91120 Palaiseau, France
[3] Kyoto University, Japan
[4] Singapore Management University, Singapore
{sein.minn, jill-jenn.vie}@inria.fr, {takeuchi, kashima}@i.kyoto-u.ac.jp, fdzhu@smu.edu.sg

Abstract

Intelligent Tutoring Systems have become critically important in future learning environments. Knowledge Tracing (KT) is a crucial part of that system. It is about inferring the skill mastery of students and predicting their performance to adjust the curriculum accordingly. Deep Learning-based KT models have shown significant predictive performance compared with traditional models. However, it is difficult to extract psychologically meaningful explanations from the tens of thousands of parameters in neural networks, that would relate to cognitive theory. There are several ways to achieve high accuracy in student performance prediction but diagnostic and prognostic reasoning are more critical in learning sciences. Since KT problem has few observable features (problem ID and student's correctness at each practice), we extract meaningful latent features from students' response data by using machine learning and data mining techniques. In this work, we present Interpretable Knowledge Tracing (IKT), a simple model that relies on three meaningful latent features: individual *skill mastery*, *ability profile* (learning transfer across skills) and *problem difficulty*. IKT's prediction of future student performance is made using a Tree-Augmented Naive Bayes Classifier (TAN), therefore its predictions are easier to explain than deep learning-based student models. IKT also shows better student performance prediction than deep learning-based student models without requiring a huge amount of parameters. We conduct ablation studies on each feature to examine their contribution to student performance prediction. Thus, IKT has great potential for providing adaptive and personalized instructions with causal reasoning in real-world educational systems.

Introduction

Learning with a computer plays an essential role in today's education. Learning can be personalized with adaptive learning instructions to improve individual learning gains and enhance students' learning experience. A personalized adaptive learning environment is much more efficient than traditional learning environments like classroom learning (Bloom 1984). Intelligent tutoring systems need to address the huge challenge of large-scale personalization for the process of real-world human learning. The most successful tutoring systems are utilized by tens or hundreds of thousands of students a year with growing numbers (Baker 2016). To fulfill the requirements of personalization, we need an efficient method to assess the mastery state of students' skills (knowledge) empirically. Knowledge Tracing (KT) is used to dynamically assess students' knowledge mastery state based on their past test outcomes. It can be used in the prediction of whether students may or may not answer the next problems correctly to adjust their personalized curriculum. AI techniques have found their way into the building of adaptive learning environments, in particular for the problem of knowledge tracing, for modeling students' conceptual or procedural knowledge from their observed performance on tasks (Corbett and Anderson 1994). The most current and well-known example is Deep Knowledge Tracing (DKT) (Piech et al. 2015) with the use of Recurrent Neural Networks (RNNs). Besides, it shows greater success in student performance prediction than early student models (Piech et al. 2015; Minn et al. 2019; Minn 2020).

Bayesian Knowledge Tracing (BKT) is the earliest and well-known sequential approach to KT with psychologically meaningful parameters. More specifically, BKT is a Hidden Markov Model (HMM), consisting of observed and latent variables that represent the student knowledge state on one specific skill (i.e., a skill can either be mastered by the student or not) and observed variables are assumed to be binary (the student can answer the associated problem correctly or not) (Corbett and Anderson 1994). Several extensions of BKT have been introduced by using contextualized guessing and slipping parameters (Baker, Corbett, and Aleven 2008), estimated transition with the use of help features (Baker and Yacef 2009), initial probability that the student knows the skill (Pardos and Heffernan 2010), item difficulty (Pardos and Heffernan 2011), clusters of different student groups (Pardos et al. 2012), student-specific parameters (Yudelson, Koedinger, and Gordon 2013). However, these extensions treat skills independently and are unable to detect learning transfer across skills.

Deep Knowledge Tracing (DKT) has obtained considerable attention thanks to its significant ability to model some of the learning transfer occurring across skills, which BKT cannot deal with. However, because DKT feeds all of past students' interactions (binary values with associated skills) to RNNs, it cannot provide a psychological interpretation (Piech et al. 2015) as the extensions of BKT do. Such

psychological information is concealed in the hidden layer of RNNs with tens of thousands of parameters (Khajah, Lindsey, and Mozer 2016). Besides, several deep learning-based knowledge tracing models have been proposed in recent years: Deep Knowledge Tracing and Dynamic Student Classification (Minn et al. 2018) enhances DKT with clusters of student ability profiles on skills at each time interval; Prerequisite-Driven Deep Knowledge Tracing (Chen et al. 2018) augments KT model by integrating prerequisite relations between skills; Exercise-Enhanced Recurrent Neural Network with Attention mechanism (Su et al. 2018) computes a weighted combination of all previous knowledge states; Sequential Key-Value Memory Networks (Abdelrahman and Wang 2019) is a hop-LSTM architecture that aggregates hidden knowledge states of similar problems into a new state; Deep hierarchical knowledge tracing (Wang, Ma, and Gao 2019) captures the relations between questions and skills to get problem representations; Graph-based Interaction Knowledge Tracing (Yang et al. 2021) utilizes graph convolutional networks to substantially incorporate question-skill correlations. Deep learning-based models show better predictive performance than the preceding approaches, in part because they are known to preserve past information in sequential data, such as student outcome traces. Nevertheless, those models are less likely to provide psychologically meaningful explanations for their inference than predicting the correctness of a given problem.

In this paper[1], we are trying to provide meaningful explanations through feature engineering and a simple probabilistic graphical model. Hence, we propose a novel student model called Interpretable Knowledge Tracing (IKT), by utilizing three meaningful features: individual *skill mastery*, *ability profile* of students (learning transfer across skills), and *problem difficulty*. We first utilize conventional machine learning techniques, such as hidden Markov models and k-means clustering, to extract meaningful features, and then incorporate the extracted features using a Tree Augmented Naive Bayes classifier for inferring the correctness of a future problem. In contrast to the family of DKT, our IKT is a novel model to provide inference interpretation through a probabilistic graphical model with meaningful features while keeping high achieving prediction performance in the task of student performance prediction. We experimentally show that IKT outperforms well-known student models for performance prediction on several well-known knowledge tracing datasets. Additionally, we also conduct ablation studies to measure the contribution of each feature, by learning different tree structures.

Background

In successful learning environments like Cognitive Tutor and ASSISTments, KT plays as a mechanism for tracing learners' knowledge (Desmarais and Baker 2012).

KT can be seen as a supervised sequential learning problem. The KT model is given student past interactions with the system that include: skills $S = (s_1, s_2, .., s_t) \in \{1, \ldots, M\}^t$ along with responses $R = (r_1, r_2, .., r_t) \in$

---

[1]This work is available at https://github.com/simon-tan/IKT

$\{0, 1\}^t$ and predicts the probability of getting a correct answer for the next problem. It mainly depends on the mastery of the skill s associated with problems $P = (p_1, p_2, .., p_t)$. So we can define the probability of getting a correct answer as $p(r_t = 1|s_t, X)$ where $X = (x_1, x_2, .., x_{t-1})$ and $x_k = (s_k, r_k)$ is a tuple containing response r_k to skill s_k at time k. Then, we review here the best known KT modeling methods for estimating student's performance.

- Item Response Theory (IRT). In standardized tests, students' proficiency is assessed by one static latent variable (Hambleton, Swaminathan, and Rogers 1991). IRT has a strong theoretical background both in terms of being grounded in psychometric measurement and relying on a sound mathematical framework. Wilson et al. (2016) proposed a Bayesian extension of IRT (BIRT) that shows competitive performance to DKT in terms of student performance prediction.

- Bayesian Knowledge Tracing (BKT) is the earliest sequential approach to model a learner's changing knowledge state and is arguably the first model to relax the assumption on static knowledge states (Corbett and Anderson 1994).

- Performance Factors Analysis (PFA) was adapted from Learning Factor Analysis (LFA) (Cen, Koedinger, and Junker 2006) with sensitivity to the indicator of student learning performance. It allows conjunction by summing the contributions from all skills needed in a performance by relaxing the static knowledge assumption and modeling multiple skills simultaneously (Pavlik, Cen, and Koedinger 2009).

- Deep Knowledge Tracing (DKT) (Piech et al. 2015) uses a Long Short-Term Memory (LSTM) to represent the latent knowledge space of students along with the number of practices dynamically. This model compactly encodes the historical information from previous time steps by using the input, forget and output gates of a LSTM.

- Dynamic Key-Value Memory Networks (DKVMN) was proposed as an alternative to DKT that is inspired by the memory network architecture (Zhang et al. 2017). It utilizes an external memory neural network module and uses two memory slots called key memory and value memory to encode the knowledge state of students. Assessments of knowledge state on particular skills are stored in memory slots and controlled by read and write operations through additional attention mechanisms.

- Deep Knowledge Tracing and Dynamic Student Classification (DKT-DSC) was proposed to enhance DKT by utilizing dynamically evaluated student ability profiles at each time interval (Minn et al. 2018). It applies k-means clustering to detect student ability profiles and takes them into account in student performance prediction.

- Context-Aware Attentive Knowledge Tracing (AKT-R) was proposed by Ghosh, Heffernan, and Lan (2020) and uses a monotonic version of the scaled dot-product attention mechanism for the encoding and retrieving of the knowledge state. It also applies Rasch model for learning the skill and problem embeddings.

Interpretable Knowledge Tracing

When a student learns with an intelligent tutoring system (ITS), they practice a specific skill through answering several questions, and the ITS checks their mastery of skill according to whether they were able to provide correct answers. However, even with a high level of mastery of the skill associated with some problems, the student may provide an incorrect answer to those problems. We tend to regard such situations as misunderstanding of the problem, or failure to utilize the related skill properly in a particular problem under a new circumstance. So we can assume that other factors, such as ability profile (learning transfer across skills) or difficulty of the occurring problem at the current timestamp, have a direct effect on the situation.

To adopt the aforementioned assumptions into ITS, we propose a feature extraction procedure consisting of three data mining techniques. Then, instead of feeding all data to neural networks in deep learning-based KT models, we propose a student model called Interpretable Knowledge Tracing (IKT) which can predict a student's future responses by relying on three meaningful latent features: individual *skill mastery* of a student, *ability profile* (across skills) and *problem difficulty*.

Interpretation through Feature Engineering

Our procedure extracts three meaningful latent features from students' response data: *skill mastery*, *ability profile*, *problem difficulty*. These features tell us about how much a student knows about the practicing skill, what kind of ability that student possesses, and how difficult the occurring problem is at each timestamp. Instead of taking students' past interaction sequence (binary values) and learning all information in a hidden state of deep learning-based KT model, the correctness of the problem is inferred by using these latent features as evidence at each timestamp.

Knowledge Tracing The formulation of skill mastery is inspired by the assessment of skill mastery (probability of learning a skill s_t) in Bayesian Knowledge Tracing (BKT), which is a well-known knowledge tracing model with psychologically meaningful parameters based on a Hidden Markov Model. BKT infers mastery states, from "not learned" to "learned" and the probabilities above depend both on fixed parameters and the state at timestamp t.

For a certain skill $s \in S$, BKT consists of four parameters representing probabilities:

- $P(L_0)$: the probability that a student masters the skill before attempting the first problem associated with s;
- $P(T)$: the probability that a student, who does not currently master the skill, will master the skill after the next practice opportunity;
- $P(G)$: the probability that a student guesses the correct answer to a question despite not knowing the skill (*guess*);
- and $P(S)$: the probability that a student answers a question incorrectly despite knowing the skill (*slip*).

We apply a brute-force search algorithm to fit BKT. We have $P(L_0), P(T), P(G), P(S)$ of each skill after fitting
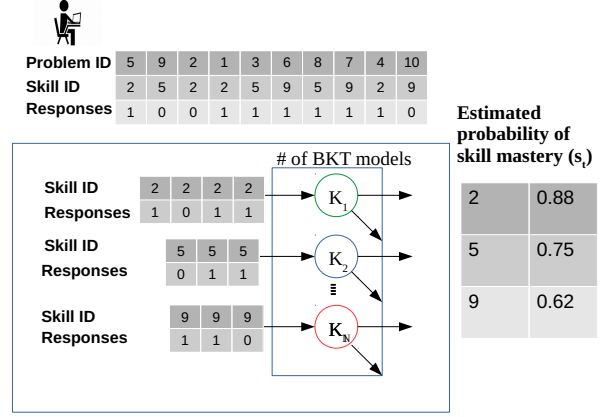

Figure 1: Skill mastery assessment for skills at each timestamp. Modeling of each skill is done independently and they do not interact with each other.

BKT. BKT is based on skill-specific modeling and can provide skill mastery of each skill according to the observed outcome *obs* that is correct ($obs = 1$) or incorrect ($obs = 0$):

$$P(L_t|1) = \frac{P(L_t)(1 - P(S))}{P(L_t)(1 - P(S)) + (1 - P(L_t))P(G)} \quad (1)$$

$$P(L_t|0) = \frac{P(L_t)P(S)}{P(L_t)P(S) + (1 - P(L_t))(1 - P(G))} \quad (2)$$

$$P(L_{t+1}) = P(L_t|obs) + (1 - P(L_t|obs))P(T) \quad (3)$$

By combining these equations, we can define the skill mastery as:

$$\textbf{skill mastery}(s_t) = \delta(P(L_t), s_t) \quad (4)$$

where $\delta(P(L_t), s_t)$ is a function that maps the skill mastery of particular skill s_t at current timestamp in the whole student interaction.

Note that *skill mastery* is the probability of learning skill s_t rather than the probability that a student applies the skill correctly in BKT. A BKT model is trained for each skill, and the inputs to each skill model are the binary responses of a student on that single skill. Other interleaved skills during the whole practice are ignored. Each skill model is independent, so there is no consideration of learning transfer across skills in this component.

Ability Profiling A strong limitation of BKT is that it treats each skill as an independent skill model, thus it can not discover or leverage inter-skill similarity. When a student is assessed on a new skill, BKT considers the learning that they have gained from previous skills as irrelevant to the new skill. Intuitively, learning transfer across skills is a naturally occurring phenomenon. General domain knowledge accumulates through practice and allows the student to

solve new problems in the same domain with greater success, even if that problem involves new and specific skills. However, BKT and most KT models lack a mechanism to deal with the *learning transfer* mechanism, and thus they cannot trace students' knowledge under transfer.

Learning transfer implies that students can transfer their acquired skills to new situations and across problems involving a different skill set. ITS models such as Learning Factor Analysis (LFA) (Cen, Koedinger, and Junker 2006) and Performance Factor Analysis (PFA) (Pavlik, Cen, and Koedinger 2009) aim to capture this learning transfer phenomenon. They introduced a factor that represents the learning accumulated on all skills through practice and then utilized this factor as a predictor of success in further practice. These models have outperformed the standard BKT model without the skill transfer mechanism, and have shed new light on the importance to consider skill transfer.

When BKT performs the prediction task, it estimates the skill mastery of a student without considering learning transfer across skills. This independent assumption makes BKT unable to evaluate any types of learning transfer that students should achieve at the current time interval in the long-term learning process. In DKT, an LSTM encodes the temporal information of student knowledge state with transfer learning in a single state vector and performs state-to-state transition globally. It is unable to assess the mastery of skills and the ability profile of the student to transfer learning.

To detect the regular changes of learning transfer across skills in the long-term learning process, we are inspired by the work of DKT-DSC (Minn et al. 2018). We reformulate the ability profile of a student and simplify it without sacrificing its originality and performance. It divides the student's interactions into multiple time intervals, then encodes student past performance for estimating their ability profile at the current time interval. The ability profile is encoded as a cluster ID and computed from the performance vector (with Equation 6) of length equal to the number of skills, and updated after each time interval by using all previous attempts on each skill. The success rates on each skill from past attempts data are transformed into a performance vector for clustering student i at time interval $1:z$ as follows (for brevity we omit indexing all terms by i in Equation 5):

$$R(x_j)_{1:z} = \sum_{t=1}^{z} \frac{(x_{jt})}{|N_{jt}|}, \tag{5}$$

$$d_{1:z}^i = (R(x_1)_{1:z}, R(x_2)_{1:z}, ..., R(x_n)_{1:z}), \tag{6}$$

where

- x_{jt} is the outcome of attempt of skill x_j being correctly answered at time interval t; 1 denotes successful attempts and 0 denotes unsuccessful attempts;
- $|N_{jt}|$ is the total number of practice attempts of skill x_j up to time interval z;
- n is the total number of skills;
- $R(x_j)_{1:z}$ represents the ratios of skill x_j being correctly answered from time interval 1 to current time interval z by student i. This is computed for all skills $(x_1, x_2, .., x_n)$;
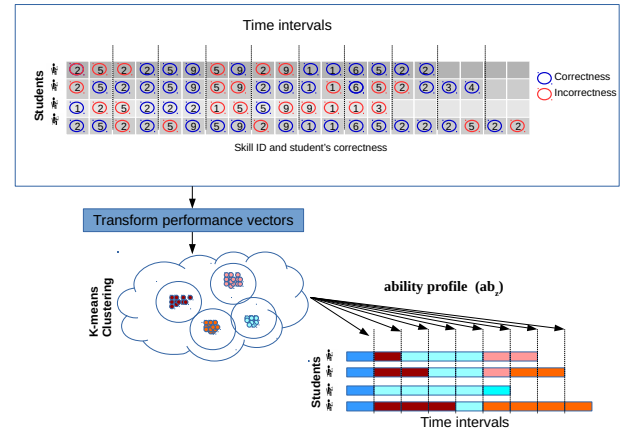

Figure 2: Detection of student's ability profile at each time interval.

- $d_{1:z}^i$ represents a *performance vector* of student i on all skills from time interval 1 until z.

Each student has a different number of total time intervals in the lifetime of their interactions with the system.

If a student's time interval has no attempt in the interval $0:z$, we assign 0.5 to ratio $R(x_j)_{1:z}$.

Therefore, data contains the encoded vector of the student's past performance and is accumulated and updated after each time interval. Time interval z and student i are ignored in the training process and only used in the clustering process later. Then the k-means algorithm is used to evaluate the temporal long-term learning ability of students in both training and testing at each time interval z, by measuring the Euclidean distance with centroids achieved after the training process as in DKT-DSC (Minn et al. 2018).

After learning the centroids of all clusters, each student i at each time interval z is assigned to the nearest cluster C_c by the following equation:

$$\textbf{ability profile}(ab_z) = \arg\min_{C} \sum_{c=1}^{K} \sum_{d_{1:z-1}^i \in C_c} ||d_{1:z-1}^i - \mu_c||^2 \tag{7}$$

where centroid μ_c is the mean of points for cluster C_c , and performance vector $d_{1:z-1}^i$ is the average performance data of student i from time interval 1 to $z - 1$.

Students are assigned to the nearest cluster and the label of this cluster ab_z represents the temporal student learning ability at time interval z. Evaluation is started after the first 20 attempts and then every 20 attempts made by a student. For the first time interval, all students are assigned with initial ability profile 1.

By adding this cluster ID ab_z (ability profile) of what group the student belongs to, we ensure that these high-level skill profiles are available to the model for making its predictions throughout the long-term interaction with the tutor.

Estimating Problem Difficulty The problem difficulty serves as a distinct feature for predicting predict student performance in previous studies (Minn, Zhu, and Desmarais

2018; Minn et al. 2019). Note that, in this study, we assume each problem is associated with a single skill, but the difficulty is associated with problems, not with the skills themselves. The difficulty of a problem p_j is determined on a scale of 1 to 10. **Problem difficulty**(P_j) is calculated as:

$$\textbf{difficulty level}(p_j) = \begin{cases} \delta(p_j) & \text{if } |N_j| \geq 4 \\ 5 & \text{otherwise} \end{cases} \quad (8)$$

where:

$$\delta(p_j) = \left\lfloor \frac{\sum_i^{|N_j|} O_i(p_j)}{|N_j|} \cdot 10 \right\rfloor \quad (9)$$

and where

- p_j represents the jth problem
- N_j is the set of students who attempted problem p_j
- $O_i(p_j)$ the outcome of the first attempt from student i to problem p_j, 1 if successful, 0 otherwise.

$\delta(p_j)$ is a function that maps the *average success rate* of problem p_j onto 10 levels. Unseen problems, those that do not have any record, and problems with fewer than 4 students ($|N_j| < 4$) in the dataset will have a difficulty of 5.

Interpretable Student Performance Prediction

In order to get an interpretation with diagnostic and prognostic reasoning, we decide to choose the Bayes net paradigm for future development. So, our approach utilizes a Tree-Augmented Naive Bayes Method (Friedman, Geiger, and Goldszmidt 1997). The TAN structure is a simple extension of the Naive Bayes network. Like Naive Bayes, the root node is the class node (correctness of the problem), causally connected to evidence nodes (skill ID, skill mastery, ability profile, and problem difficulty). Additionally, the TAN structure relaxes the assumption of independence between the evidence nodes (Minn, Fu, and Desmarais 2014). It allows most evidence nodes to have another parent, which can be a related evidence node. This model inherits the directed acyclic graph structure (Minn, Fu, and Lv 2016; Minn and Fu 2016) and produces a tree that captures relationships among the evidence nodes. The learning of this structure is not as computationally expensive as a general Bayesian network and much more cost effective than building a neural network for knowledge tracing in DKT. An example TAN structure is illustrated in Figure 3. The class node is the student's correctness hypothesis under consideration. The other nodes represent supporting evidence for the particular student's performance hypotheses at time t. Dependencies among the evidence nodes are captured as additional causal links in the TAN structure. Even though the direction of arrows represents the causal links between two nodes, information can flow in any direction based on the reasoning process (Pearl 2001).

The structure of TAN can be learned in different ways:

- a greedy search with the constraint that a node having more than one parent from the evidence nodes is not allowed (Cohen et al. 2004);
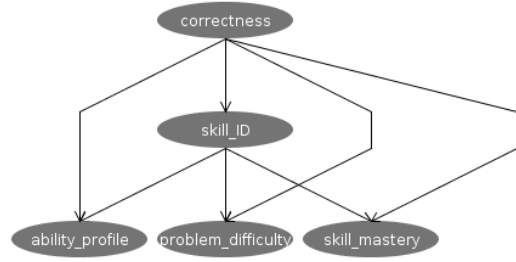

Figure 3: Tree Structure of TAN.

- a Minimum Weighted Spanning Tree (MWST) approach that builds a minimum spanning tree to capture the dependencies among evidence nodes, and then connects the class node to all of the evidence nodes (Friedman, Geiger, and Goldszmidt 1997).

The MWST algorithm is applied in this research by using the data mining toolkit Weka (Hall et al. 2009). Learning the structure of TAN is done by only using training data.

Our model predicts whether a student will be able to answer the next problem p_t based on their current knowledge state **skill mastery**(s_t), learning transfer across skills **ability profile**(ab_z) and difficulty level of problem occurring **problem difficulty**(P_j). Instead of only feeding all student previous interactions $X = (x_1, x_2, .., x_t)$ to a neural network, we propose a novel model called IKT.

IKT performs inference by using three meaningful extracted features f_t: skill mastery, ability profile, and problem difficulty as evidence at the current timestamp t.

$$P(correctness_t = y | f_t) = \frac{P(y)P(f_t | y)}{\sum_{y'} P(y')P(f_t | y')} \quad (10)$$

where $P(f_t | y) = P(s_t | y)P(\textbf{ability profile}(ab_z) | y, s_t)$
$$P(\textbf{problem difficulty}(P_j) | y, s_t)$$
$$P(\textbf{skill mastery}(s_t) | y, s_t)$$

Those are assessed **skill mastery**(s_t) of student i on skill s at time t, the temporal **ability profile**(ab_z) of student i at current time interval z, and **problem difficulty**(P_j) of problem P_j at time t (for brevity we omit indexing all terms by student i, s_t, ab_z P_j in Figure 3 and Equation 10). The inference is estimated in the context of discretized values, i.e. conditional probability tables (Hall et al. 2009). It doesn't handle continuous variables. Discretization algorithm bin all features into sets for best discrimination among classes (Mack et al. 2011).

The class node (correctness) represents the predicted probability that the student would answer the problem with the associated skill correctly. Thus the prediction of problem associated with skill s_t can be retrieved from $correctness_t$ as described in Figure 3. We can achieve interpretation via the conditional probability tables of each node with their causal links. We can trace back the cause of failure in students' problem-solving by detecting whether there is a deficiency in practicing skills or the problem is too difficult for individual students with their evidence at each timestamp.

Models	Ability profile	Problem information	Sequential
IRT	No	Yes	No
PFA	No	No	No
BKT	No	No	Yes
DKT	No	No	Yes
DKT-DSC	Yes	No	Yes
DKVMN	No	No	Yes
AKT-R	No	Yes	Yes
IKT	Yes	Yes	Yes

Table 1: Comparison on characteristics of student models.

Experiments

We compare the next problem student performance prediction of our model with well-known KT models mentioned above: BIRT (Wilson et al. 2016), BKT (Corbett and Anderson 1994), PFA (Pavlik, Cen, and Koedinger 2009), DKT (Piech et al. 2015), DKT-DSC (Minn et al. 2018) and DKVMN (Zhang et al. 2017). But we do not compare with other variants, because they are more or less similar and do not show significant performance differences. Table **??** summarizes the characteristics of compared student models, in which each model has its significant characteristics.

We implement all NN models with Tensorflow and all LSTM-based models share the same structure of fully-connected hidden nodes with an embedding size of 200. For speeding up the training process, mini-batch stochastic gradient descent is used to minimize the loss function. The batch size for our implementation is 32. We train the model with a learning rate of 0.01 and dropout is also applied for avoiding overfitting. We set the number of epochs to 100. All the models are trained and tested on the same sets of training and testing students. For BKT, we learn models for each skill and make predictions separately and the results for each skill are averaged.

In this experiment, we assume 20 attempts made by a student as a time interval for that student for detecting the student's ability profile. The total number of temporal values for students' ability profiles used in our experiment is 8 (7 clusters and 1 for initial ability profile before evaluation in initial time interval for all students). Five-fold cross-validation is used to make predictions on all datasets. Each fold involves splitting into 80% training students and 20% test students of each dataset. We conduct our analysis through Area Under Curve (AUC) as the majority class (correct) ranges 65-75 % across all datasets, and Root Mean Squared Error (RMSE) as it is a variant of Brier score commonly encountered in the knowledge tracing community (Gervet et al. 2020; Bergner, Halpin, and Vie 2021). For the input of DKVMN, values in both key and value memory are learned in the training process. For other models, one-hot encoded method is applied. The learned embeddings in value memory represent pre-trained weights which may contain the difficulty of each skill. For IKT, we learn the tree structure of TAN by using training data and making inference in testing data with WEKA (Hall et al. 2009).

Dataset	Number of			
	Skills	Problems	Students	Records
Algebra	437	15663	574	808,775
ASS-09	123	13002	4,163	278,607
ASS-12	198	41918	28,834	2,506,769

Table 2: Overview of datasets.

	Datasets			
Models	ASS-09	ASS-12	Algebra	Average
BIRT	0.750	0.744	0.812	0.768
PFA	0.701	0.672	0.754	0.709
BKT	0.651	0.623	0.642	0.638
DKT	0.721	0.713	0.784	0.739
DKT-DSC	0.735	0.721	0.792	0.749
DKVMN	0.710	0.707	0.780	0.732
AKT-R	<u>0.767</u>	**0.777**	<u>0.845</u>	0.796
IKT-3	**0.797**	<u>0.767</u>	**0.851**	**0.805**

Best scores are in bold, second best scores are underlined.

Table 3: AUC result for all tested datasets.

Datasets

In order to validate the proposed model, we tested it on three public datasets from two distinct tutoring scenarios in which students interact with a computer-based learning system in educational settings: 1) ASSISTments[2] is an online tutoring system that was first created in 2004 which engages middle and high-school students with scaffolded hints in their math problem. If students working on ASSISTments answer a problem correctly, they are given a new problem. If they answer it incorrectly, they are provided with a small tutoring session where they must answer a few questions that break the problem down into steps. Datasets are as follows: ASSISTments 2009-2010 (skill builder), ASSISTments 2012-2013 (Feng, Heffernan, and Koedinger 2009). 2) Cognitive Tutor. Algebra 2005-2006[3] is a development dataset released in KDD Cup 2010 competition from Carnegie Learning of PSLC DataShop.

For all datasets, only the first correct attempts to original problems are considered in our experiment. We remove data with missing values for skills and problems with duplicate records. To the best of our knowledge, these are among the most well known publicly available knowledge tracing datasets.

Results

The TAN structure results in a student model with better explanation with causal relations and higher predictive performance. The results in Tables 3 and 4 demonstrate that IKT outperforms significantly the well-known KT models in all tested datasets. IKT-3 has superior performance than any other models tested in our experiments. When we compare IKT-3 with our second best performer AKT-R (scores are underlined in table 3), the improvement is ranging from

---

[2]https://sites.google.com/site/assistmentsdata/

[3]https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

Models	Datasets			Average
	ASS-09	ASS-12	Algebra	
BIRT	0.440	0.441	0.374	0.418
PFA	0.454	0.440	0.391	0.428
BKT	0.471	0.510	0.440	0.473
DKT	0.450	0.430	0.380	0.420
DKT-DSC	0.434	0.427	0.373	0.411
DKVMN	0.451	0.430	0.380	0.42
AKT-R	<u>0.423</u>	**0.409**	**0.354**	<u>0.395</u>
IKT-3	**0.411**	<u>0.413</u>	**0.354**	**0.392**

Best scores are in bold, second best scores are underlined.

Table 4: RMSE result for all tested datasets.

Models	AUC		
	ASS-09	ASS-12	Algebra
IKT-1	0.705	0.690	0.731
IKT-2	0.715	0.696	0.734
IKT-3	**0.797**	**0.767**	**0.846**
Models	RMSE		
	ASS-09	ASS-12	Algebra
IKT-1	0.443	0.437	0.395
IKT-2	0.441	0.435	0.394
IKT-3	**0.411**	**0.413**	**0.354**

Table 5: AUC and RMSE result for ablation study.

0.71% to 3.91% in Algebra and ASS-09 datasets in terms of AUC. When we compare in terms of RMSE, it shows an improvement up to 2.84% over the second best performer AKT-R on the ASS-09 dataset. So IKT shows better performance than any other method in both AUC and RMSE (except AKT-R on the ASS-12 dataset).

Ablation Studies

The results so far suggest there may be a different impact of each factor on the predictive performance of KT models, and in particular the impact of item difficulty. This question is further analyzed in this section.

We compare our IKT model through an ablation study with the following different features:

- IKT-1: skill ID, skill mastery.
- IKT-2: the features of IKT-1 + ability profile.
- IKT-3: the features of IKT-2 + problem difficulty.

This study helps us understand the contribution of each feature in student performance prediction. IKT-1 takes only skill ID and skill mastery into account for student performance prediction which achieves higher performance than original BKT and has a bit lower performance than DKT (where DKT takes only binary values of student previous interaction). When IKT-2 takes skill ID, skill mastery and ability profile of a student, it shows similar (or a bit lower) performance as DKT and higher performance than BKT, PFA on two Assistments datasets. Our proposed model IKT-3 shows better performance than any KT methods compared in this experiment.

Table 5 reports the results. Ability profile results in a mild improvement in AUC smaller than 1.4% between IKT-1 to IKT-2. Results from each model with different combination of features show us which features provide more information in student performance prediction among various datasets. We can also see that the problem difficulty factor is the most influential factor to student performance prediction. When we apply problem difficulty to IKT-3, it increases around 11.2% to 15.7% in AUC (e.g. 0.731 AUC in IKT-1 to 0.846 IKT-3 in Algebra) respectively. It also explains why both BIRT and AKT-R have a better performance in student performance prediction among other models (see Tables 3, 4 and 5).

Conclusion

We extract three meaningful latent features from students' behavioral data by using data mining techniques in feature engineering. The TAN structure with causal relations based on these features results in a KT model with better performance for student performance prediction, that does not require a huge number of parameters nor a complex structure. It saves huge computational resources compared to deep learning models and provides a causal explanation for better understanding with meaningful features.

We proposed a causal probabilistic student model called IKT with three extracted meaningful features: student's skill mastery (probability of learning a skill), ability profile (learning transfer of a student), and problem difficulty to predict student performance. Unlike deep learning-based KT models, which only take students past interaction and learn all information in a hidden state with a huge amount of parameters and complex relations in structure, IKT can predict student performance by utilizing extracted features from skills, problems and students and gives us meaningful causal explanations. Experiments with three public datasets show that the proposed model outperforms well-known KT models (including deep learning-based models) and requires less computational power than deep learning-based models. Feature engineering helps us capture more variance from the data and Tree-Augmented Naive Bayes plays a critical role in providing causal explanations. It leads to more accurate and personalized performance predictions.

Although we focused only on prediction performance, we are also interested in studying the causal effect of the meaningful features, aiming to serve adaptive instructions in personalized learning systems. Meanwhile, a trade-off between student prediction performance and causal explanation should be found. More experiments will be designed and conducted to understand more about knowledge acquisition and learning behaviors to optimize human learning in future educational environments.

References

Abdelrahman, G.; and Wang, Q. 2019. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 175–184.

Baker, R. S. 2016. Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2): 600–614.

Baker, R. S.; Corbett, A. T.; and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on intelligent tutoring systems*, 406–415. Springer.

Baker, R. S.; and 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.

Bergner, Y.; Halpin, P.; and Vie, J.-J. 2021. Multidimensional Item Response Theory in the Style of Collaborative Filtering. *Psychometrika*, 1–23.

Bloom, B. S. 1984. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6): 4–16.

Cen, H.; Koedinger, K.; and Junker, B. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*, 164–175. Springer.

Chen, P.; Lu, Y.; Zheng, V. W.; and Pian, Y. 2018. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, 39–48. IEEE.

Cohen, I.; Chase, J. S.; Goldszmidt, M.; Kelly, T.; and Symons, J. 2004. Correlating Instrumentation Data to System States: A Building Block for Automated Diagnosis and Control. In *OSDI*, volume 4, 16–16.

Corbett, A. T.; and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4): 253–278.

Desmarais, M. C.; and Baker, R. S. 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.

Feng, M.; Heffernan, N.; and 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.

Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian network classifiers. *Machine learning*, 29(2): 131–163.

Gervet, T.; Koedinger, K.; Schneider, J.; Mitchell, T.; et al. 2020. When is Deep Learning the Best Approach to Knowledge Tracing? *JEDM— Journal of Educational Data Mining*, 12(3): 31–54.

Ghosh, A.; Heffernan, N.; and Lan, A. S. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2330–2339.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1): 10–18.

Hambleton, R. K.; Swaminathan, H.; and Rogers, H. J. 1991. *Fundamentals of item response theory*. Sage.

Khajah, M.; Lindsey, R. V.; and Mozer, M. C. 2016. How Deep is Knowledge Tracing?. In *Proceedings of the 9th International Conference on Educational Data Mining*. IEDMS.

Mack, D. L.; Biswas, G.; Koutsoukos, X. D.; and Mylaraswamy, D. 2011. Using Tree Augmented Naive Bayes Classifiers to Improve Engine Fault Models. In *Uncertainty in Artificial Intelligence: Bayesian Modeling Applications Workshop*. Citeseer.

Minn, S. 2020. BKT-LSTM: Efficient Student Modeling for knowledge tracing and student performance prediction. *arXiv preprint arXiv:2012.12218*.

Minn, S.; Desmarais, M. C.; Zhu, F.; Xiao, J.; and Wang, J. 2019. Dynamic Student Classification on Memory Networks for Knowledge Tracing. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 163–174. Springer.

Minn, S.; Fu, S.; and Desmarais, M. C. 2014. Efficient learning of general Bayesian network Classifier by Local and Adaptive Search. In *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, 385–391. IEEE.

Minn, S.; Fu, S.; and Lv, T. 2016. Algorithm for exact recovery of Bayesian network for classification. *Application Research of Computers*, 33(05): 1327–1334.

Minn, S.; and Fu, S.-k. 2016. Accelerating Structure Learning of Bayesian Network. *Computer Science*.

Minn, S.; Yu, Y.; Desmarais, M. C.; Zhu, F.; and Vie, J.-J. 2018. Deep knowledge tracing and dynamic student classification for knowledge tracing. In *2018 IEEE International conference on data mining (ICDM)*, 1182–1187. IEEE.

Minn, S.; Zhu, F.; and Desmarais, M. C. 2018. Improving knowledge tracing model by integrating problem difficulty. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 1505–1506. IEEE.

Pardos, Z. A.; and Heffernan, N. T. 2010. Modeling individualization in a Bayesian networks implementation of knowledge tracing. In *International Conference on User Modeling, Adaptation, and Personalization*, 255–266. Springer.

Pardos, Z. A.; and Heffernan, N. T. 2011. KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *International conference on user modeling, adaptation, and personalization*, 243–254. Springer.

Pardos, Z. A.; Trivedi, S.; Heffernan, N. T.; and Sárközy, G. N. 2012. Clustered knowledge tracing. In *International Conference on Intelligent Tutoring Systems*, 405–410. Springer.

Pavlik, P. I.; Cen, H.; and Koedinger, K. R. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. In *14th International Conference on Artificial Intelligence in Education*.

Pearl, J. 2001. Bayesian networks, causal inference and knowledge discovery. *UCLA Cognitive Systems Laboratory, Technical Report*.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*, 505–513.

Su, Y.; Liu, Q.; Liu, Q.; Huang, Z.; Yin, Y.; Chen, E.; Ding, C.; Wei, S.; and Hu, G. 2018. Exercise-enhanced sequential modeling for student performance prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Wang, T.; Ma, F.; and Gao, J. 2019. Deep hierarchical knowledge tracing. In *Proceedings of the 12th International Conference on Educational Data Mining*.

Wilson, K. H.; Karklin, Y.; Han, B.; and Ekanadham, C. 2016. Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. In Barnes, T.; Chi, M.; and Feng, M., eds., *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, 539–544. International Educational Data Mining Society (IEDMS).

Yang, Y.; Shen, J.; Qu, Y.; Liu, Y.; Wang, K.; Zhu, Y.; Zhang, W.; and Yu, Y. 2021. GIKT: A Graph-Based Interaction Model for Knowledge Tracing. In Hutter, F.; Kersting, K.; Lijffijt, J.; and Valera, I., eds., *Machine Learning and Knowledge Discovery in Databases*, 299–315. Cham: Springer International Publishing. ISBN 978-3-030-67658-2.

Yudelson, M. V.; Koedinger, K. R.; and Gordon, G. J. 2013. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*, 171–180. Springer.

Zhang, J.; Shi, X.; King, I.; and Yeung, D.-Y. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, 765–774.