Knowledge Tracing in Intelligent Tutoring System

Runyu Wang

I. Introduction

The notion of a computer-assisted instruction system teaching students using AI technique was first raised by Jaime Carbonell in 1970. 20 years later, in early 1990s, a consensus on the infrastructure of such system has formed, as well as the name Intelligent tutoring system (ITS). Such infrastructure is still at the core of most ITSs today.

An ITS consists of 4 fundamental modules: an expert knowledge module that maintains the domain knowledge of the system, a student model module that keeps track of learner information relevant to learning in the system, a tutoring module that houses teaching algorithm, and the user interface module that students interact and learn from [1].

In this paper, I will focus on the student model component in ITS, which is the most complex yet important component of the system [2]. A good student model can provide precise feedback to the tutoring module, and this results in effective reinforcement and positive learner experience. On the other hand, even small deviation from actual student state could largely harm the effectiveness of such system. This is because in most domains, learning efficiency is highly precedence-dependent. In this paper, I investigate three milestone knowledge tracing models in the field of ITS: Bayesian knowledge tracing, deep knowledge tracing, and dynamic key-value memory network.

II. Student Model Module

The main tasks of the student model module are:

1. To maintain the student's personal information.
2. To predict the domain knowledge state of the participating student

Although some ITSs do take in student's personal information as information input, task 1 is often more logistical than technical. Task 2 is usually the more interesting and crucial responsibility of the student model module in an ITS.

To be more specific, a student model module is in charge of predicting the knowledge state of a student in ITS as a subset of the expert module, more specifically it predicts whether a student can correctly apply a certain piece of knowledge. This task is also known as knowledge inference. Student models face great challenges in tracing knowledge, because human minds are complex, and different humans could vary greatly in terms of learning ability and learning habit. There are also real-world factors such as forgetting curving that makes things even harder.

III. Bayesian Knowledge Tracing model

Before BKT

The earliest approach that ambiguously handled this task is a naïve test-based method that utilizes pop quizzes to determine whether a concept has been mastered by the student. However, this method is inefficient and inaccurate as it is hard to pinpoint the core precedent concept that causes the error in the testing of a later concept.

Hidden Markov Model as Student Model

A Bayesian knowledge tracing model is the first practical system that attempts the responsibility of a student model module in ITS. First incepted by Atkinson in 1972 and concretely formalized in 1995 by Corbett and Anders, Bayesian knowledge tracing model (BKT) has been the state-of-the-art knowledge tracing method before the emergence of recurrent neural network in mid 2010s. A BKT is in essence a hidden Markov model (HMM). An HMM is a temporal probabilistic model, with each of its states being a single discreate representation of the world state [3]. Hinted by the name, the states in an HMM are hidden, but we are able to observe indirect evidences and make inference of current or past state. In the context of knowledge tracing, the state that we are trying to infer is whether student has mastered a specific skill. And the evidence is the student's performance on applying the knowledge(quizzes).

Formal Definition and Usage of a BKT

Let's formalize Bayesian knowledge tracing model by defining its parameters and assumptions [4]-[6]:

- An item in BKT is specific to a single well-defined skill Q.
- State of Q is binary: it is either mastered or not mastered
- A student's inferred probability of mastering Q at time n is $P(L_n)$
- The probability of a student applying Q correctly is $P(Corr)$.
- The probability of a student mastering Q before first applying it, $P(L_0)$.
- The probability of a student mastering Q after each time applying it, $P(T)$.
- The probability of a student guessing correctly without mastering Q, $P(G)$.
- The probability of a student slipping and answering incorrectly even when he/she has already mastered Q, $P(S)$.
- Once a student has mastered Q, she/he never forgets it.

Note that a BKT Model holds the memoryless property of an HMM, meaning that a student's mastery of Q at time n is only dependent on that at time n-1. And the four parameters, $P(L_0)$, $P(T)$, $P(G)$, and $P(S)$ are to be acquired by machine learning in a modern setting. Armed with these assumptions and parameters, we can infer $P(L_n)$ and $P(Corr)$ using the following formulas:

$$P(L_n) = P(L_{n-1}|evidence) \ + \ (1 - \ P(L_{n-1}|evidence)) \times P(T)$$

Using Bayes' Theorem [7], $P(L_{n-1}|evidence)$ can be calculated based on the evidence (correct or incorrect):

$$P(L_{n-1}|Correct_n) = \frac{P(L_{n-1}) \times (1 - P(S))}{P(L_{n-1}) \times (1 - P(S)) + (1 - P(L_{n-1})) \times P(G)}$$

$$P(L_{n-1}|Incorrect_n) = \frac{P(L_{n-1}) \times P(S)}{P(L_{n-1}) \times P(S) + (1 - P(L_{n-1})) \times (1 - P(G))}$$

Note that in the formular for $P(L_{n-1}|Correct_n)$ the denominator consists of two parts, the first half $P(L_{n-1}) \times (1 - P(S))$ represents the case that the student correctly applies Q through mastery; and the second half $(1 - P(L_{n-1})) \times P(G)$ denotes the case that student correctly applies Q through guessing. Vice versa, the similar logic applies to the formula for $P(L_{n-1}|Incorrect_n)$.

Variants of BKT

Over the years, many more advanced variants of BKT were introduced. Beck's Help Model [8] introduces the notion of help from tutoring module, which helps refining the parameters, depending on whether help was requested. Contextual Guess and Slip model digs into the notion of slipping and treating it more as evidence than stochastic behavior [9]. Bayesian diagnosis tracing model [10] investigates incorrect answers in detail to capture the cause of the mistakes. There are many more variants of BKT, and many advanced BKT models have their places in specific domains. However, BKT has a strong assumption: it assumed that once a student has mastered Q, she/he never forgets it. This is far from the truth, as most learners have experienced forgetting previously mastered skill [11]. BKT is a crude simulation to how our memory system works.

IV. Recurrent neural network (RNN) and Deep knowledge tracing (DKT)

The application of artificial neural network (referred as neural network in this paper) in knowledge tracing marks the modern age of the field. With the power of deep learning techniques, knowledge tracing models became more capable at modeling student memory.

Neural Network and Deep Learning

Deep learning is a branch of machine learning technique that utilizes neural network models to perform complicated AI tasks. Neural network is a family of learning models that were inspired by "the early model of sensory processing by the human brain" [12]. Before neural network took over the fields of AI, learning models were built using a more primitive method, logistic regression. Logistic regression is a fundamental machine learning technique. It works well for classifying linearly separated data, but not as well when there are many features in the model, and the learning process is naïve compared to neural network. Neural networks are capable of learning non-linear model functions, and they can handle much more demanding tasks such as image recognition and speech/text completion [13]. A neural network takes training data to find the optimized parameters. This process is known as learning.

Infrastructure of a Neural Network

The atomic component of a neural network is a perceptron. (Figure 1) It is represented as a node in the data flow. It combining inputs from inward perceptrons, and send the result to outward perceptrons. A neural network can have multiple perceptrons as input source, and it can have multiple perceptrons as output terminal. The middle layers are also known as the hidden layers. A neural network can have more than 1 hidden layers. Note that in figure 2, all the arrow are pointing to the next layer until output.

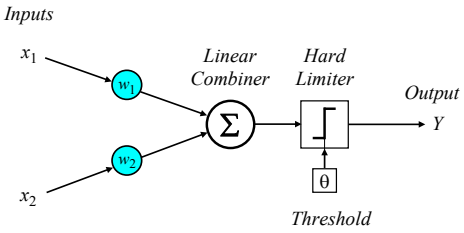

Figure 1: a perceptron [14]

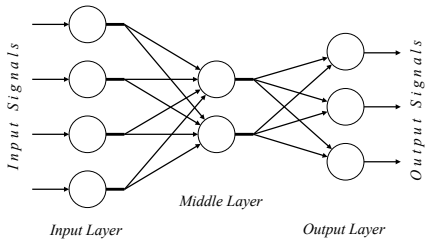

Figure 2: a Feedforward Network [14]

Recurrent Neural Network and Long-Short Term Memory Model

Recurrent neural network is one of the most widely-used member in the neural network family. Its major distinction from other neural networks is that it also maintains a recurrent state, that is the previous computation state, allowing it to revisit its previous computation. This mechanism is crucial to model student memory as a proper student model should have incorporate previous knowledge states. At any time, t, the output h_t is processed from the previous computation vector h_{t-1} and the input vector at time t, x_t. See figure 3.

An important variant of RNN for knowledge tracing is Long-short term memory (LSTM). LSTM model maintains a cell state C_t that keeps track of necessary information from previous outputs until time t. See figure 4. This cell state streamline functions as the long-term memory of the system. The storage, selection, and update of cell state is achieved using control gates (nodes with "x" or "+" mark). Note that yellow boxes in figure4 are activation function that serves to control the magnitude of data flow [3], [15]. Compared to plain RNN, LSTM is more similar to human memory system, as it has a separate memory stream C_t that keeps track of long-term memory.

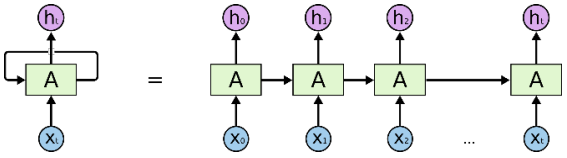

Figure 3: a Recurrent Neural Network [15]

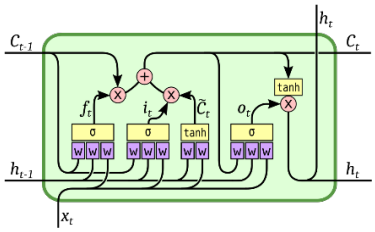

Figure 4: Inside a LSTM [15]

Piech et al. [16] first applied LSTM in a knowledge tracing model, which is known as deep knowledge tracing model (DKT). The parameters to be learned are the purple boxes marked "w", the weight matrices. Note that there are 11 different weight matrices. And they are learned using supervised learning. A loss function measures the deviation of the prediction from the truth. Optimal parameters minimize the loss function of the model. The optimal parameters are learned using gradient descent and backpropagation through time on the loss function. At the terminal of each time state, a probability is generated for knowledge inference purpose. By the nature of LSTM, both the correctness of the student answer and the previous inference result are passed on to the next state. Even with the power of memory storage from LSTM, such model maintains only a single memory matrix at any time state. Meaning at any time, the system is only aware of the memory state at that specific time: it cannot utilize detailed information from the past [16].

Attention Mechanism

To combat this shortcoming, newer iterations of DKT models often deploy attention mechanism. Through an independent neural network learning, an attention weight can help signal the system of the past memory components with the highest correlation to input. In practice, this links a testing question to the previous memory state(s) where similar concept(s) were tested.

Performance of DKT

To predict a student correctly answering a question at time t in DKT models, the string of input (student exercise data) is taken into a model, and output h_t can be queried after r iterations of transition. Research shows that DKT models with attention mechanism outperform almost all previous knowledge tracing model in the task of knowledge inference [17]. Furthermore, a well-trained DKT model can improve the curricula by suggesting the most promising sequence of learning activity based on the learned weight matrices. It can also be used to discover connections between exercises [16].

DKT's success looks promising, but it has serious downsides. First of all, it is extremely inefficient. Due to the recurrent nature, parallelism cannot help booster the computation, and the computation size is enormous. Secondly, as memory states increase over time, it becomes more demanding to manage the previous memory states. This inefficiency in storage could further impede efficiency in computation, especially over long arc of learning. Finally, the computations are clustered in one global state. This means that although the performance of DKT trumps that of BKT, it cannot provide detailed information about any particular concept.

V. Dynamic Key-Value Memory Networks for Knowledge Tracing

Dynamic Key-Value Memory Networks (DKVMN) answers some of these issues present in DKT models. The key intuition behind DKVMN is to maintain mastery of each specific concepts separately, as human brains categorize knowledge internally.

Memory-Augmented Neural Network

Memory-augmented neural network (MANN) is an intermediate step toward DKVMN. A MANN is a variant of RNN, and its key feature is an external memory module. Under a knowledge tracing context, such memory module includes two parts: a memory matrix maintaining the information of current knowledge state and a control unit that observes the environment and read from or write to the memory matrix. Such structure is fit for ITS knowledge tracing: it mimics human mind better than DKT as it separates the functions of memory storage and controlling.

Attention mechanism is also utilized in determining the specifics in reading/writing. The memory matrix is a $N \times M$ matrix, where N is the number of different concepts, and M is the number of memory states. See figure 5 for a MANN model [19].

With the external memory module, MANN addresses the issue of memory storage in DKT since memory data is stored more efficiently as a separate piece of data. The matrix structure of memory also allows MANN to keep track of concept-specific information. However, there is a subtle drawback for MANN as a knowledge tracing model. Since there is a singular matrix storing the memory information, it means the model reads and writes in the same space. Note that a MANN model reads the exercises the student receives, and writes the correctness of the student's answer. These two values shouldn't have a uniform space as they represent different types of data [17].

DKVMN: A Step Further from MANN

DKVMN utilizes key-value pair matrices to answer this issue. Key matrix is of dimension $N \times k$ where N is the number of different concepts, and k is the number of attributes(features) of each concept. It contains information about each concept, and is immutable. It serves as a separate attention unit, that finds the corresponding weight of each concept underlying the current input question, and then pass it to output and writing. Value matrix is of dimension $N \times v$ where N is the number of different concepts, and v is the number of memory states. Value matrix is similar to the memory matrix in a MANN model, particularly its role as memory storage. See figure 6 for a DKVMN model. Comparing with figure 5, we can see that the biggest difference between MANN and DKVMN is the extra key matrix as attention mechanism [17], [20]. The function key matrix is analogous to the human brain process of recognizing and recalling concepts associated to new questions.

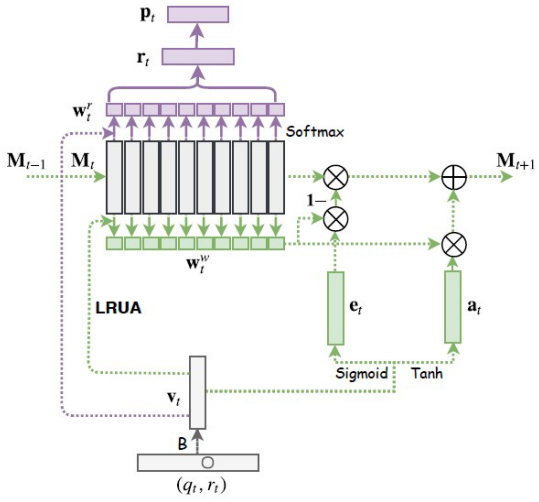

Figure 5: inside a MANN model [17]

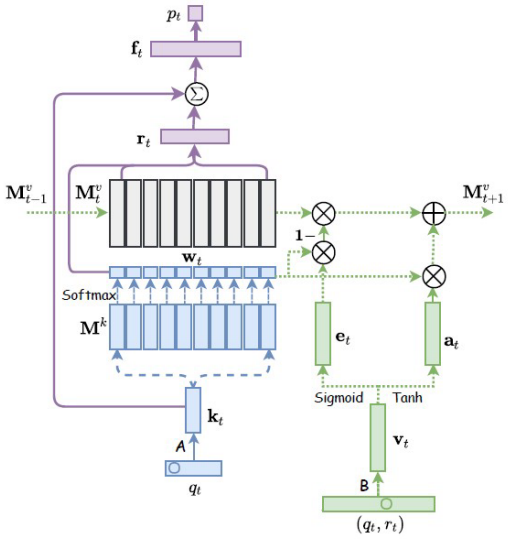

Figure 6: Inside a DKVMN model [17]

The high-level flow of DKVMN is as follow:

1. When an input question is perceived, initialization component preprocesses input data and key matrix is used to calculate concept weight according to the key matrix. Then calculate read content from the weight and the value matrix, and make prediction of correctness according to read content and the input.
2. After the student has answered the question, according to the correctness, the model adjusts the value matrix in the related concepts, and move to the next state.

Let M^k denote the key matrix, and let M_t^v denote the value matrix at time t. When a DKVMN model gets an input question at time t, it is first adjusted to a compatible form k_t by an initialization component, and processed through M^k to get w_t, the weight of concepts in the input question. Then r_t, read content is concatenated with input k_t to generate W. This process is to maintain the difficulty level in the question within the calculation. Finally, W is passed in to a series of activation function to get the prediction of the student correctly solving the input question [17], [20].

Note the model has an erase unit, e and an add unit, a. The erase unit can erase certain memory stored in the value matrix, and add unit can add certain memory stored in the value matrix. After the student answered, each of the two unit will decide whether they do their job and make change to M_t^v. These two units can work at the same time, when the answer student provided suggests that some concepts were correctly applied, while others were not. Finally, the moderated value matrix is passed on to next time state as M_{t+1}^v. And that is the full circle.

DKVMN Performance

Early experiment by Zhang et al. [17] showed the competence of DKVMN model in comparison to MANN, DKT, BKT, and optimal variations of BKT(BKT+). The experiment was run on 4 separate test sets, where Synthetic-5 is simulated data, and other three datasets were real-world data from online learning environments.

Test set	Knowledge Tracing Model AUC (in %)				
	DKVMN	MANN	DKT(LSTM)	BKT	BKT+
Synthetic-5	**82.7**	81.0	80.3	62	80
ASSISTments2009	**81.6**	79.7	80.5	63	N/A
ASSISTments2015	**72.7**	72.3	72.5	64	N/A
Statics2011	**82.8**	77.6	80.2	73	75

Table1: Performance of different KT models over 4 test sets [17]

VI. Conclusion

In this paper, we looked at three major knowledge tracing models, Bayesian knowledge tracing (BKT), deep knowledge tracing (DKT), and dynamic key-value memory network (DKVMN). As test result shows, DKVMN is the most robust knowledge tracing method we have at hand. Other studies also showed that DKVMN structure is robust compared to others [21]. In the future, we can expect to see a lot of variations of DKVMN, and DKVMN will be a baseline for knowledge tracing performance comparison for quite some time.

References

[1] Nwana, H.S. Intelligent tutoring systems: an overview. *Artif Intell Rev* **4,** 251–277 (1990). https://doi.org/10.1007/BF00168958

[2] C. Yang, F.-K. Chiang, Q. Cheng, and J. Ji, "Machine Learning-Based Student Modeling Methodology for Intelligent Tutoring Systems," Journal of educational computing research, vol. 59, no. 6, pp. 1015–1035, 2021, doi: 10.1177/0735633120986256.

[3] Russell, Stuart J. (Stuart Jonathan). Artificial Intelligence: a Modern Approach 4th edition. Hoboken, N.J., Prentice Hall, 2021. ISBN 9780134610996

[4] Corbett, A.T., Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model User-Adap Inter* **4,** 253–278 (1994). https://doi.org/10.1007/BF01099821

[5] Atkinson, R. C. (1972). Optimizing the learning of a second-language vocabulary. *Journal of Experimental Psychology, 96*(1), 124–129. https://doi.org/10.1037/h0033475

[6] Atkinson, R. C., & Paulson, J. A. (1972). An approach to the psychology of instruction. *Psychological Bulletin, 78*(1), 49–61. https://doi.org/10.1037/h0033080

[7] Joyce, James, "Bayes' Theorem", *The Stanford Encyclopedia of Philosophy* (Fall 2021 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/fall2021/entries/bayes-theorem/>.

[8] Beck J.E., Chang K., Mostow J., Corbett A. (2008) Does Help Help? Introducing the Bayesian Evaluation and Assessment Methodology. In: Woolf B.P., Aïmeur E., Nkambou R., Lajoie S. (eds) Intelligent Tutoring Systems. ITS 2008. Lecture Notes in Computer Science, vol 5091. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-69132-7_42

[9] J. Ocumpaugh, R. Baker, S. Gowda, N. Heffernan, and C. Heffernan, "Population validity for educational data mining models: A case study in affect detection," British journal of educational technology, vol. 45, no. 3, pp. 487–501, 2014, doi: 10.1111/bjet.12156.

[10] J. Feng, B. Zhang, Y. Li, and Q. Xu, "Bayesian Diagnosis Tracing: Application of Procedural Misconceptions in Knowledge Tracing," in Artificial Intelligence in Education, Cham: Springer International Publishing, 2019, pp. 84–88.

[11] B. A. Richards and P. W. Frankland, "The Persistence and Transience of Memory," Neuron (Cambridge, Mass.), vol. 94, no. 6, pp. 1071–1084, 2017, doi: 10.1016/j.neuron.2017.04.037.

[12] A. Krogh, "What are artificial neural networks?," Nature biotechnology, vol. 26, no. 2, pp. 195–197, 2008, doi: 10.1038/nbt1386.

[13] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015, doi: 10.1016/j.neunet.2014.09.003.

[14] Gormley, Matt, Carnegie Mellon University Neural Networks slide: https://www.cs.cmu.edu/~mgormley/courses/10601b-f16/lectureSlides/lecture15-neural-nets.pptx

[15] MIT open course 6.S191: Introduction to deep learning lecture 2 Recurrent Neural Networks: https://www.youtube.com/watch?v=qjrad0V0uJE&list=PLtBw6njQRUrwp5__7C0oIVt26ZgjG9NI&index=2&ab_channel=AlexanderAmini

[16] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In Advances in Neural Information Processing Systems, pages 505-513, 2015.

[17] Zhang, J., Shi, X., King, I., & Yeung, D. (2017). Dynamic Key-Value Memory Networks for Knowledge Tracing. *Proceedings of the 26th International Conference on World Wide Web*.

[18] Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR, abs/1409.0473*.

[19] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48JMLR.org, 1842–1850.

[20] CS885 Reinforcement Learning by Pascal Pupart lecture 19c, Memory Augmented Networks https://cs.uwaterloo.ca/~ppoupart/teaching/cs885-spring18/slides/cs885-lecture19c.pdf

[21] X. Sun, X. Zhao, B. Li, Y. Ma, R. Sutcliffe, and J. Feng, “Dynamic Key-Value Memory Networks With Rich Features for Knowledge Tracing,” IEEE transactions on cybernetics, vol. PP, pp. 1–7, 2021, doi: 10.1109/TCYB.2021.3051028.