Deep Hierarchical Knowledge Tracing

Tianqi Wang
University at Buffalo
Buffalo, NY
twang47@buffalo.edu

Fenglong Ma
University at Buffalo
Buffalo, NY
fenglong@buffalo.edu

Jing Gao
University at Buffalo
Buffalo, NY
jing@buffalo.edu

ABSTRACT

Knowledge tracing is an essential and challenging task in intelligent tutoring systems, whose goal is to estimate students' knowledge state based on their responses to questions. Although many models for knowledge tracing task are developed, most of them depend on either concepts or items as input and ignore the hierarchical structure of items, which provides valuable information for the prediction of student learning results. In this paper, we propose a novel deep hierarchical knowledge tracing (DHKT) model exploiting the hierarchical structure of items. In the proposed DHKT model, the hierarchical relations between concepts and items are modeled by the hinge loss on the inner product between the learned concept embeddings and item embeddings. Then the learned embeddings are fed into a neural network to model the learning process of students, which is used to make predictions. The prediction loss and the hinge loss are minimized simultaneously during training process.

Keywords

knowledge tracing, hierarchical structure modeling, deep learning

1. INTRODUCTION

Knowledge tracing is an essential and challenging task in intelligent tutoring systems. The goal of knowledge tracing task is to estimate the mastery state of a specific knowledge component based on students' responses to items. In other words, knowledge tracing aims to predict the correctness of a student's response to the next item according to all the previous response records.

In order for a student to answer an item correctly, he/she needs to master the concepts related to this item first. For example, a student can provide correct responses to both "1 + 1" and "28 + 36", which illustrates that this student may master the general concept of addition. Some existing knowledge tracing models [1, 8, 9] are proposed to predict the students' performance only based on general concept information of items. A common drawback of such models is that they ignore the differences among different items even under the same general concept. In fact, if a student knows how to solve the items related to the concept "Addition of Two Integers", this student may correctly answer "28 + 36" but make a mistake when answering a harder item "285 + 361". In addition, the learning gains of answering different items related to the same concept are different. Correctly solving a more complex item indicates a higher gain towards the desired knowledge states than that obtained by solving an easier one. To distinguish and model the differences among items, Item Response Theory model [10] is proposed, which directly uses items as the input to estimate a student's ability. However, students may visit these online platforms very infrequently and only attempt on a small subset of items. Therefore, for each item in the dataset, only a small number of attempts are made, which leads to the issue of data sparsity. On such sparse data, existing knowledge tracing models, for example, Item Response Theroy [10], that takes items as input may have limited performance. Deep knowledge tracing [9], which applies RNN to predict the performance of students, has shown improved prediction performance in knowledge tracing, but it requires a large amount of data for training. Such models would suffer more from the data sparsity issue.

To handle the data sparsity issue and better distinguish items, we propose a novel deep hierarchical knowledge tracing (DHKT) model, which can leverage the hierarchical information between items and concepts. Specially, DHKT learns the embeddings of items and concepts and models the relations among items and concepts by calculating the hinge loss of the inner product of the embeddings. The main contribution of this work can be summarized as follows: We propose a novel DHKT model by leveraging the hierarchical structure between items and concepts into the state-of-the-art deep knowledge tracing model. Experimental results show that the DHKT model learns meaningful representations and outperforms the state-of-the-art baselines.

2. METHODOLOGY

In this section, we first introduce how to model the students' learning process using a deep learning framework, and then illustrate how to incorporate the concept-item graph into the model.

2.1 Problem Formulation

We denote the set of students as $\mathcal{K}$. For a student $k \in \mathcal{K}$ interacting with the system t times, the interactions are denoted as $\mathcal{X}_k = \{x_{k,1}, x_{k,2}, ..., x_{k,t}\}$. In this work, the inter-

Tianqi Wang, Fenglong Ma and Jing Gao "Deep Hierarchical Knowledge Tracing" In: *The 12th International Conference on Educational Data Mining*, Michel Desmarais, Collin F. Lynch, Agathe Merceron, & Roger Nkambou (eds.) 2019, pp. 671 - 674

actions specifically refer to the students' responses to corresponding items. $x_{k,t} = (i_{k,t}, y_{k,t})$ is a tuple representing the item $i_{k,t} \in \mathcal{I}$ attempted by student k at time t, where $\mathcal{I}$ represents the set of all items, and $y_{k,t} \in \{0, 1\}$ represents the correctness of the response. $y_{k,t} = 0$ indicates an incorrect response and $y_{k,t} = 1$ represents a correct one.

These items are related to different knowledge concepts, which are the more general representations of the items. The set of knowledge concepts is denoted as $\mathcal{C}$ in this work. The relations between items and concepts are annotated by experts. We denote the mapping matrix, i.e., the concept-item graph, as $\mathcal{Q} \in \{0, 1\}^{|\mathcal{C}|\times|\mathcal{I}|}$. If $q_{m,n} = 1$, which means the item n is related to the concept m; otherwise, $q_{m,n} = 0$.

With the above definitions, the knowledge tracing task can be formulated as a binary sequence prediction problem: Given a series of interactions $\mathcal{X}_k = \{x_{k,1}, x_{k,2}, ..., x_{k,t}\}$, $x_{k,t} = (i_{k,t}, y_{k,t})$ of a student k, the next item $i_{k,t+1} \in \mathcal{I}$ and the concept-item mapping matrix $\mathcal{Q}$, our goal is to predict the value of $y_{k,t+1}$ representing if student k will answer the new given item $i_{k,t+1}$ correctly based on the current knowledge state $\mathbf{h}_{k,t}$ of the student.

2.2 Model Student Learning

We use dense embeddings instead of one-hot encoding as the input of the DHKT model. These dense embeddings can be automatically learned from the training dataset. To distinguish the difference among items related to the same concept and preserve the concept-level information, the concatenation of item embedding and concept level embedding is employed to represent an item in the DHKT model.

Let $\mathbf{e}_{i_{k,t}} \in \mathbb{R}^d$ denote the embedding of the item $i_{k,t}$, where d is the dimension of the embedding. Since one item $i_{k,t}$ can be related to more than one concept, we use the average embeddings of all the concepts related to $i_{k,t}$ as its concept-level embedding $\mathbf{e}_{c_{k,t}}$. Mathematically, the concept-level embedding for the item $i_{k,t}$ is defined as:

$$\mathbf{e}_{c_{k,t}} = \frac{1}{|\mathcal{C}_{k,t}|} \sum_{c_m \in \mathcal{C}_{k,t}} \mathbf{e}_{c_m}, \tag{1}$$

where $\mathcal{C}_{k,t}$ denotes the set of concepts related to item $i_{k,t}$, $|\mathcal{C}_{k,t}|$ is the number of such concepts, c_m represents a concept in $\mathcal{C}_{k,t}$, and $\mathbf{e}_{c_m}$ is the embedding of the concept c_m.

The new hierarchical representation of the item $i_{k,t}$ can be described as the concatenation of item and concept embeddings:

$$\mathbf{v}_{k,t} = \mathbf{e}_{i_{k,t}} \oplus \mathbf{e}_{c_{k,t}}, \tag{2}$$

where $\mathbf{v}_{k,t} \in \mathbb{R}^{2d}$ and $\oplus$ denotes vector concatenation.

To jointly represent the item and the correctness of student k's response, we introduce $\mathbf{a}_{k,t} \in \mathbb{R}^{4d}$ as:

$$\mathbf{a}_{k,t} = \begin{cases} \mathbf{v}_{k,t} \oplus \mathbf{0} & y_{k,t} = 1 \\ \mathbf{0} \oplus \mathbf{v}_{k,t} & y_{k,t} = 0 \end{cases}, \tag{3}$$

where $\mathbf{0} \in \{0\}^{2d}$ is the zero-vector. $\mathbf{a}_{k,t}$ is the input when we model the process of student learning.

In the student learning process, the current knowledge state of a student is highly correlated with the previous knowledge state and the learning gains from the new materials. Thus, the student learning process can be modeled by Long Short-Term Memory network [3] as:

$$\begin{aligned} \mathbf{g}_{k,t} &= \sigma(\mathbf{W}_i \mathbf{a}_{k,t} + \mathbf{U}_i \mathbf{h}_{k,t-1} + \mathbf{b}_i), \\ \mathbf{f}_{k,t} &= \sigma(\mathbf{W}_f \mathbf{a}_{k,t} + \mathbf{U}_f \mathbf{h}_{k,t-1} + \mathbf{b}_f), \\ \mathbf{o}_{k,t} &= \sigma(\mathbf{W}_o \mathbf{a}_{k,t} + \mathbf{U}_o \mathbf{h}_{k,t-1} + \mathbf{b}_o), \\ \mathbf{r}_{k,t} &= \mathbf{f}_{k,t} \otimes \mathbf{r}_{k,t-1} \\ &\quad + \mathbf{g}_{k,t} \otimes tanh(\mathbf{W}_c \mathbf{a}_{k,t} + \mathbf{U}_c \mathbf{h}_{k,t-1} + \mathbf{b}_c), \\ \mathbf{h}_{k,t} &= \mathbf{o}_{k,t} \otimes tanh(\mathbf{r}_{k,t}), \end{aligned} \tag{4}$$

where h denotes the dimensionality of hidden state vector. $\mathbf{g}_{k,t}, \mathbf{f}_{k,t}, \mathbf{o}_{k,t}, \mathbf{r}_{k,t}, \mathbf{h}_{k,t} \in \mathbb{R}^h$ are the activation vector of the input gate, forget gate, output gate, the memory cell and the hidden state vector of student k at time t. $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{W}_c \in \mathbb{R}^{h\times 2d}$ and $\mathbf{U}_i, \mathbf{U}_f, \mathbf{U}_o, \mathbf{U}_c \in \mathbb{R}^{h\times h}$ are weight matrices, $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o, \mathbf{b}_c \in \mathbb{R}^h$ is the bias vector which need to be learned during training. $\otimes$ denotes element-wise product. The $\sigma(\cdot)$ and $tanh(\cdot)$ denote the Sigmoid and Hyperbolic Tangent function seperately.

The correctness of a student's response to an item is dependent on both the current knowledge state of the student and the characteristics of the item. Thus we use the concatenation of student k's current knowledge state $\mathbf{h}_{k,t}$ outputted by the LSTM and the representation of item $i_{k,t+1}$ that denotes the characteristics of the item, i.e., $\mathbf{v}_{k,t+1}$, to make prediction. The concatenated vector is fed into a fully connected layer to obtain a summary vector $\mathbf{s}_{k,t}$, and then this vector is fed into a Sigmoid activation layer to calculate the probability of correctly answering item $i_{k,t+1}$ by student k. The process can be represented as:

$$\begin{aligned} \mathbf{s}_{k,t} &= \tanh(\mathbf{W}_{fc}(\mathbf{h}_{k,t} \oplus \mathbf{v}_{k,t+1}) + \mathbf{b}_{fc}), \\ p_{k,t+1} &= \sigma(\mathbf{W}_s \mathbf{s}_{k,t} + \mathbf{b}_s), \end{aligned} \tag{5}$$

where $\mathbf{W}_{fc} \in \mathbb{R}^{d_s\times(h+2d)}$, $\mathbf{W}_s \in \mathbb{R}^{d_s}$, $\mathbf{b}_{fc} \in \mathbb{R}^{d_s}$ and $\mathbf{b}_s \in \mathbb{R}^1$ are the weight matrices and biases to be learned, and d_s denotes the dimension of the summary vector. $p_{k,t+1}$ is the probability that student k can answer item $k + 1$ correctly.

2.3 Hierarchical Structure Constraint

In fact, there exists a concept-item graph between items and concepts. The concept-item graph provides us with the grouping information of the items. The items related to the same concept can be considered as belonging to the same group. They should be similar with other items within the same group, while dissimilar with items in other groups. At the same time, the concept should capture the characteristics of all items related to it and can approximately represent all these items.

Based on the above analysis, we introduce a hinge loss which tries to maximize the margins among different groups to model the hierarchical structure of items. We apply the embeddings of concepts to represent the general group characteristics of all the items related to the concept. When deriving the representations of items and concepts, we keep the item similar to its corresponding concepts, and on the contrary, make it far away from other concepts. Thus, the hinge loss between an item n and a concept m is defined as

$$l_h^{m,n} = \begin{cases} max\{0, 1 - \mathbf{e}_{i_n}^T \mathbf{e}_{c_m}\} & q_{m,n} = 1 \\ max\{0, 1 + \mathbf{e}_{i_n}^T \mathbf{e}_{c_m}\} & q_{m,n} = 0 \end{cases}, \tag{6}$$

where $\mathbf{e}_{i_n}$ and $\mathbf{e}_{c_m}$ are the embeddings of the item i_n and concept c_m, and $q_{m,n}$ is an indicator in the item to concept

mapping matrix $\mathcal{Q}$ indicating whether the item n is related to concept m. $q_{m,n} = 1$ indicates that item n is related to concept m, while $q_{m,n} = 0$ means that item n is not related to concept m.

2.4 Loss Function

The objectives of the proposed model are two folds: One is to make accurate predictions of students' responses, and the other is to learn meaningful embeddings of items. Based on these two objectives, in the training stage, we need to minimize the prediction loss and the hinge loss simultaneously.

For student k answering item $i_{k,t}$ at time t, the prediction loss $l_{k,t}$ can be modeled with the binary cross-entropy:

$$l_{k,t} = -(y_{k,t} \log(p_{k,t}) + (1 - y_{k,t}) \log(1 - p_{k,t})), \quad (7)$$

where $p_{k,t}$ is the probability that student k can answer item $i_{k,t}$ correctly and $y_{k,t}$ is the correctness of the response of student k at time t.

The total loss can be represented as the weighted sum of the total prediction loss and the total hinge loss:

$$\mathcal{L} = \sum_{k=1}^{|\mathcal{K}|} \sum_{t=1}^{t_k} l_{k,t} + \alpha \sum_{m=1}^{|\mathcal{C}|} \sum_{n=1}^{|\mathcal{I}|} l_h^{m,n}, \quad (8)$$

where t_k is the total number of questions that a student k attempted, and $|\mathcal{K}|$ is the number of students. $|\mathcal{I}|$ and $|\mathcal{C}|$ are the number of items and concepts in $\mathcal{I}$ and $\mathcal{C}$ respectively. $l_h^{m,n}$ is the hinge loss defined in Eq. 6. α is a hyper-parameter to balance the weight of prediction loss and hinge loss.

3. EXPERIMENTS

In this section, we present the experiments that evaluate the proposed DHKT model on knowledge tracing task. These experiments are performed on three real-world datasets.

3.1 Datasets

The ASSIST09[1] and ASSIT12[2] datasets were collected from the ASSISTments tutoring system [2]. In the experiment, we use skill builder dataset. In the preprocessing, we remove all duplicated records and the records without a skill id or without a skill name and the records without a $\{0, 1\}$ value of the "correctness" attribute. In addition, we also remove the students with a sequence length less than three. After preprocessing, there are 3,991 students, 227,156 records, 13,876 items and 96 concepts in the ASSIST09 dataset. The ASSIST12 dataset contains 270,66 students, 2,541,201 records, 45,716 items and 245 concepts after preprocessing.

The Statics dataset[3] was collected from a college-level engineering statics course [5]. This dataset includes transactions from two different modes: tutor mode and assessment mode. Since the transactions from assessment mode for the same student have the same timestamp, we cannot determine the order of these transactions. Thus, only the transactions from tutor mode are included in the experiment. Also, we remove the students with less than three transactions. After preprocessing, there are 317 students, 137,711 records, 987 items and 280 concepts. The statistics of the three datasets are shown in Table 1.

[1] https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data

[2] https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect

[3] https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507

Table 1: Overview of the Three Datasets.

	ASSIST09	ASSIST12	Statics
# of students	3,991	27,066	317
# of items	13,876	45,716	987
# of concepts	96	245	280
# of records	227,156	2,541,201	137,711
attempts per student	57	94	434
items per concept	145	187	4
attempts per item	16	56	140
attempts per concept	2,366	10,372	492

3.2 Baselines and Experimental Settings

To evaluate the effectiveness of the DHKT model, we compare the DHKT model with the state-of-the-art knowledge tracing models, including Item Response Theory (IRT) [10], Hierarchical Item Response Theory (HIRT) [10], Performance Factor Analysis (PFA) [8], Bayesian Knowledge Tracing (BKT) [1] and Deep Knowledge Tracing (DKT) [9]. IRT, HIRT, which is a Bayesian extension of IRT by considering the hierarchical structures between concepts and items, and PFA make predictions based on the logit function. BKT models the sequence of responses and makes predictions based on the Hidden Marcov Model. DKT applies RNN to model the response sequence and make predictions. To evaluate the effect of incorporating concept-item graph in deep knowledge tracing, the variations of the proposed DHKT model: EDKT, Fine-grained EDKT and DHKT-, are also compared. These variations share the same network structure with DHKT, but they are different in terms of the input and the value of α. EDKT and Fine-grained EDKT use the item embedding and the concept embedding as the input seperately and $\alpha = 0$. DHKT- is the reduced model of DHKT where only the item embedding is used as input in Eq. (2). In the ASSIST datasets the skill_id is considered as the concept and the problem_id is considered as the item. In the Statics dataset, the problem_name is the concept and the step_name is the item. The PFA, BKT, DKT and EDKT take concepts as input while IRT, Fine-grained EDKT and DHKT- take items as input. The concept-item graphs are constructed according to the relations between items and concepts and are used by HIRT and DHKT.

We split each of these datasets into training and testing datasets on student level. For each dataset, we randomly select 20% of the students as the testing dataset and keep 80% of the students as the training dataset to learn the parameters. We randomly select 20% of the training students for validation. The training and testing datasets for all the models are the same. For training the DHKT model, batch size is set to 32, and the number of epochs is set to 100. The hidden state dimensionality h is set to 100. The hyper-parameters are tuned on the validation datasets. We tune the embedding dimensinality and the balance parameter α using grid search. The candidate values for embedding dimensinality d are $\{25, 50, 100\}$, and α's in Eq. (8) are $\{0.001, 0.01, 0.1, 1\}$. The loss function is optimized by Adam algorithm [4], which is a gradient-based optimization algorithm based on adaptive estimates of lower-order moments. We set the learning rate to 0.01. To avoid the exploading gradient problem, gradient norm clipping strategy [7] is

Table 2: AUC Values on the Three Datasets.

Model	ASSIST09	ASSIST12	Statics
IRT	0.6891	0.7317	0.8249
HIRT	0.6912	0.7234	0.8251
PFA	0.7040	0.6706	0.7705
BKT	0.6722	0.6141	0.7318
DKT	0.7483	0.7346	0.7736
EDKT	0.7513	0.7310	0.7823
Fine-grained EDKT	0.7342	0.7428	0.8251
DHKT-	0.7780	0.7677	0.8311
DHKT	**0.7866**	**0.7747**	**0.8333**

adopted at the threshold of 20. We use dropout with a probability of 0.6 to alleviate the overfitting issue.

The Area Under ROC Curve (AUC) is used to evaluate the performance of all the models, in which ROC curve plots true positive rate versus false positive rate in a binary classification task. For each model, we run five times with random initialization and report the average AUC. The AUC of the testing dataset is calculated using the model with the highest validation AUC value among 100 epochs.

3.3 Results

The AUC values for different models on all three datasets are shown in Table 2. The proposed DHKT model achieves the highest AUC on all the three datasets. The reduced model DHKT- cannot beat DHKT, but it still outperforms all other baselines on the three datasets. The improvement from Fine-grained EDKT to DHKT- demonstrates the effectiveness of incorporating the concept-item graph. The difference in the performance between DHKT and DHKT- indicates that exploiting general level information is useful in deep knowledge tracing.

3.4 Embedding Visualization

We use t-SNE [6] to visualize the learned embeddings of items by DHKT in a 2-D space to qualitatively assess the intepretability of the representations. For comparison, we also plot the learned item embeddings on the three datasets of the Fine-grained EDKT. The color of the dots represents the concept related to the items.

The learned embeddings are shown in Figure 1. Figure 11(a), 1 1(b) and 11(c) show the learned item representations of DHKT on ASSIST09, ASSIST12 and Statics, and Figure 11(d), 1 1(e) and 11(f) show the learned item representations of the Fine-grained EDKT model on corresponding datasets. Compared with the items mixed together learned by Fine-grained EDKT, the item embeddings learned by DHKT are well separated and more consistent with the hierarchical structures on the ASSIST09 and ASSIST12 datasets. On the Statics dataset, although some clusters are mixed with each other, the representations learned by DHKT are much better than that learned by Fine-grained EDKT. In addition, the prediction performance of DHKT is better than that of Fine-grained EDKT on the three datasets, which demonstrates the importance of meaningful item representations for knowledge tracing.

4. CONCLUSIONS AND FUTURE WORK

In this work, we propose a novel deep hierarchical knowledge tracing model by incorporating the hierarchical structure of items. The proposed model not only improves the performance of knowledge tracing task, but also provides

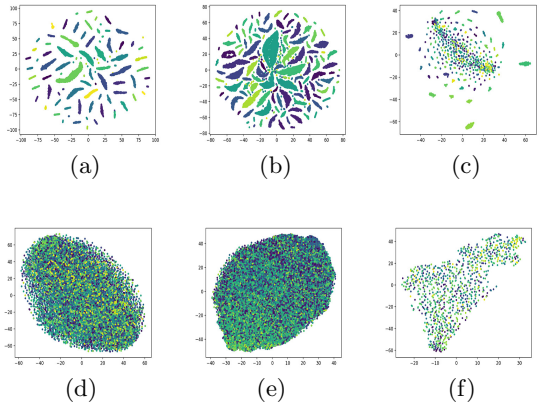

Figure 1: Item representations learned by DHKT and Fine-grained EDKT.

meaningful representations of items. The item representations learned by the proposed model are consistent with the hierarchical structure of the items. The superior prediction performance indicates that the hierarchical structure of items plays an important role in deep knowledge tracing, and meaningful representations can help improve deep knowledge tracing performance.

We plan to investigate how to apply multi-level hierarchical structures in knowledge tracing and how to recommend learning materials and items to students based on their knowledge state in the future.

5. ACKNOWLEDGEMENTS

This work is sponsored by NSF IIS-1553411. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

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

[2] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction*, 19(3):243–266, 2009.

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

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] K. R. Koedinger, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper. A data repository for the edm community: The pslc datashop.

[6] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[7] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[8] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. *Online Submission*, 2009.

[9] 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.

[10] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. *arXiv preprint arXiv:1604.02336*, 2016.