Graph-Enhanced Multi-Activity Knowledge Tracing

Siqian Zhao(✉)[0009−0008−3913−7836] and Shaghayegh Sahebi[0000−0002−8933−3279]

Computer Science, University at Albany - SUNY, Albany, NY 12222, USA
{szhao2, ssahebi}@albany.edu

1 Supplementary Material

1.1 Assessed-only Baselines Extension Detail

DKT+M [3] extends DKT by adding an feature upon the assessed material embeddings that concatenates the embedding vectors of all non-assessed materials that a student has interacted with between two assessed activities. **DKVMN+M** extend DKVMN that employs this same strategy to handle non-assessed learning activities.
SAINT+M [1] is a variant of SAINT that includes the activity position embedding as a feature, in addition to summarizing all non-assessed material embeddings that occur between two assessed activities as an additional feature. **AKT+M** and **AKT+M** are variants of SAKT and AKT that follow this same strategy to incorporate non-assessed learning activities.

1.2 Implementation Detail

The proposed method is implemented in PyTorch [1] using the Adam optimizer to learn the model parameters. All model parameters are initialized randomly using the Gaussian distribution with a mean of 0 and a standard deviation of 0.2. The norm clipping threshold is applied to avoid gradient explosion. We truncate or pad all sequences to a fixed length, and sequence length (L_s) is treated as another hyperparameter. If a sequence is longer than L_s, it is truncated into multiple sequences, and if it is shorter than L_s, it is padded with 0's up to length L_s.

1.3 Student Knowledge State Visualization

In this section, we would like to see how GMKT works in discovering student knowledge states. To achieve this, we calculate a student's knowledge state for each concept using equation 10 with a masked $\boldsymbol{w}_t$ and $\boldsymbol{W}_f$. To be specific, at each time step, we use $\tilde{\boldsymbol{w}}_t = [0, ..., w_i, ..., 0]$ to calculate the read content for latent concept i as $\tilde{\boldsymbol{c}}_t = \sum i = 1^N \tilde{w}_t(i) \boldsymbol{M}^v_t(i)$. Then, we use $\tilde{\boldsymbol{c}}_t$ to compute

[1] https://pytorch.org/

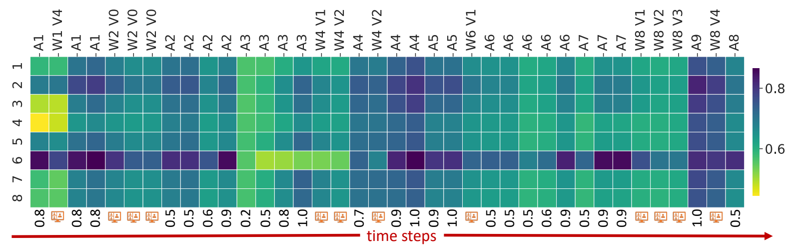


Fig. 1: Visualization of knowledge state for a sample student in the MORF dataset. The top x-axis ticks are learning material titles the student has tried at each time step. The bottom x-axis ticks are real student performance (in assessed activities) or the 'screen' icon (in non-assessed ones). The y-axis ticks are latent concepts.

the summary vector, which is $\tilde{\boldsymbol{f}}_t = Tanh([\tilde{\boldsymbol{W}}_f, 0]^\intercal [\tilde{\boldsymbol{c}}_t \oplus \boldsymbol{q}_t] + \boldsymbol{b}_f)$. Finally, we use equation 12 to calculate the knowledge state of each concept. We plot the heatmap in Figure 1 to show the knowledge state of a sample student from the MORF dataset. Top x-axis indicates the title of the attempted learning activity, where we use abbreviations "W* V**" to represent the video lecture ** of week * and "A*" to represent the Assignment of week *. Bottom x-axis shows the student's performance for an assignment attempt or a 'screen' for a video lecture attempt. y-axis represents the latent concept. Each cell shows the student's knowledge state of each concept after each attempt.

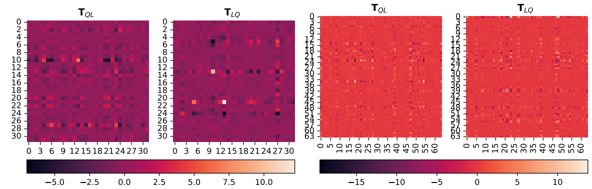
We first observe that the student's knowledge state fluctuates throughout the learning process. This is because the student's assignment grades are unstable. Typically, students receive low grades on their first attempts at assignments, such as A2 and A3, which may lead the GMKT model to learn that their knowledge has decreased for certain concepts at low grade attempts. However, as the student practices more and receives better grades, their corresponding knowledge increases. For instance, when the student received a good grade (0.9) as they keep practicing, their knowledge of concept six improves a lot. Our experiments indicate that student knowledge does not always increase, and understanding their learning trajectory is crucial [2, 4, 5].

1.4 Knowledge Transfer Matrices For EdNet and Junyi

The heatmaps of transition weight matrices for EdNet and Junyi dataset are shown in 2.

References

1. Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the 7th ACM Conference on Learning at Scale*, pages 341–344, New York, NY, USA, 2020. ACM.



(a) EdNet (b) Junyi

Fig. 2: Heatmaps for comparison of the weight matrices $\boldsymbol{T}_{QL}$ (assessed to non-assessed type) and $\boldsymbol{T}_{LQ}$ (non-assessed to assessed type) for EdNet and Junyi dataset.

2. C. Wang, S. Zhao, and S. Sahebi. Learning from non-assessed resources: Deep multi-type knowledge tracing. *International Educational Data Mining Society*, 2021.
3. L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan. Incorporating rich features into deep knowledge tracing. In *Proceedings of the 4th ACM Conference on Learning at Scale*, pages 169–172, New York, NY, USA, 2017. ACM.
4. S. Zhao, C. Wang, and S. Sahebi. Modeling knowledge acquisition from multiple learning resource types. In *Proceedings of The 13th International Conference on Educational Data Mining*, pages 313–324. International Educational Data Mining Society, 2020.
5. S. Zhao, C. Wang, and S. Sahebi. Transition-aware multi-activity knowledge tracing. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1760–1769. IEEE, 2022.