Evaluating DAS3H on the EdNet Dataset

Benoît Choffin, Fabrice Popineau, Yolaine Bourda, Jill-Jênn Vie

Evaluating DAS3H on the EdNet Dataset

Benoît Choffin,[1] **Fabrice Popineau,**[1] **Yolaine Bourda,**[1] **Jill-Jênn Vie**[2]
[1]Université Paris-Saclay, CNRS, CentraleSupélec
Laboratoire Interdisciplinaire des Sciences du Numérique, 91400, Orsay, France
[2] Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 – CRIStAL, F-59000 Lille, France
{benoit.choffin, fabrice.popineau, yolaine.bourda}@lisn.upsaclay.fr, jill-jenn.vie@inria.fr

Abstract

The EdNet dataset is a massive English language dataset that poses unique challenges for student performance prediction. In this paper, we describe and comment the results of our award-winning model DAS3H in the context of knowledge tracing in EdNet.

Introduction

Knowledge Tracing (KT) consists in modeling the evolution of a student's knowledge state as they are interacting with a set of items. Based on the past answers of a student, KT models allow to infer the probability that this student will correctly answer any other item. This task is essential in educational data mining as it helps personalize learning to suit every learner's needs.

A wide variety of KT models have been proposed by the past (Corbett and Anderson 1994; Yudelson, Koedinger, and Gordon 2013; Piech et al. 2015; Pavlik, Cen, and Koedinger 2009). Even though some of them take into account the relationships between items and skills (Pavlik, Cen, and Koedinger 2009; Gonzalez-Brenes and Mostow 2013) and others model the forgetting effect (Khajah, Lindsey, and Mozer 2016; Nagatani et al. 2019), only few models take both of these aspects into account.

We developed DAS3H (Choffin et al. 2019), a student model that explicitly accounts for memory decay and the benefits of practice when items can involve multiple skills at the same time. We showed on three different educational datasets that DAS3H outperforms four other state-of-the-art student models.

However, in our original paper, DAS3H was only tested on datasets of students learning math (Stamper et al. 2010; Feng, Heffernan, and Koedinger 2009). We mentioned that we would like to reproduce our experimental results on more diverse datasets, which Gervet et al. (2020) did. The Shared Task, organized by Riiid, was the opportunity to see if our results generalize to different datasets. Thus, in this article, we conduct the same set of experiments as in (Choffin et al. 2019), comparing DAS3H to four other student models, on the EdNet dataset.

This article is organized in the following way. We first review the different student predictive models that we compared in our experiments. Then we present the EdNet dataset and we detail the preprocessing and encoding steps that we performed. We report the results of our experiments on the EdNet dataset, followed by discussion.

Competing Models

To assess the generalizability of our results in (Choffin et al. 2019), we used the same set of student models: IRT, AFM, PFA, DASH and DAS3H. Based on previous interactions, each of these models predicts if a student will answer correctly a given item. In this section, we briefly review these models.

In what follows, we will index students by $s \in [\![1, S]\!]$, items (or questions, exercises) by $j \in [\![1, J]\!]$, skills or Knowledge Components (KCs) by $k \in [\![1, K]\!]$ and timestamps by $t \in \mathbb{R}^+$ (in seconds). To be more convenient, we assume that timestamps are encoded as the number of seconds elapsed since the first interaction with the system. $Y_{s,j,t} \in \{0, 1\}$ represents the binary correctness of student s answering item j at time t. We denote σ the logistic function:

$$\forall x \in \mathbb{R}, \sigma(x) = \frac{1}{1 + \exp(-x)}.$$

KC(.) takes as input an item index j and outputs the set of skill indices involved by item j.

Let us quickly detail what we mean by *skill*. In this article, we assimilate skills and knowledge components. A Knowledge Component (KC) is "a description of a mental structure or process that a learner uses, alone or in combination with other knowledge components, to accomplish steps in a task or a problem. [...] A knowledge component is a generalization of everyday terms like concept, principle, fact, or skill"[1]. An item may involve one or more KCs, and this information is synthesized by a so-called binary q-matrix (Tatsuoka 1983):

$$\forall (j, k) \in [\![1, J]\!] \times [\![1, K]\!],\ q_{jk} = \mathbb{1}_{k \in \text{KC}(j)}.$$

We assume that the probability of answering correctly an item j that involves KC k depends on the student's mastery of KC k.

[1]https://www.learnlab.org/research/wiki/Knowledge_component

IRT: Item Response Theory

IRT (Rasch 1961) is a student predictive model that stems from psychometrics. In its simplest form, IRT models the probability that student s answers correctly item j at time t as follows:

$$\mathbb{P}(Y_{s,j} = 1) = \sigma(\alpha_s - \delta_j)$$

with α_s ability of student s and δ_j difficulty of item j. The main assumption of IRT is that the student ability is static and does not change during the examination. Despite its apparent simplicity, IRT has proven to be a robust and reliable student model, even outperforming much more complex architectures such as Deep Knowledge Tracing (Piech et al. 2015) with minor modifications (Wilson et al. 2016).

AFM: Additive Factor Model

Other student models take the past history of student-item interactions into account to predict correctness probability. Contrary to IRT, these models assume that the probability that the student answers an item correctly may vary with practice. In particular, AFM (Cen, Koedinger, and Junker 2006) reads:

$$\mathbb{P}(Y_{s,j} = 1) = \sigma\left(\sum_{k \in \text{KC}(j)} \beta_k + \gamma_k a_{s,k}\right)$$

with β_k easiness of skill k and $a_{s,k}$ number of attempts of student s on skill k prior to this attempt. In AFM, the correctness probability depends on fixed skill parameters β_k but also on student practice: as they interact with a KC k, their correctness probability on k will vary according to γ_k.

PFA: Performance Factor Analysis

PFA (Pavlik, Cen, and Koedinger 2009) builds on AFM and uses past outcomes of practice instead of simple encounter counts:

$$\mathbb{P}(Y_{s,j} = 1) = \sigma\left(\sum_{k \in \text{KC}(j)} \beta_k + \gamma_k c_{s,k} + \rho_k f_{s,k}\right)$$

with $c_{s,k}$ number of correct answers of student s on KC k prior to this attempt and $f_{s,k}$ number of wrong answers of student s on KC k prior to this attempt. PFA uses a finer representation of past practice and modulates the effect of practice according to the binary outcomes of past interactions.

DASH: Difficulty, Ability and Student History

The formulation of DASH (Lindsey et al. 2014) reads:

$$\mathbb{P}\left(Y_{s,j,t} = 1\right) = \sigma(\alpha_s - \delta_j + h_\theta(\mathrm{t}_{s,j,1:\ell}, \mathrm{y}_{s,j,1:\ell-1}))$$

with h_θ a function parameterized by θ (learned by DASH) that summarizes the effect of the $\ell - 1$ previous attempts where student s reviewed item j ($\mathrm{t}_{s,j,1:\ell-1}$) and the binary outcomes of these attempts ($\mathrm{y}_{s,j,1:\ell-1}$).

Their main choice for h_θ is:

$$h_\theta(\mathrm{t}_{s,j,1:\ell}, \mathrm{y}_{s,j,1:\ell-1}) = \sum_{w=0}^{W-1} \theta_{2w+1} \log(1 + c_{s,j,w}) - \theta_{2w+2} \log(1 + a_{s,j,w})$$

with w indexing a set of expanding time windows, $c_{s,j,w}$ is the number of correct outcomes of student s on item j in time window w out of a total of $a_{s,j,w}$ attempts. The time windows w are not disjoint and span increasing time intervals. They allow DASH to account for both learning and forgetting processes. The use of log counts induces diminishing returns of practice inside a given time window and the difference of log counts formalizes a power law of practice. The time module h_θ is inspired by ACT-R (Anderson, Matessa, and Lebiere 1997) and MCM (Pashler et al. 2009) memory models.

Lindsey et al. make use of the additive factor models framework for taking memory decay and the benefits of past practice into account. Their model outperformed IRT and a baseline on their dataset COLT, consisting of student-item interactions on a Spanish vocabulary learning ITS.

DAS3H

Let us now describe DAS3H (Choffin et al. 2019). DAS3H stands for item Difficulty, student Ability, Skill, and Student Skill practice History and builds on DASH.

DAS3H was originally cast within the Knowledge Tracing Machines (Vie and Kashima 2019) framework. In this article, we only used DAS3H without pairwise interactions. In this situation, the quadratic term of KTM is cancelled out and our model DAS3H is simplified:

$$\mathbb{P}\left(Y_{s,j,t} = 1\right) = \sigma\Bigg(\alpha_s - \delta_j + \sum_{k \in \text{KC}(j)} \beta_k + h_\theta\left(\mathrm{t}_{s,j,1:\ell}, \mathrm{y}_{s,j,1:\ell-1}\right)\Bigg).$$

Following Lindsey et al., we choose:

$$h_\theta(\mathrm{t}_{s,j,1:\ell}, \mathrm{y}_{s,j,1:\ell-1}) = \sum_{k \in \text{KC}(j)} \sum_{w=0}^{W-1} \theta_{k,2w+1} \log(1 + c_{s,k,w}) - \theta_{k,2w+2} \log(1 + a_{s,k,w}).$$

Thus, the probability of correctness of student s on item j at time t depends on their ability α_s, the difficulty of the item δ_j and the sum of the easiness β_k of the skills involved by item j. It also depends on the temporal distribution and the outcomes of past practice, synthesized by h_θ. In h_θ, w denotes the index of the time window, $c_{s,k,w}$ denotes the amount of times that KC k has been correctly recalled in window w by student s earlier, $a_{s,k,w}$ the amount of times that KC k has been encountered in time window w by student s earlier. Intuitively, h_θ can be seen as a sum of memory strengths, one for each skill involved in item j.

The EdNet Dataset

In this section, we describe EdNet, the dataset on which we perform our experiments.

Dataset Overview

EdNet (Choi et al. 2020) consists in more than 2 years of student-item interactions collected on Santa, an Intelligent Tutoring System (ITS) that helps students prepare for the TOEIC (Test of English for International Communication) exam. To the best of our knowledge, EdNet is the largest public educational dataset available to date, with more than 131 millions of interactions and more than 784,000 users.

Students use Santa to self-study for improving their English and can practice on multiple platforms (Android, iOS and the Web). Santa provides students with video lectures, exercises and expert commentaries.

Four datasets, ranging from the least to the most granular, are available for EdNet. Since we focus in this article on the task of predicting students' correctness on items, we just needed to use the least granular dataset (KT1). Each row of this dataset contains the data concerning a given student-item interaction: e.g. timestamp of the interaction, user ID, item ID and user answer.

We needed another dataset for our experiments, containing metadata on the questions. In particular, we had to know the correct answer as well as the KC labels of each item in order to build the q-matrix. We used the `tags` column in this metadata dataset to get the skill labels of an item.

Preprocessing the Dataset

We preprocessed the dataset in the following order:

- we removed duplicate rows and rows for which the user answer was not available;
- we removed items for which the skill tag was equal to -1. Since all other skill tags were positive and that items with this skill tag had no other skill tag, we inferred that these were unknown skills. We assumed that they were too heterogeneous for being synthesized under a single abstract KC label;
- we built the (binary) q-matrix (Tatsuoka 1983) from the metadata dataset; some skills were mapped several times to an item, so we removed duplicates;
- we removed users which had less than 10 interactions with Santa.

Table 1 reports EdNet's characteristics after this preprocessing step. It also reports the characteristics of the datasets that we employed in our previous experiments. The mean KC delay refers to the mean time interval (in days) between two interactions with the same KC, and the mean study period refers to the mean time difference between the last and the first interaction for each student. We can see here that EdNet differs from the datasets that we employed in (Choffin et al. 2019) in multiple ways:

- it is significantly larger and contains many more users;
- it contains fewer items and these items involve more skills on average;
- users spend less time on the platform on average. Since students use Santa as a self-study tool, we hypothesize that their motivation solely determines the time they spend on the platform, contrary to an ITS like ASSISTments.

Encoding the Dataset

To train our five student models, we used the data encoding trick proposed by Vie and Kashima (2019). It consists in representing each row of the original dataset as a sparse vector containing all the features corresponding to this model, and running a standard machine learning algorithm (such as logistic regression) on this sparsely encoded matrix. For instance, IRT can be easily represented by one-hot encoding both users and items and concatenating the resulting columns. Please refer to Vie and Kashima (2019) for further details on this method.

Table 2 synthesizes the types of features that are used in each of the competing models.

Experiments

All Python code for reproducing our experimental results can be found on GitHub[2].

Experimental setting

To compare the performance of our competing models on the EdNet dataset, we use 5-fold cross-validation at the student level. This means that we split the student population into 5 disjoint groups of equal size: afterwards, we apply cross-validation on these 5 folds. As a consequence, students from the test folds are never seen by our models before the test phase. This allows us to reproduce the well-known cold-start problem when a new student arrives on an e-learning platform and to evaluate the models in this more difficult setting. Without any data on the test students, the models we compared fix these missing student-specific parameters (e.g. α_s in IRT) to their average values seen in the training set.

To help with the convergence of our models, we choose to normalize our data so that the maximum absolute value of each feature would be equal to 1. For this purpose, we use the `MaxAbsScaler` preprocessing tool from `scikit-learn`. Since a large part of our features were one-hot encodings (of users, items or KCs), this normalization only affects counter variables.

For DAS3H and DASH, we use the same time windows as Lindsey et al. (2014); Choffin et al. (2019): $\{1/24, 1, 7, 30, +\infty\}$ (in days).

We use the `scikit-learn` (Pedregosa et al. 2011) implementation of the logistic regression with L_2 regularization for our experiments. As we explained earlier, all the models we compare can be formulated as logistic regression models. We set the maximum number of iterations for the SAGA (Defazio, Bach, and Lacoste-Julien 2014) solver to 200 and fix the L_2 regularization parameter C to 1. The EdNet dataset blew up our RAM, resulting in slower computation; so we also tried to directly encode the features in the

[2]https://github.com/BenoitChoffin/das3h

Dataset	Users	Items	Skills	Interactions	Mean correctness	Skills per item	Mean KC delay (days)	Mean study period (days)
ednet	441,996	12,277	188	93,326,647	0.658	2.260	3.7	44.9
assist12	24,750	52,976	265	2,692,889	0.696	1.000	8.54	98.3
bridge06	1,135	129,263	493	1,817,427	0.832	1.013	0.83	149.5
algebra05	569	173,113	112	607,000	0.755	1.363	3.36	109.9

Table 1: Datasets characteristics after preprocessing

	users	**items**	**KCs**	**wins**	**fails**	**attempts**	**time windows**
DAS3H	✓	✓	✓	✓		✓	KC
DASH	✓	✓		✓		✓	items
IRT	✓	✓					∅
PFA			✓	✓	✓		∅
AFM	✓		✓			✓	∅

Table 2: Synthesis of the features considered by each student model that we compared

Model	AUC ↑	RMSE ↓	NLL ↓
DASH	**0.743**	**0.439**	**0.566**
DAS3H	0.733	**0.438**	**0.565**
IRT	0.716	0.449	0.588
PFA	0.632	0.463	0.618
AFM	0.608	0.466	0.626

Table 3: Performance comparison of the different student models. Metrics are averaged over the 5 folds. Standard deviations over the folds were all smaller than 0.001. ↑ and ↓ respectively indicate that higher (lower) is better.

liblinear format and ran the multicore liblinear solver (Lee, Chiang, and Lin 2015) with the same regularization parameter. This way, the DAS3H model took "only" 56 GB of RAM and we got the same results than the SAGA solver.

Finally, we use three different metrics for measuring the models' performance: AUC (Area Under the ROC Curve), RMSE (Root Mean Square Error) and NLL (mean Negative Log-Likelihood).

Results

We report our main experimental results in Table 3, which compares the three averaged performance metrics of all models. We did not report standard deviations across folds since they were all smaller than 0.001. We can see here that DASH outperforms all other models in terms of AUC, with an average AUC of 0.743. DAS3H and DASH perform on-par in terms of RMSE and NLL. Among the remaining models and for all three metrics, IRT fares better than PFA, which outperforms AFM.

These results are at odds with those that we reported on three other datasets in Choffin et al. (2019): although AFM, PFA and IRT were generally ranked in this order, DASH performed equally with IRT, and DAS3H substantially outperformed all the other models.

Discussion

What could explain the performance differences with our previous experiments?

We formulate several hypotheses that help explain the performance differences of DASH and DAS3H with our previous experiments (Choffin et al. 2019).

Knowledge domain All the other datasets on which we compared DAS3H consisted in students learning mathematics. On the Santa platform which collected the EdNet dataset, students prepare for an English exam, the TOEIC. Moreover, Lindsey et al. (2014) tested the DASH model on their COLT dataset, consisting in interactions of students learning Spanish: DASH substantially outperformed a hierarchical Bayesian version of IRT on this dataset. We can hypothesize that DAS3H may be less suited than DASH to model student learning of foreign languages.

Students' repetitions of identical items and KCs We built DAS3H based on the assumption that in datasets where the number of different items is high, students are less likely to practice multiple times a same item. In this situation, DASH would be less competitive since the counters of previous wins and attempts would be very sparse, and would perform similarly to IRT.

Here, this assumption might be wrong. We computed the number of interactions on which a student practiced multiple times the same item in EdNet and we found that these interactions represent 16.4% of all interactions. On ASSISTments 2012-2013, Algebra 2005-2006 and Bridge to Algebra 2006-2007, these interactions represent only, respectively, 4.7%, 4.8% and 1.1%.

We also computed the average number of KC repetitions, i.e. the average amount of repetitions of every KC in the dataset. In EdNet we found that on average, every KC was repeated 8.5 times by a student, whereas on ASSISTments 2012-2013, Algebra 2005-2006 and Bridge to Algebra 2006-2007, every KC was repeated, respectively, 6.3,

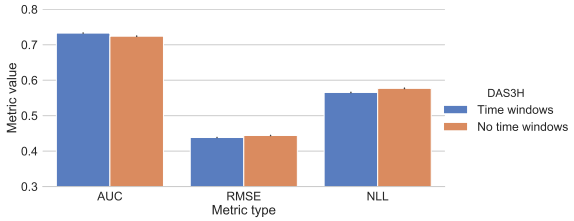

Figure 1: Comparison of the metrics between DAS3H and an alternative version in which time window features are replaced by simple counters of the past interactions and outcomes. For AUC, higher is better; for RMSE and NLL, lower is better.

37.8 and 19.6 times. Even though the number of KC repetitions is slightly smaller in ASSISTments 2012-2013, it is much higher in the two other datasets.

We hypothesize that the overperformance of DASH and the underperformance of DAS3H come from these differences: in EdNet, items and KCs are respectively more and less repeated. This difference makes it more difficult (respectively, easier) for DAS3H (respectively, DASH) to track a student's knowledge states. Moreover, attempts of a student on the same item are probably a stronger signal of future correctness on this item than repeated interactions with a set of KCs.

Is the finer past study representation responsible for the higher performance of DAS3H?

In (Choffin et al. 2019), we performed some ablation studies to see if the temporal module h_θ in DAS3H played a critical role in DAS3H's predictive power. We compared DAS3H to an alternative version in which its temporal module h_θ was replaced by simple win and fail counters. We performed the same experiments on EdNet, but we replaced h_θ by

$$\sum_{k \in \text{KC}(j)} \gamma_k c_{s,k} + \rho_k a_{s,k}$$

instead, since win, fail and attempt counters are collinear.

Results of this experiment are reported in Figure 1, which compares the performance metrics of DAS3H (in blue) and this ablated version of DAS3H (in orange). We see that DAS3H is systematically but only marginally better than the other model. These results are partially consistent with (Choffin et al. 2019): we previously reported that this difference was larger.

To help explain this difference, we plot in Figure 2 the total study durations of all Santa users. Up to 300,000 users spend only a couple of days on the Santa platform. In this situation, the last (and largest) time windows of DAS3H (7 days, 30 days and $+\infty$) are rarely of any use, which could be the reason why simple counters perform almost as well as the more complex DAS3H time module h_θ.

Also, we noticed in the dataset that a significant part of the interactions of any student occur at the same timestamp. Indeed, 17.6% of the interactions in the dataset occur at the

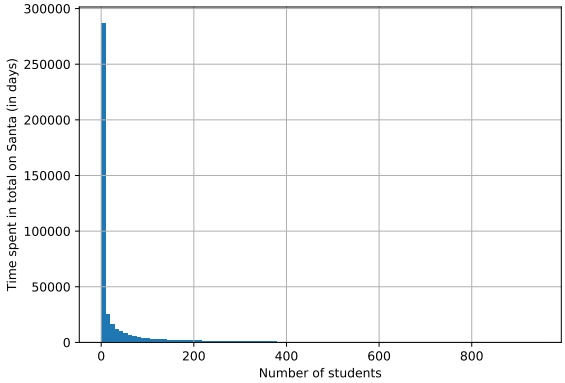

Figure 2: Histogram of the total study durations of each user on Santa

same timestamp as another interaction from the same student. On e-learning platforms, data are often collected by batches: when a user completes a series of items, the systems stores this sequence under the same timestamp, the final one. For this reason, fine-grained time windows such as the first one in DAS3H (1 hour) cannot distinguish between two temporally close interactions with the same KC and fail to correctly model memory decay at a short time scale.

Do different KCs have different learning and forgetting curves?

One of the assumptions of DAS3H is that KCs are learnt and forgotten at different rates: this is formalized by the fact that DAS3H estimates for each KC k a different set of $\theta_{w,k}$ parameters. To check if this assumption was relevant, we compared DAS3H to an alternative version of it that estimates the *same* set of θ_w parameters for all KCs. h_θ is thus replaced by:

$$h_\theta(\mathrm{t}_{s,j,1:\ell}, \mathrm{y}_{s,j,1:\ell-1}) = \sum_{k \in \text{KC}(j)} \sum_{w=0}^{W-1} \theta_{2w+1} \log(1 + c_{s,k,w}) - \theta_{2w+2} \log(1 + a_{s,k,w}).$$

Notice here that the dependency of the θ parameters on k has been removed. We coined DAS3H$_{1p}$ this alternative model.

We performed the same experiments on EdNet and we reported our results in Table 4.

Here the results are consistent with our previous results (Choffin et al. 2019): DAS3H outperforms DAS3H$_{1p}$, strengthening our initial assumption that different KCs should me modeled by different learning and forgetting curves.

How does DAS3H perform compared to deep learning methods?

Massive dataset, short sequences Gervet et al. (2020) indicate that for massive datasets (several millions of samples), DKT may be more suited than logistic regression-based approaches. Especially in settings where the learners

Model	AUC ↑	RMSE ↓	NLL ↓
DAS3H	**0.733**	**0.438**	**0.565**
DAS3H_{1p}	0.724	0.444	0.577

Table 4: Performance comparison between DAS3H and DAS3H_{1p}, an alternative version of DAS3H for which the influence of past practice is identical for all KCs. Metrics are averaged over the 5 folds. Standard deviations over the folds were all smaller than 0.001. ↑ and ↓ respectively indicate that higher (lower) is better.

progress sequentially through the material, DKT can distinguish the order in which items are solved while count-based approaches cannot. On the other hand, DKT fails to retain long-term information (when the sequences are long like in Bridge to Algebra 2006-2007 or Algebra 2005-2006, more than 1000 interactions in average per user) while the counts do not forget. When the sequences are short, attention-based approaches like SAKT (Pandey and Karypis 2019), AKT (Ghosh, Heffernan, and Lan 2020) or RKT (Pandey and Srivastava 2020) could be particularly appropriate, as we can see on the EdNet leaderboard.

Conclusion

In this article, we compared five student knowledge tracing models from the educational data mining literature on a massive educational dataset, EdNet. These five models were previously compared (Choffin et al. 2019) and one of our goals was to try to reproduce these results on this new dataset. Such a massive dataset generated several technical difficulties, that we did not encounter for previous datasets.

As further work, we would like to improve the scalability of our DAS3H model. This was the most computationally expensive model that we compared, both during the encoding phase and during training. As a result, we could not perform any hyperparameter search, notably regarding the size of time windows. Doing so would probably improve the performance of each of our models. Another approach could be to design models that do not rely on specifically designed time windows, using for example self-exciting processes such as Hawkes processes (Yao, Sahebi, and Feyzi-Behnagh 2020; Yao et al. 2021). We leave it for future work.

Acknowledgments

This work was funded by Caisse des Depôts et Consignations, e-FRAN program.

References

Anderson, J. R.; Matessa, M.; and Lebiere, C. 1997. ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction* 12(4): 439–462.

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.

Choffin, B.; Popineau, F.; Bourda, Y.; and Vie, J.-J. 2019. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills. In *Proceedings of the Twelfth International Conference on Educational Data Mining (EDM 2019)*, 29–38.

Choi, Y.; Lee, Y.; Shin, D.; Cho, J.; Park, S.; Lee, S.; Baek, J.; Bae, C.; Kim, B.; and Heo, J. 2020. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, 69–73. Springer.

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.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, 1646–1654.

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.

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.

Gonzalez-Brenes, J.; and Mostow, J. 2013. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In *Educational Data Mining 2013*.

Khajah, M.; Lindsey, R. V.; and Mozer, M. C. 2016. How deep is knowledge tracing? *Proceedings of the Ninth International Conference on Educational Data Mining* .

Lee, M.-C.; Chiang, W.-L.; and Lin, C.-J. 2015. Fast matrix-vector multiplications for large-scale logistic regression on shared-memory systems. In *2015 IEEE International Conference on Data Mining*, 835–840. IEEE.

Lindsey, R. V.; Shroyer, J. D.; Pashler, H.; and Mozer, M. C. 2014. Improving students' long-term knowledge retention through personalized review. *Psychological science* 25(3): 639–647.

Nagatani, K.; Zhang, Q.; Sato, M.; Chen, Y.-Y.; Chen, F.; and Ohkuma, T. 2019. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In *The World Wide Web Conference*, 3101–3107.

Pandey, S.; and Karypis, G. 2019. A Self Attentive model for Knowledge Tracing. In Desmarais, M. C.; Lynch, C. F.; Merceron, A.; and Nkambou, R., eds., *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*, 384–389.

Pandey, S.; and Srivastava, J. 2020. RKT: Relation-Aware Self-Attention for Knowledge Tracing. In d'Aquin, M.; Dietze, S.; Hauff, C.; Curry, E.; and Cudré-Mauroux, P., eds., *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*, 1205–1214. ACM.

Pashler, H.; Cepeda, N.; Lindsey, R. V.; Vul, E.; and Mozer, M. C. 2009. Predicting the optimal spacing of study: A multiscale context model of memory. In *Advances in neural information processing systems*, 1321–1329.

Pavlik, P. I.; Cen, H.; and Koedinger, K. R. 2009. Performance Factors Analysis - A New Alternative to Knowledge Tracing. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education, AIED 2009*, 531–538.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12(Oct): 2825–2830.

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.

Rasch, G. 1961. On General Laws and the Meaning of Measurement in Psychology. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 4: Contributions to Biology and Problems of Medicine*, 321–333. Berkeley, Calif.: University of California Press. URL https://projecteuclid.org/euclid.bsmsp/1200512895.

Stamper, J.; Niculescu-Mizil, A.; Ritter, S.; Gordon, G.; and Koedinger, K. 2010. Algebra I 2005-2006 and Bridge to Algebra 2006-2007. Development data sets from KDD Cup 2010 Educational Data Mining Challenge. Find them at http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp.

Tatsuoka, K. K. 1983. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of educational measurement* 20(4): 345–354.

Vie, J.-J.; and Kashima, H. 2019. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 750–757.

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 *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016*, 539–544.

Yao, M.; Sahebi, S.; and Feyzi-Behnagh, R. 2020. Analyzing Student Procrastination in MOOCs: A Multivariate Hawkes Approach. In Rafferty, A. N.; Whitehill, J.; Romero, C.; and Cavalli-Sforza, V., eds., *Proceedings of the 13th International Conference on Educational Data Mining, EDM 2020, Fully virtual conference, July 10-13, 2020*. International Educational Data Mining Society.

Yao, M.; Zhao, S.; Sahebi, S.; and Feyzi-Behnagh, R. 2021. Relaxed Clustered Hawkes Process for Student Procrastination Modeling in MOOCs. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Fully virtual conference, February 2-9, 2021*, to appear.

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.