Does Time Matter? Modeling the Effect of Time in Bayesian Knowledge Tracing

YUMENG QIU, YINGMEI QI, HANYUAN LU, ZACHARY A. PARDOS, NEIL T. HEFFERNAN
Worcester Polytechnic Institute, USA

Intelligent tutoring systems that utilize Bayesian Knowledge Tracing (KT) have the ability to predict student performance well. However, models currently in use do not consider that a student performing on ITS may not be finishing their work in the same day. We looked at KT's predictions on student responses where a day or more had elapsed since the previous response and found that KT consistently over predicted these data points in particular. We made two hypotheses to explain the over prediction behavior: 1) the student forgot since the last time on the tutor and 2) the student made a mistake (or slipped) on that first question of the day. We developed two models; KT-Forget and KT-Slip, modifications on Knowledge Tracing, to represent these two hypotheses. We evaluated and compared the performance of the KT-slip, KT-Forget and regular KT model by calculating prediction residuals and Area under Curve (AUC) on a Cognitive Tutor and ASSISTments dataset. The results showed that a significant improvement was obtained on the overall prediction by our KT-Forget model, suggesting that forgetting is the more likely cognitive explanation for the data and that there is a place for modeling forgetting, something that has not common practice in student modeling.

Key Words and Phrases: Bayesian Network, Knowledge Tracing, Intelligent Tutoring Systems, Data Mining, Model Evaluation

1. INTRODUCTION

The knowledge tracing model [Corbett & Anderson 1995] has been widely used to model student knowledge and learning over time. It assumes that each skill has two knowledge parameters, *prior* and *learn*; and two performance parameters, *slip* and *guess*. The *learn* parameter represents the probability that a student will transition between the unlearned and the learned state after each question. The *slip* parameter is the probability that a student who understands a skill can make a careless mistake and the *guess* parameter is the probability a student may answer correctly in spite of not knowing the skill. There is also a *forget* parameter; however, in standard knowledge tracing this is fixed at 0, which means that there is no forgetting happen in this model.

When using the standard Knowledge Tracing (KT) model, it is assumed that the students' probability of making the transition from the unlearned to the learned state is constant opportunities (or questions). Many researchers have proposed extensions to Bayesian Knowledge Tracing [Conati, Abigail, Gertner, VanLehn and Druzdzel 1997; Reye 2004], however none have tried to incorporate how much time has elapsed between opportunities into the model. They all assume that student performance a minute later is the same as the next day. Nonetheless, ever since Ebbinghaus inaugurated the scientific study of memory [Ebbinghaus 1913], researchers have examined the manner in which memory performance declines with time or intervening events [Pavlik & Anderson 2005].

In the real world coming into class on a new day may result in a student forgetting the material or a higher probability of them slipping. By taking this real world fact into consideration, in this paper we look into how KT performs on each new day's responses. We define a new day's response as a response that occurred on a later calendar date than the student's previous response to a question of the same skill. We found that KT's new day error is far higher than same day error. A residual analysis showed that KT was largely over predicting student performance on each new day response.

Based on the residual result (Table I), we made two hypotheses to explain this phenomenon; 1) that students may forget between days and 2) that students may slip when answering the first question on a new day. The slip hypothesis only affects the model's prediction of new day events while the forget hypothesis could affect prediction of subsequent responses since it hypothesizes a change in the latent of knowledge. We developed two new models based on Knowledge Tracing: a KT-Forget Model and a KT-Slip Model, where a new day variable is taken into account to affect either students' knowledge or performance. To implement this, we introduced a new split-parameter KT model, which allowed us to, for instance, learn a different *forget* parameter for new day opportunities than for same day but learn only a single learn rate parameter for each.

Table I. Knowledge Tracing residual analysis

Problem Set	Residual Same Day	Residual New Day
1	0.039803	-0.363268
2	-0.026765	-0.110578
3	0.088299	-0.076079
4	-0.014643	-0.117302
5	-0.003538	-0.062383
6	0.018866	-0.160024
7	0.009965	-0.109267
8	-0.049156	-0.169034
9	0.023225	0.032221
10	-0.029405	-0.010356
11	0.013791	-0.275969
12	0.082811	-0.054692
Average	0.012771	-0.123060

2. TIME MODEL DESIGN

When using the Knowledge Tracing model, it is assumed that the student's probability of making the transition from the unlearned to the learned state is not changing across opportunities, while in the real world students may forget the previously learned knowledge when coming into class on a new day. This fact assumes that there is a great possibility that a student's forgetting rate is not zero. The standard KT model assumes no probability of forgetting. Prior work has modeled forgetting between sessions in a lab but did not allow within-day learning to occur [Pardos, Heffernan, Ruiz & Beck 2008]. Alternatively, poor performance on a new day may also suggest that students may not actually be "forgetting" but instead, they might just be "slipping." We used Bayesian networks and Expectation Maximization to detect whether time had any influence on the *forget* parameter and the *slip* parameter of the KT model. The model with the better predictive accuracy will indicate the better cognitive explanation of the data.

2.1 Split-KT Model Design

In order to determine the validity of this method, we represent the above two hypothesis in the Bayesian Knowledge Tracing model by introducing a novel modification to the model that allows us to fit a same day and new day parameter for one parameter in a conditional probability table (CPT) while keeping the other parameter in the CPT constant. In Knowledge tracing; *learn* and forget share a CPT and guess and *slip* share a CPT. As shown below, the difference between split-KT and the original-KT is the ability

Authors' addresses: Department of Computer Science, Worcester Polytechnic Institute, USA. E-mail: ymqiu@wpi.edu, Yingmei.qi@wpi.edu, hylu_cs@wpi.edu,zpardos@wpi.edu, nth@wpi.edu

to separate the *forget*, *learn*, *guess*, and *slip* parameters individually. The equivalence between these two KT models was confirmed empirically by learning parameters for each model from a shared dataset, without new day data, and confirming that the learned parameters and predictions were the same.

The individualization of the four parameters were achieved by adding a forget node and a learn node to the knowledge node, as well as adding a guess node and slip node to the question node. Therefore, the knowledge nodes and question nodes are conditioned upon the four new nodes. The CPT for knowledge node is given in Table II. The CPT for the question node is also of this form, the only difference is changing the *learn* and *forget* parameters to *guess* and *slip* parameters and changing the previous and current knowledge to previous and current student performance. The question and knowledge CPTs are fixed and essentially serve as logic gates. The guess, slip, learn and forget node CPTs contain the continuous probabilities that are familiar to the standard KT model. Take the first row as an example, knowing that the students do not have previous knowledge of the skill (Knowledge_previous=F), and they neither learn nor forget (learn=F, forget=F), then we can infer the probability that students have the current knowledge is 0 (P(Knowledge_current=T) = 0).

Table II. The CPT for Knowledge node

Learn	**Forget**	**Knowledge_previous**	**P(Knowledge_current=T)**
F	F	F	0
T	F	F	1
F	T	F	0
T	T	F	1
F	F	T	1
T	F	T	1
F	T	T	0
T	T	T	0

This model can easily let us set individualized learn rates, forget rates, guess rates and slip rates. By this way we are able to fix the learn parameter and guess parameter in order to investigate how new day instances would affect the *forget* and *slip* parameters.

2.2 The KT-Forget

In this section we focus on one of the hypothesis: how would the new day instance affect the forget parameter. We think that it is highly possible that students could be forgetting the previously learned knowledge when there are several days interval between the practices on the ITS.

The model we used to test our hypotheses is a new model built based on the Split-KT model discussed in the previous section. By adding a time node to the Split-KT model we are able to easily specify which parameters of the model should be affected by a new day. The new day node is fixed with a prior probability of 0.2, which is the overall proportion of the new day instances in the dataset. The topology of the KT-Forget model is shown in Fig. 1. The forget node is only conditioned on the added new time node, so there is only one new parameter "*forget_n*" introduced in this KT-Forget model and represents the forget rate on a new day. We use "*forget_s*" to denote the forget rate on a same day, which we set to be 0 just as the *forget* parameter in the original Knowledge Tracing model implying that there is no forgetting between opportunities in the same day.

Table III. CPT of the forget node

New Day	P(Forget=T)
F	0
T	forget_n

The CPT for the forget node in this model is shown in Table III. This table says that when a new day response occurs, New Day=T, the probability that student forget knowledge is *forget_n*, P(Forget=T|New Day=T) and is 0, otherwise.

2.3 KT-Slip model

An alternate hypothesis is that while students might be performing on the ITS across several days, they are not forgetting the previously learned material. Rather, the students are just making a mistake on the first question of the day (rustiness effect) after which they no longer slip at a higher than usual rate. So the low accuracy on first attempt on a new day might not be captured in the *forget* parameter, it could be that they just slipped and answered wrong. This explanation makes it quite necessary for us to look into the *slip* parameter.

The KT-Slip model is similar to the KT-Forget model and can be represented simply by connecting the time node to the slip node instead of connecting to the forget node as in the Forget model. The Slip model allows us to model the different slip rates of the new days and the same days. The Slip model is shown above in Fig. 1 in the bottom box.

Table IV. CPT of the slip node

New Day	P(slip=T)
F	slip_s
T	slip_n

Since the slip node is only conditioned on the added new time node, there is also one new parameter *slip_n* introduced in this KT-slip model, which represents the slip rate on a new day, and the original *slip* parameter is denoted as *slip_s* here, which is shown in Table IV. This table says that when a new day response occurs, New Day=T, the probability of slipping is *slip_n*, P(slip=T|New Day=T) and is *slip_s*, otherwise.

2.4 Topology of the models

The Split-KT model's topology is shown together with KT-Slip and KT-Forget in Fig. 1. Boxes in the figure denote the portions of the figure that are used in each model. While all models are shown in this figure so the relationship between them can be seen, when the models are run they are run separately as a separate topology and not one big model.

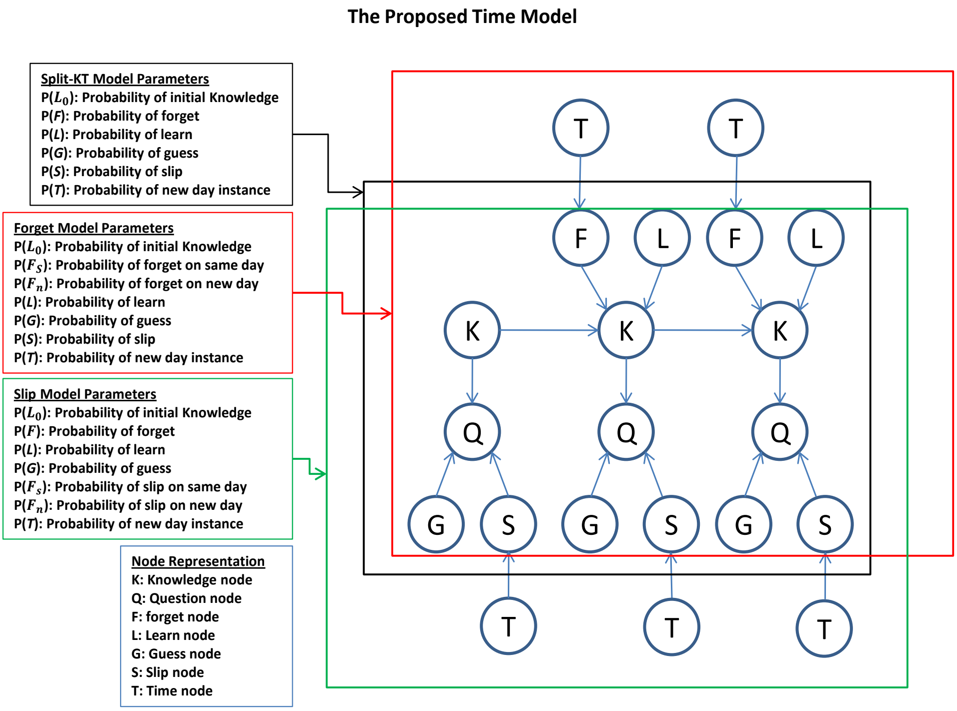

The Proposed Time Model
Split-KT Model Parameters
P(L_0): Probability of initial Knowledge
P(F): Probability of forget
P(L): Probability of learn
P(G): Probability of guess
P(S): Probability of slip
P(T): Probability of new day instance
Forget Model Parameters
P(L_0): Probability of initial Knowledge
P(F_s): Probability of forget on same day
P(F_n): Probability of forget on new day
P(L): Probability of learn
P(G): Probability of guess
P(S): Probability of slip
P(T): Probability of new day instance
Slip Model Parameters
P(L_0): Probability of initial Knowledge
P(F): Probability of forget
P(L): Probability of learn
P(G): Probability of guess
P(F_s): Probability of slip on same day
P(F_n): Probability of slip on new day
P(T): Probability of new day instance
Node Representation
K: Knowledge node
Q: Question node
F: forget node
L: Learn node
G: Guess node
S: Slip node
T: Time node
T
T
F
L
F
L
K
K
K
Q
Q
Q
G
S
G
S
G
S
T
T
T

Fig. 1. The topology of the models – Split-KT, KT-Forget, KT-Slip

2.4 Methodology

The analysis method consisted of two steps: run Expectation Maximization to fit the parameter on the training set for each model, and apply the trained parameters to the test sets to predict the student performance of each question.

The motivation behind this method is to compare the overall performance of each model, including the original KT model, the KT-Forget model, and the KT-Slip model. We trained the proposed model on datasets that are collected from real-world Intelligent Tutoring System – the Cognitive Tutor, and for further reference of the models' results we also apply our models to the datasets that are collected from the ASSISTments Platform. We evaluated and compared the accuracy of the KT-slip, KT-forget and regular KT model by calculating Residual and Area Under Curve (AUC). Residual is the mean of the actual performance subtracted by the predicted performance. AUC is a robust accuracy measure where a score of 0.50 represents a model that is only as good as chance and 1.0 represents a perfectly predicting model.

3 MODEL PERFORMANCE EVALUATIONS

To evaluate the performance of the KT-Forget and the KT-Slip models, we used a Cognitive Tutor dataset and ASSISTments dataset to test the real world utility of these models by comparing their predictive performance with a standard KT model. For each problem set, which represents a certain skill, we trained regular KT, KT-Forget and KT-Slip models to make predictions on all the question responses of each student. Then the Residuals and AUC is calculated for predictions and actual responses on same day events, new day events as well as overall events to analyze the three models' performance.

3.1 Datasets for Prediction

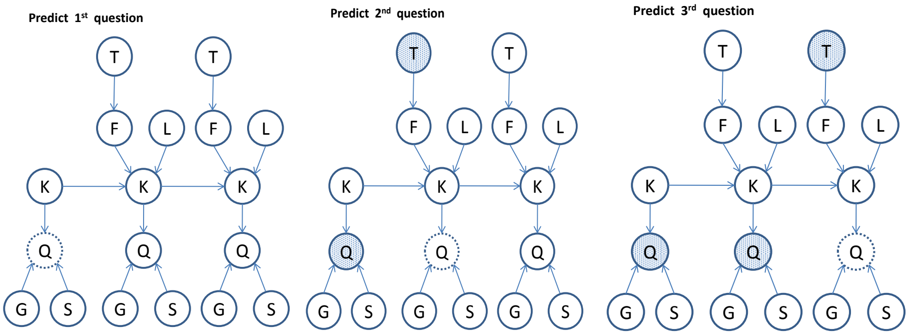
One of the datasets comes from the Cognitive Tutor System called Bridge to Algebra and is from the 2006-2007 school year. This was one of the smaller, development datasets made public as part of the 2010 Knowledge Discover and Data mining competition [Pardos & Heffernan, In Press]. In this tutor, students answer algebra problems from their math curriculum which is split into sections. The problems consist of many steps that the students must answer to go to the next problem. A student no longer needs to answer steps of a given skill when the Cognitive Tutor's Knowledge Tracing model believes the student knows the skill with probability 0.95 or greater. When a student has mastered all the skills in their current section they are allowed to move on to the next. The time for students using this system is determined by teachers. Twelve skills were chosen at random from this dataset for analysis (excluding skills such as "press enter" which do not represent math skills). There were an average of 122 student per skill in this dataset.

Another dataset is collected from ASSISTments Platform's Skill Builder problem sets. The ASSISTments Platform is an educational research platform better known for its e-learning [Feng, Heffernan, Mani and Heffernan, 2006] that provides web based math tutoring to 8th-10th grade students. Unlike the Cognitive Tutor System, students are forced to leave the tutor after 10 questions have been finished in one day and will come back to the tutor in a new day. If a student answers three questions correct in a row, they are "graduated" from the problem set. The help the tutorial provided is consist of a series of questions that brake a problem in to sub steps. A student can also request a hint, but requesting a hint will mark the student as getting the step wrong in the system. Only answers to the original questions are considered. The largest twelve Skill Builder datasets were selected from the ASSISTments Platform. There was an average of 1,200 students per problem set in this dataset. The highest student count problem sets were selected here because new day events are far more sparse in ASSISTments skill problem sets than the Cognitive Tutor skill problem sets.

The twelve datasets from each tutor were randomly divided into two equal parts by student, one part was used as the training set, the other as the testing set.

3.2 Prediction Procedure

Parameters were learned for each skill problem set individually. The parameters were unbounded and initial parameters were set to a *Guess* of 0.14, *Slip* of 0.09, *Prior* of 0.50 and *Learn* of 0.14, these initial values were the average parameter values across all skills in prior modeling work conducted on the ASSISTments tutor. For parameter learning, the new day observation (0 or 1) was presented as evidence in addition to the student responses. After training, the time and actual response values were given to the model as evidence for our new models to do the prediction (for regular KT, only actual responses were given as evidence) one student at a time. In order to predict every response of each student in the test set, the student data for prediction was presented to the network in the following fashion: for predicting the first question, no evidence was entered; for the second question, the new day information for that question and the actual response of first question were entered as evidence; for the third question, the first two new day information and responses information were entered as evidence. Apply this procedure until the prediction of the last question. This predicting process is shown in Fig. 2. By applying this prediction process, the probability of student answering each question correctly was computed and saved.



Dotted Outline Node = the question to be predict   Shaded Node = the evidence for the prediction

Fig. 2. The process of entering evidence data.

Finally, we calculated the Residuals and AUC values between predictions and actual responses on same day events, new day events as well as overall events of the whole problem set. The results summary of all three models across problem sets as well as the results of pairwise t-test is shown in Table VII.

3.3 Prediction Result Analysis

The prediction performance of the three models were calculated in terms of Residuals and AUC values between predictions and actual responses on same day events, new day events as well as overall events of the whole problem set. The model with higher AUC values for a problem set was deemed to be the more accurate predictor of that problem set. In addition, a two-tailed paired t-test was calculated between KT and KT-Forget and KT and KT-Slip. We first applied this to the datasets collected from Cognitive Tutor. The specific results of each problem sets are shown below for regular KT (Table V) and KT-Forget (Table VI).

Table V. Residual and AUC results on Regular KT

Regular KT	**Residuals**			**AUC**		
Problem Set	**Overall**	**Same Day**	**New Day**	**Overall**	**Same Day**	**New Day**
1	-0.0263	0.0398	-0.3633	0.5952	0.6570	0.4972
2	-0.0390	-0.0268	-0.1106	0.7588	0.7434	0.8669
3	0.0623	0.0883	-0.0761	0.6496	0.6914	0.5656
4	-0.0272	-0.0146	-0.1173	0.7023	0.7324	0.6126
5	-0.0125	-0.0035	-0.0624	0.5822	0.5654	0.6728
6	0.0092	0.0189	-0.1600	0.7892	0.8171	0.6290
7	-0.0063	0.0100	-0.1093	0.6374	0.6446	0.6236
8	-0.0664	-0.0492	-0.1690	0.6936	0.7210	0.6003
9	0.0251	0.0232	0.0322	0.5384	0.5218	0.6278
10	-0.0267	-0.0294	-0.0104	0.6456	0.6204	0.7892
11	-0.0422	0.0138	-0.2760	0.4922	0.5176	0.5055
12	0.0483	0.0828	-0.0547	0.6149	0.6558	0.5129
Average	-0.0085	0.0128	-0.1231	0.6416	0.6573	0.6253

Table VI. Residual and AUC results on KT-Forget

KT-forget	**Residuals**			**AUC**		
Problem Set	**Overall**	**Same Day**	**New Day**	**Overall**	**Same Day**	**New Day**
1	-0.0121	0.0208	-0.1802	0.7765	0.7771	0.5238
2	-0.0103	-0.0037	-0.0484	0.7373	0.7183	0.8588
3	0.0755	0.0855	0.0223	0.7368	0.7497	0.5528
4	-0.0364	-0.0292	-0.0876	0.7262	0.7433	0.5938
5	-0.0045	-0.0022	-0.0174	0.6681	0.6080	0.7712
6	0.0095	0.0115	-0.0270	0.8331	0.8370	0.6399
7	0.0020	0.0116	-0.0587	0.6834	0.6857	0.6012
8	-0.0549	-0.0435	-0.1230	0.7209	0.7407	0.5805
9	0.0257	0.0165	0.0608	0.6070	0.6301	0.6768
10	-0.0162	-0.0246	0.0331	0.6115	0.6024	0.7746
11	-0.0414	-0.0118	-0.1645	0.6751	0.6376	0.6067
12	0.0445	0.0699	-0.0312	0.6278	0.6525	0.5133
Average	-0.0016	0.0084	-0.0518	0.7003	0.6985	0.6411

Table VII. Summary and T-test on Regular KT, KT-Forget and KT-Slip (Cognitive Tutor)

	Residuals (across problem sets)			**AUC (across problem sets)**		
Model	**Overall**	**Same Day**	**New Day**	**Overall**	**Same Day**	**New Day**
1. Regular KT	-0.0085	0.0128	-0.1231	0.6416	0.6573	0.6253
2. KT-forget	-0.0016	0.0084	-0.0518	0.7003	0.6985	0.6411
3. KT-slip	-0.0047	-0.0048	0.0017	0.6110	0.5917	0.5175
t-test (1,2)	0.0352	0.2697	0.0004	0.0129	0.0178	0.2445
t-test (1,3)	0.5149	0.0154	0.0017	0.1690	0.0017	0.0033

From the above results, generally, we can see that the new KT-Forget model performed better on both the residuals and AUC compared to the regular KT model. Inversely, the KT-Slip model performed worse than we expected. The specific evaluation of the two new models is shown in Table VII and Table VIII .

For the KT-Forget model, improved results were obtained both on residuals and AUC. Especially for the AUC, although KT-forget did not get significant improvement on new day events in terms of AUC (p value is 0.5175); however, it got significant improvement on same day events prediction and overall prediction (p value is 0.0178 and 0.0129), which means the performance of KT-Forget model is more accurate on predicting of Cognitive Tutor data compared to the regular KT model. Moreover, the better prediction performance also supported our hypothesis that students probably forget knowledge when it comes to a new day.

For the KT-Slip model, the results of overall data's AUC were worse but not significantly compared to regular KT. However, both same day and new day AUC were significantly worse, which overthrew our assumption that students may slip when it comes to a new day.

Similarly, we applied our models to the ASSISTments datasets. The results of residuals and AUC across all problem sets are as below:

Table VIII. T-test on Regular KT, KT-forget and KT-slip (ASSISTments)

	Residuals (across problem sets)			**AUC (across problem sets)**		
Model	**Overall**	**Same Day**	**New Day**	**Overall**	**Same Day**	**New Day**
1. Regular KT	0.0019	-0.0019	0.0241	0.6719	0.6704	0.6364
2. KT-forget	-0.0036	-0.0129	0.0488	0.6678	0.6672	0.6366
3. KT-slip	-0.0105	-0.0240	0.0628	0.6486	0.6520	0.5981
t-test (1,2)	0.1449	0.0099	0.0001	0.1640	0.0885	0.9603
t-test (1,3)	0.0133	0.0003	0.0057	0.0085	0.0353	0.0057

From Table VIII, we can observe that the new models, both KT-Forget and KT-Slip lost to the regular KT model, especially on the AUC. We looked into the reason why our new models perform much worse and found that the way the data was collected lead to this result. As we mentioned in the previous section, students are forced to leave the tutor after a certain number of questions have been finished in one day and will come back to the tutor in a new day. Thus, we observed that the datasets collected from ASSISTments have much fewer new day events (average 1 per student) and is not as amenable to a time analysis as the Cognitive Tutor data which has many new days per student and students experience the new day more naturally. Therefore, the results obtained from Cognitive Tutor are more practical for this analysis.

4 CONTRIBUTIONS

This paper makes two contributions. First, we show assumptions made in Knowledge Tracing model, that student don't forget, is false. While this might not be terribly surprising, we identify a particular situation in which the standard KT model has systematic errors in predicting student performance, which is on new day responses.

Secondly, we present a model to account for this phenomenon which does a reliably better job of fitting student data in some datasets. This is significant as KT has proved itself to be a very effective model, difficult to improve upon. It is also noteworthy that KT is easily interpretable and it is beneficial to be able to have a clean model that fits easily into the Bayesian framework and inherits this interpretability. Our contribution is that researchers should pay attention to "time" and we have demonstrated a method that takes this into account and improves modeling performance.

5 DISCUSSIONS AND FUTURE WORK

In this work we attempt to model the time factor to better predict students' learning performance in intelligent tutoring systems. Due to our experiment results that new KT-forget model worked very well on Cognitive Tutor datasets while failed on ASSISTments Tutor datasets, we need to further investigate into the real reasons which caused this. Thus, we would like to know when using KT-forget model is not beneficial.

In this paper we only made two assumptions that the parameters "forget" and "slip" will be affected by time factor. We have not yet looked into the performance of other parameters that might be affected by time, for example: students may have a fresh mind and learn more on a new day, which means a new parameter "learn new day" should be modeled. Also, it is possible that "time" should connect to these two parameters at once. We will keep on delving into these possibilities to see whether further improvement incorporating time can be obtained. If this is achieved in future, we can build an ensemble model [Caruana, Niculescu-Mizil, Crew and Ksikes 2004] that combines regular KT's results on same day with the new model's results on new day.

Our work only focuses on whether students answer the questions in one day or in a new day, we do not pay attention to the intervals between same day and a new day. Pavlik and Anderson's [Pavlik & Anderson 2005] study showed that longer intervals should have a greater impact more on students' performance while shorter intervals may have very little effect on actual responses. These topics deserve further investigation to figure out how to leverage the valuable time information and build better user models.

ACKNOWLEDGEMENTS

This research was supported by the National Science foundation via grant "Graduates in K-12 Education" (GK-12) Fellowship, award number DGE0742503 and Neil Heffernan's CAREER grant. We would like to thank the Pittsburg Science of Learning Center for the Cognitive Tutor datasets and Matthew Dailey for his data preparation assistance.

REFERENCES

Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. Ensemble selection from libraries of models. Proceedings of the 21st International Conference on Machine Learning, (2004)

Conati, C., Abigail S. Gertner, VanLehn, K. , and Druzdzel, M.. On-Line Student Modeling for Coached Problem Solving Using Bayesian Networks (1997)

Corbett and Anderson 1995. A.T. Corbett and J.R. Anderson, Knowledge tracing: modeling the acquisition of procedural knowledge, pp. 253–278. User Modeling and User-Adapted Interaction 4, (1995)

Corbett, A.T, 2001, Cognitive computer tutors: Solving the two-sigma problem, pp 137-147. User Modeling: Proceedings of the Eighth International Conference, Sonthofen, Germany, UM (2001)

Ebbinghaus, H. Memory: A Contribution to Experimental Psychology, translated in English, (1913)

Feng, M., Heffernan, N. T., Mani, M., & Heffernan, C. Using Mixed-Effects Modeling to Compare Different Grain-Sized Skill Models. In Beck, J., Aimeur, E., & Barnes, T. (Eds). Educational Data Mining: Papers from the AAAI Workshop. Menlo Park, CA: AAAI Press. pp. 57-66. Technical Report WS-06-05. ISBN 978-1-57735-287-7, (2006)

Pardos, Z. A., Heffernan, N. T., Ruiz, C. & Beck, J. Effective Skill Assessment Using Expectation Maximization in a Multi Network Temporal Bayesian Network. The Young Researchers Track at the 20th International Conference on Intelligent Tutoring Systems. Montreal, Canada, In press (2008)

Pardos, Z.A., Heffernan, N. T.: Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset. To appear in Journal of Machine Learning Research W & CP, (In Press)

Pavlik, P. Jr., Anderson,J. R., Practice and Forgetting Effects on Vocabulary Memory: An Activation-Based Model of the Spacing Effect, pp. 559–586. Cognitive Science 29 (2005)

Reye, J.: Student modeling based on belief networks. International Journal of Artificial Intelligence in Education 14, 1-33. (2004)