In Proceedings of the 1st Workshop on Massive Open Online Courses at the 16th Annual Conference on Artificial Intelligence in Education (2013). Memphis, TN. http://www.moocshop.org

Revisiting and Extending the Item Difficulty Effect Model

Sarah Schultz and Trenton Tabor
Worcester Polytechnic Institute
100 Institute Rd, Worcester, MA
{seschultz, tstabor}@wpi.edu

Abstract: Data collected by learning environments and online courses contains many potentially useful features, but traditionally many of these are ignored when modeling students. One feature that could use further examination is item difficulty. In their KT-IDEM model, Pardos and Heffernan proposed the use of question templates to differentiate guess and slip rates in knowledge tracing based on the difficulty of the template- here, we examine extensions and variations of that model. We propose two new models that differentiate based on template- one in which the learn rate is differentiated and another in which learn, guess, and slip parameters all depend on template. We compare these two new models to knowledge tracing and KT-IDEM. We also propose a generalization of IDEM in which, rather than individual templates, we differentiate between multiple choice and short answer questions and compare this model to traditional knowledge tracing and IDEM. We test these models using data from ASSISTments, an open online learning environment used in many middle and high school classrooms throughout the United States.

Keywords: Knowledge tracing, student modeling, item difficulty, Bayesian networks, educational data mining

1. Introduction

Traditionally, knowledge tracing (KT), does not take into account much of the data collected by tutoring system. Some work has been done on leveraging hint and attempt counts in KT [8], [9], and in individualizing based on student [6], but one area that merits more exploration is the use of item difficulty to more accurately model students. Pardos and Heffernan proposed a model to do just that [5], but explored only one such possible model. We created two variations on this model and a generalization of it in order to determine which of these models is the best predictor of student knowledge. Our goal is to discover how item difficulty really affects students' knowledge and performance.

2. Models

2.1 Knowledge Tracing

In classic knowledge tracing [1], the goal is to predict whether a student will answer the next question correctly based upon the current estimate of their knowledge. In the Bayesian network, the responses are the observed nodes, and the student's knowledge at each time-step are the latent nodes. Using Expectation Maximization (EM) or another

algorithm, we learn values for the probability of initial knowledge, $P(L_0)$; the probability of learning the skill from one time step to the next, P(T); the probability of guessing correctly when the skill is in the unlearned state, P(G); and the probability of slipping, or answering incorrectly when the skill is in the learned state, P(S) (Figure 1).

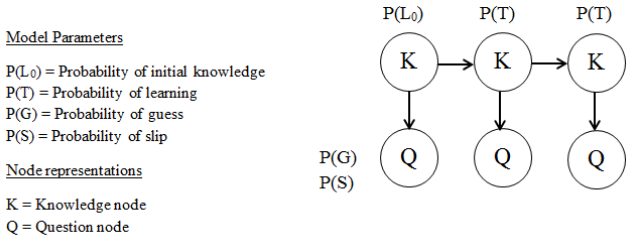

Fig. 1- Standard Knowledge Tracing

2.2 KT-IDEM

In 2011, Pardos and Heffernan proposed the Knowledge Tracing- Item Difficulty Effect Model (KT-IDEM), which adds difficulty to the traditional KT model by adding an item difficulty node affecting the question node. This model learns a separate guess and slip rate for each item, and therefore has N*2+2 parameters, where N is the number of unique items, in comparison to KT's four [5]. Figure 2 illustrates the KT-IDEM model.

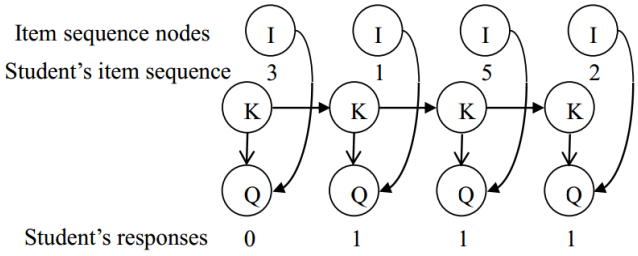

Fig. 2- Knowledge Tracing- Item Difficulty Effect Model

2.3 Extensions to IDEM

We believe that question difficulty not only affects performance, but will also have an effect on learning. By answering questions of different difficulties and receiving feedback on whether or not the answer is correct, students could learn differing amounts. We therefore propose two new variations on KT-IDEM. The first individualizes learn rates by item difficulty, but keeps guess and slip consistent. The second individualizes learn, guess, and slip rates based on item difficulty. In a ten item dataset, KT would have four

parameters, KT-IDEM would have 22, the first of our models, Item Difficulty Effect on Learning (IDEL), would have 12, and the second, Item Difficulty Effect All (IDEA), would have 32. It is possible that certain datasets will be over-parameterized in some of these models if there are not enough data points per item, but as Pardos and Heffernan pointed out in their original KT-IDEM paper, "there has been a trend of evidence that suggests models that have equal or even more parameters than data points can still be effective" [5]. These models are illustrated below (Figures 3 and 4).

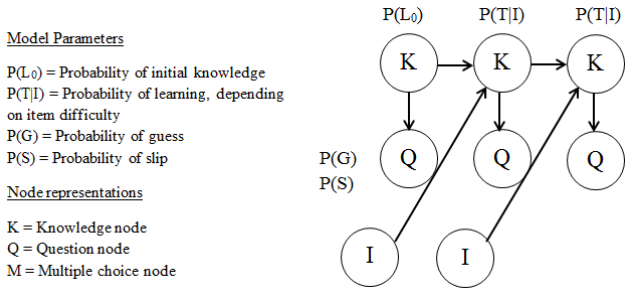

Fig. 3- Item Difficulty Effect on Learning

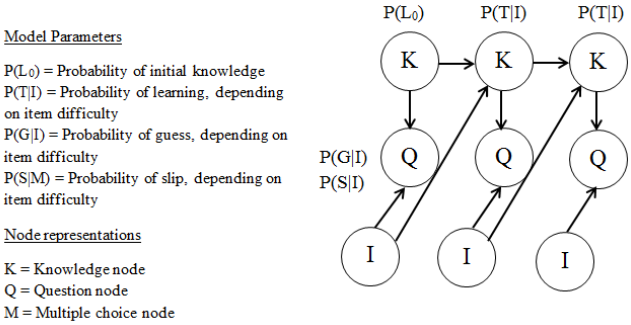

Fig. 4- Item Difficulty Effect All

2.4 MC

The final model we implemented is a generalization of KT-IDEM, which adds a multiple choice node to KT at each time step, indicating whether the particular question is multiple choice or not, rather than an item difficulty node. We now learn two different guess and slip rates, one each for multiple choice questions and for non-multiple choice questions. As is standard in KT and all other models explored in this paper, we assume that students do not forget. The multiple choice model (MCKT) is illustrated in Figure 5.

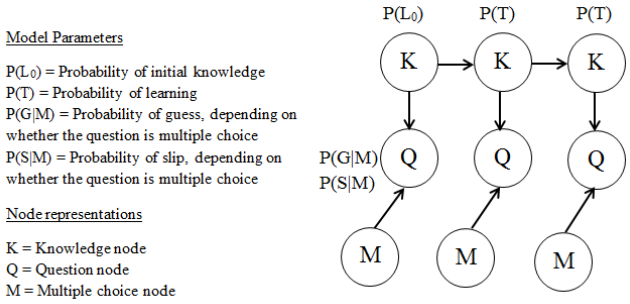

Fig. 5- Multiple Choice Model

We expected that the guess rate for multiple choice questions would be higher than the guess rate for non-multiple choice questions, since there are a finite number of options presented as opposed to an open response where it is possible to enter almost anything. We also expected that the slip rate would be lower for multiple choice questions, as recognizing the correct answer is generally easier than recalling it [3].

3. Dataset

3.1 The ASSISTments Tutoring System

The data used in this work is from ASSISTments, a freely available online mathematics tutoring system for grades 4 to 10 [2]. This system is used in classrooms across the country, and while it is not currently in itself a course, it is certainly an open, large-scale online learning tool.

In ASSISTments, multiple items can be built using the same template, where the only difference is the actual numbers in the problem. We consider problems generated from the same template to be the same item when working with the models that consider item difficulty.

We used six skills from the dataset, all of which came from skill builder data. In ASSISTments, skill builders are sessions where a student practices a certain skill until s/he gets three questions correct in a row, at which point it is considered to be learned. Within each skill, there are different sequences of templates that a student could encounter. In order to be sure that all students in our dataset were seeing the same templates, we used one sequence from each skill, except for Ordering Integers, from which we sampled two sequences separately. Table 1 shows information about the sequences we used in our experiments.

Table 1- Sequences used to test the models

Skill Name	Percent correct	Number of Templates	Percent Multiple Choice
Pythagorean Theorem	34	8	70
Ordering Integers (1)	88	3	34
Ordering Integers (2)	84	3	65
Square Root	89	2	38
Ordering Positive Decimals	74	3	100
Percent	33	13	67
Pattern Finding	48	5	45

4. Methods

Using Kevin Murphy's Bayes Net Toolbox for Matlab [4], we built each of our proposed models. We performed a 5-fold cross-validation on each of the seven sequences from the ASSISTments dataset using all five models, where four folds were used for training and the fifth for testing. The data was partitioned into folds randomly such that each student within a skill was in only one fold and the same folds were used for every model to guarantee a fair comparison. To avoid over-fitting the models to any student who practiced a skill a large number of times, only the first five opportunities of the skill for each student were used. We used expectation maximization to learn the parameters for each of our models.

5. Results

In order to compare models, we calculated mean absolute error (MAE), root mean square error (RMSE), and area under the curve (AUC) of each model's predictions compared to the actual data. We performed a paired t-test of each of these measures using the runs from each fold and found that RMSE was the most consistently reliable measure, so we use that to determine which model is best. Table 2 shows an example of all metrics, obtained from the skill "Percent," which has 13 templates. From this data, it appears that KT has the worst MAE and AUC of all the models, but KT-IDEL has a worse RMSE.

Table 2- Results for "Percent"

	Knowledge Tracing	KT-IDEM	KT-IDEL	KT-IDEA	MCKT
MAE	0.433231	0.350409	0.433039	0.352525	0.352107
AUC	0.531074	0.762205	0.56607	0.706951	0.754057
RMSE	0.472552	0.449915	0.481702	0.441461	0.462738

Comparing the template-based models to KT, we found that for this skill, the MAE was reliably better for KT-IDEM than KT or KT-IDEL and the AUC of KT-IDEM was reliably better than KT and both other template models. On the other hand, KT-IDEA had a reliably better RMSE than KT-IDEM for this skill.

Taking the data from all seven sequences, we unfortunately did not find a conclusive answer to the question of which template-based model performs best. For the skill "Pattern Finding," we found that KT-IDEM did best in all three measures, whereas for the first sequence of "Ordering Integers," KT-IDEL outperformed the other two template-based models, but was not significantly different from KT. (A few additional results tables can be found in the appendix of this paper.)

Our next question, was whether the multiple choice model would perform better than KT or KT-IDEM. While theoretically, the multiple choice model should be the same as KT when all problems are of one type, when we ran the models over a sequence that was all multiple choice, the models learned different parameters. This is probably because the multiple choice nodes must always have two values in their CPT tables. We therefore exclude this sequence from analysis of the multiple choice model. On the other hand, we did test a sequence that was all one template, and all template models behaved the same, since the number of values in the template nodes' CPT tables is the same as the number of templates. Out of the six remaining sequences in which we can compare MCKT, each with three metrics, for a total of 18 comparisons, we found that MCKT was reliably better than KT six times, and reliably better than KT-IDEM four times. Out of these, only two instances showed MCKT better than both of the other models. Out of the remaining nine comparisons, four showed that MCKT was better than the others, but not reliably so, in one case KT-IDEM outperforms MCKT, which is marginally better than KT, and in six cases the both of the other models performed better than MCKT. Since MCKT is at least marginally better than KT a majority of the time, and significantly better in 6 out of 18 cases, it looks like it could be a promising model, although more research is needed.

6. Contributions and Future Work

In this work, we proposed three new models; IDEL, IDEA, and MCKT. We compared these models to traditional KT and to KT-IDEM and found that different models worked best for different sequences. Our findings are not in agreement with [5], which states that IDEM works better than KT in ASSISTments skill builder data, and our observations also seem to indicate that other item difficulty models could work better than KT-IDEM. The interesting contribution here is that this means question difficulty does, in fact, appear to affect learning, possibly more than performance on the current item.

We used only six sequences (and had to exclude one from analysis), all from the same system, in this preliminary look at these models and would like to, in the future, try using more sequences and data from other tutors to see be sure that findings hold true in other scenarios and are not useful only in ASSISTments. Although, even if the latter is the case, having a better student modeling technique for this system would be very useful in developing ways to make it better.

One clear next step is to implement the same extensions made to the IDEM model to the multiple choice model in order to determine how the different types of questions- multiple choice and short answer- effect student knowledge and performance.

Acknowledgements

This work was supported by the WPI Computer Science department and Robotics Engineering program. We would like to thank all of the people associated with creating ASSISTments, as well as the schools and teachers using ASSISTments. We thank Professor Neil Heffernan for helping us to conceptualize this work and for his ideas and Yutao Wang for the code base for knowledge tracing. We would also like to thank Zach Pardos for his advice.

Appendix

Table 3- Results for “Pythagorean Theorem”

	KT	KT-IDEM	KT-IDEL	KT-IDEA	MCKT
MAE	0.480245	0.448852	0.478075	0.431558	0.472431
AUC	0.610767	0.630755	0.661355	0.671785	0.587751
RMSE	0.491635	0.517432	0.487239	0.511354	0.530694

Table 4- Results for "Ordering Positive Decimals" (MCKT excluded)

	KT	KT-IDEM	KT-IDEL	KT-IDEA
MAE	0.352754	0.434477	0.362968	0.451735
AUC	0.58984	0.549476	0.61913	0.577328
RMSE	0.422419	0.474215	0.418596	0.492843

Table 5- Results for "Ordering Positive Integers (1)"

	KT	KT-IDEM	KT-IDEL	KT-IDEA	MCKT
MAE	0.223823	0.268527	0.223668	0.270949	0.2948
AUC	0.545965	0.351229	0.560837	0.36537	0.38731
RMSE	0.333427	0.365692	0.335122	0.394251	0.3903

References

1. Corbett, A.T. and Anderson, J.R., 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User Adapted Interaction, 4(4), pp.253–278.
2. Heffernan, N.T. ASSISTments. http://teacherwiki.assistment.org/wiki/About www.assistments.org
3. Moreno, R., 2010. Education Psychology, John Wiley & Sons, Inc.
4. Murphy, K. 2007. Bayes Net Toolbox for Matlab.
5. Pardos, Z.A. and Heffernan, N.T., 2011. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In J. A. Konstan, R. Conejo, J. L. Marzo, & N. Oliver, eds. User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, pp. 243–254.
6. Pardos, Z.A. and Heffernan, N.T., 2010. Modeling Individualization in a Bayesian Networks Implementation of Knowledge Tracing. User Modeling Adaptation and Personalization, In press, pp.255–266.
7. Qiu, Y., Qi, Y., Lu, H., Pardos, Z. and Heffernan, N., 2011. Does time matter modeling the effect of time in Bayesian knowledge tracing. In M. Pechenizkiy, T. Calders, C. Conati, S. Ventura, C. Romero, & J. Stamper, eds. Proceedings of the 4th International Conference on Educational Data Mining. pp. 139–148.
8. Wang, Y. and Heffernan, N.T., 2010. Leveraging First Response Time into the Knowledge Tracing Model.
9. Wang, Y. and Heffernan, N.T., 2011. The "Assistance" model: Leveraging how many hints and attempts a student needs. In 24th International Florida Artificial Intelligence Research Society FLAIRS 24 May 18 2011 May 20 2011. AAAI Press, pp. 549–554.