Extend the Knowledge Tracing Framework using Partial Credit as Performance

Y. WANG
Worcester Polytechnic Institute, USA
AND
N.T. HEFFERNAN
Worcester Polytechnic Institute, USA

In an ITS, students typically have two types of performance to a problem: correct and incorrect, and all the other information such as how many hints the student sees in this question and how many attempts he/she does to get the correct answer is ignored. Feng and Heffernen (2010) showed that we can predict better by accounting for problem solving behavior as well as correctness. By introduce continuous performance nodes into knowledge tracing model, we are able to represent partial credit – which is computed using all the information in a student’s response to a question. In this paper, we present the algorithm to compute partial credit, as well as the modified Knowledge Tracing (KT) model using partial credit as performance. We compared the two KT model with different types of performance node. The result shows that partial credit reliably improves the knowledge tracing model in predicting both the partial credit performance and the binary performance.

Key Words and Phrases: Knowledge Tracing, Educational Data Mining, Student Responses, Partial Credit

1. INTRODUCTION

In many dominated student models, such as the Knowledge Tracing model and the Performance Factor Analysis (Pavlik, Cen and Koedinger 2009), student performance are presented as a binary value of correct or incorrect. The amount of assistance a student needed to eventually get a problem correct is ignored in these models. Feng and Heffernen (2010) showed that we can predict student performance better by accounting for amount of assistance, but didn't reduce it to a value that could be shared with students. Arroyo, Cooper, etc.(2010) showed how to use this information to predict learning gains. Their work suggests that using hints and attempts to model student behavior online could be effective. In our previous work (Wang and Heffernan 2010), we present a naïve algorithm to assign partial credit given detailed student responses, and used two evaluations ways to confirm that the partial credit helps to improve student model fitting in compare to the traditional binary performance (correct/incorrect). In this paper we want to see if we can improve one of the dominant methods of student modeling: the Knowledge Tracing model by replacing the discrete performance node with continuous partial credit node.

In the rest of this section, we describe the tutoring system and dataset used in our experiments. Section 2 contains a introduction to the partial credit algorithm. Section 3 introduces the Knowledge Tracing model with continuous performance node. Experimental results are shown in Section 4. In Section 5 and 6 we discuss our

Authors’ addresses: Y. Wang, Department of Computer Science, Worcester Polytechnic Institute, MA, USA. E-mail: yutaowang@wpi.edu; N.T. Heffernan, Department of Computer Science, Worcester Polytechnic Institute, MA, USA; E-mail: nth@wpi.edu
conclusions and future directions for our work.

1.1 The Tutoring System and Dataset

Our dataset consisted of student responses from The ASSISTment System, a web based math tutoring system for 7th-12th grade students that provides preparation for the state standardized test by using released math items from previous tests as questions on the system. The tutorial helps the student learn the required knowledge by breaking the problem into sub questions called scaffolding or giving the student hints on how to solve the question. Fig. 1 shows an example of a hint, which is one type of assistance. A second type of assistance is presented if they click on (or type in) an incorrect answer, at which point the student is given feedback that they answered incorrectly (sometimes, but by no means always, students will get a context-sensitive message we call a “buggy message”).

![Fig. 1. Assistance in ASSISTment](image)

The data we analyzed consisted of 52,529 log records during the period Jan 2009-Feb 2009. These data was from 72 twelve- through fourteen-year old 8th grade students in urban school districts of the Northeast United States and 106 skills (e.g. area of polygon, Venn diagram, division, etc). In this study, for each log record we use the matrix: correctness of the first attempt, total attempt number, correct answer number, total answer number, required hint number, total hint number, and the relationship of main question and scaffolding question to compute partial credit.

2. PARTIAL CREDIT

We assign partial credit to each problem a student answers. As in binary performance
model, a student would get a ‘1’ if the first attempt is correct, otherwise, the credit would
ranges between ‘0’ and ‘1’ according to how many helps and attempts is needed.

Intuitively, the more hints are asked, the less chance the student understands the
knowledge, so we penalize for each hint asked by subtracting $1/\text{total hint}$ from the
credit. Also, more attempts indicates a lower possibility of understanding the required
skill, we penalize each attempt by a value which is corresponding to the time of trying. In
multiple choice problems, the value can be computed as $\text{correct_answer}/(\text{total_answer}-$
i), in which $i$ is the time of trying. This formula indicates a heavier penalize as the
attempt time increasing. This is because as the remaining choices in the question become
fewer, the difficulty of solving the problem decreased. In other types of problems, such as
fill in blank problem, since we are not sure the relationship of the time of trying and the
difficulty of solving the problem, the penalize for each attempt is set to a constant, which
is 0.1 in our experiments. After computing hint penalize ($\text{phint}$) for each hint and attempt
penalize ($\text{pattempt}$) for each attempt, we add them together to compute the total hint
penalize ($\text{total_phint}$) and the total attempt penalize ($\text{total_pattempt}$) for this problem.

Table I shows the whole algorithm of computing partial credit.

<table>
<thead>
<tr>
<th>Table I. Partial Credit Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>function $\text{pc} = \text{partial_credit}(\text{problem})$</td>
</tr>
<tr>
<td>if first attempt correct then</td>
</tr>
<tr>
<td>$\text{pc} = 1$</td>
</tr>
<tr>
<td>else if problem has no scaffold then</td>
</tr>
<tr>
<td>$\text{pc} = 1 - #\text{hint} \times \text{phint} - \text{total_pattempt}$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>for each scaffold question $i$ in the problem do</td>
</tr>
<tr>
<td>$\text{pc}_\text{scaffold}(i) = \text{partial_credit}(\text{scaffold}(i))$</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>$\text{pc} = 0.9 \times \text{sum}(\text{pc}_\text{scaffold}(i)) / #\text{scaffold}$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>if $\text{pc} &lt; 0$ then</td>
</tr>
<tr>
<td>$\text{pc} = 0$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>return $\text{pc}$</td>
</tr>
<tr>
<td>end function</td>
</tr>
</tbody>
</table>

When computing partial credit for a problem which has scaffold problems, the
penalizing for ask for scaffolding is set to 0.1. We use the same method described above
to compute the partial credit of each scaffold problem and assign the final partial credit as
the average partial credit of all the scaffold problems times a coefficient 0.9.

3. KNOWLEDGE TRACING WITH CONTINUES PERFORMANCE NODE
The Knowledge Tracing model shown in Fig.2 has been widely used in ITS to model
student knowledge and learning over time. It has become the dominant method of student
modeling, and many variants have been developed to improve its performance (Baker et al. 2010, Pardos and Heffernan 2010).

Knowledge Tracing uses one latent one observable dynamic Bayesian network to model student learning. As shown in Fig.2, 4 parameters are used for each skill, with two for student knowledge (initial knowledge and probability of learning the skill) and the other two for student performance (the probability of guessing correctly when the student doesn’t know the skill and the probability of slipping when the student does know the skill).

The structure of the Knowledge Tracing model with continuous performance node is the same as the original Knowledge Tracing model. The only differences between these two models are the “Guess” and “Slip” parameters. Since now the performance node is continuous, we assume it satisfies the Gaussian distribution. We use four parameters: guess_mu, guess_sigma, slip_mu, slip_sigma, to describe the two Gaussian distributions corresponding to the “Guess” and “Slip” situations.

In our experiment, we used the Bayes Net Toolbox for Matlab developed by Murphy (2001) to implement Knowledge Tracing, and the Expectation Maximization (EM) algorithm to fit the model to the dataset. The EM algorithm finds a set of parameters that maximize the likelihood of the data by iteratively running an expectation step to calculate expected likelihood given student performance data and a maximization step to compute the parameters that maximize that expected likelihood. We choose initial parameters empirically according to our experiment results. For each skill: initial knowledge = 0.5, learning = 0.1, guess_mu = 0.1, guess_sigma = 0.02, slip_mu = 0.1, slip_sigma = 0.02.

4. EVALUATIONS
To evaluate how well the new model fits the data, we used the Root Mean Squared Error (RMSE) to examine the predictive performance on a unseen test set. The lower the RMSE is, the better the prediction.

Table II shows the results of the comparison for the two different models: the original Knowledge Tracing model and the Knowledge Tracing with partial credit model. We compared the RMSE for predicting the partial credit performance and for predicting the traditional binary performance respectively. The Knowledge Tracing with partial credit model has lower RMSE value in both situations.
Table II. original KT (OKT) vs KT with partial credit (KTPC)

<table>
<thead>
<tr>
<th>Model</th>
<th>Predicting Performance</th>
<th>Partial Credit</th>
<th>Binary Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>OKT</td>
<td>0.4128</td>
<td>0.4637</td>
<td></td>
</tr>
<tr>
<td>KTPC</td>
<td>0.2824</td>
<td>0.4572</td>
<td></td>
</tr>
</tbody>
</table>

We determine whether the difference between these two models is statistically significant by computing the RMSE value for each student to account for the non-independence of their actions, then comparing these two models using a two tailed paired t-test. The $p$ value of the RMSE between using the original Knowledge Tracing and the Knowledge Tracing with partial credit model to predict the partial credit is 0. The $p$ value between using the original Knowledge Tracing and the Knowledge Tracing with partial credit model to predict the binary performance is 1.05E-49. Thus the Knowledge Tracing with partial credit model statistically reliably improves student performance prediction than the original Knowledge Tracing model.

5 CONCLUSIONS AND FUTURE WORK
In this paper we present an algorithm to assign partial credit given detailed student responses. Partial credit performance contains much more information than binary performance, which is currently used in almost all researchers in the educational data mining field. Experiments on the Knowledge Tracing model with partial credit as performance showed that partial credit can help improve the predicting of student performance in both partial credit and binary performance.

One question we are interested in is how to refine the algorithm to better fit student data and infer student knowledge. Also, we are interested in finding out in which situations partial credit does better than binary performance, so we can use it as an efficient complement to all the current models that use only the binary performance.

6 CONTRIBUTIONS
In this paper we present an understandable and easy to refined algorithm to assign partial credit according to student detailed responses. The idea of partial credit makes the performances contain much more information than the binary performances which is used by almost all the researchers in the educational data mining field. In this sense, a proper partial credit has great potential to indicate student knowledge better then binary performances.

We then extended the Knowledge Tracing frame work to include continuous performance node to combine with the partial credit. We analyzed the result and got statistical reliably improvement in predicting both students’ partial credit performance and binary performance. This result indicates that with small modification, it is highly possible that many current models that use only the binary student performance could be improved by using partial credit as a represent of student performance.

ACKNOWLEDGEMENTS
This research was made possible by the U.S. Department of Education, Institute of Education Science (IES) grants #R305K03140 and #R305A070440, the Office of Naval Research grant # N00014-03-1-0221, NSF CAREER award to Neil Heffernan, and the Spencer Foundation. All the opinions, findings, and conclusions expressed in this article are those of the authors, and do not reflect the views of any of the funders.
REFERENCES


Feng, M., Heffernan, N. T. Can We Get Better Assessment From A Tutoring System Compared to Traditional Paper Testing? Can We Have Our Cake (Better Assessment) And Eat It too (Student Learning During The Test)? Submitted to the 10th International Conference on Intelligent Tutoring Systems, 2010. Pittsburgh, PA.


Wang, Y., Heffernan, N.T.: The “Assistance” Model: Leveraging how many hints and attempts a student needs. Accepted by the 24th International FLAIRS Conference ITS special track, Palm Beach, Florida (2011)