Towards a Model-Free Estimate of the Limits to Student Modeling Accuracy

Binglin Chen
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
chen386@illinois.edu

Matthew West
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
mwest@illinois.edu

Craig Zilles
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
zilles@illinois.edu

ABSTRACT

This paper attempts to quantify the accuracy limit of "next-item-correct" prediction by using numerical optimization to estimate the student's probability of getting each question correct given a complete sequence of item responses. This optimization is performed without an explicit parameterized model of student behavior, but with the constraint that a student's likelihood of getting a problem correct only increases or remains unchanged with additional practice (i.e., no forgetting). We present results for this method for the Assistments 2009–2010 data where it suggests that there is only modest opportunity for improvement beyond the state of the art predictors. Furthermore, we describe a framework for applying this method to datasets where problems can be tagged with multiple skills and problem difficulties. Lastly, we discuss the limitations of this method, specifically its inability to give tight bounds on short sequences.

1. INTRODUCTION

Student modeling is a fundamental building block of educational systems that are intelligent or adaptive. With a model of a student, such a system can consider all of the actions it has available and make a prediction about which ones are likely to be the most profitable for a particular student at the current time.

One class of student models tries to predict *next-item-correct*, i.e., what is the probability that a student's attempt on the next item presented will be correct given the student's results on all previous items. For a number of years, this topic saw vigorous research with non-trivial improvements using improved model parameterizations [1, 6, 7, 11] and recurrent neural networks [10]. Yet, performance of next-item-correct predictors has seemed to reach an asymptote that is far below perfect prediction.

This gap between the current state of the art and perfect prediction raises the question of how much headroom remains for further improvements to next-item-correct prediction. Previous work by Beck and Xiong [2] has attempted to characterize the accuracy limit by analyzing the performance of a collection of "cheating" prediction algorithms that employ a partial knowledge of future results. They conclude that further large improvements in prediction accuracy are unlikely.

Estimating a tight bound to prediction accuracy is challenging, because one needs to utilize some information about future correctness without merely regurgitating the stream of actual outcomes as one's predictions, which would yield the tautological bound of 100% accuracy. Beck and Xiong navigate this conundrum by allowing their cheating model to correctly predict the transitions from giving an incorrect response to giving a correct response (e.g., learning), but not those from giving an correct response to giving a incorrect response (in their words, "forgetting"). We found this approach to be unsatisfying in two respects. First, the time period in which the data is collected is too short for true forgetting to take place, it is rather more likely to be slipping, so we feel that the model is a mismatch for the phenomena at hand. Second, we feel that perfectly predicting incorrect-to-correct transitions but not correct-to-incorrect transitions seems arbitrary.

Instead, we posit that the limits of accuracy for next-item-correct prediction derive from the fact that learning is not a binary transition from a state of not knowing to a state of knowing, but rather that there is a continuum of knowledge levels that a student could be at. For example, there is a point on this continuum where a student will get 50% of the problems attempted correct and the other 50% incorrect. The challenge for next-item-correct prediction for such a student is precisely determining whether the next attempt will be correct or incorrect, much like the hopeless task of trying to consistently predict the outcome of flipping a fair coin. More precisely, it is the student responses as they transition from not knowing to knowing that are hard to predict, as the behavior of perfectly knowledgeable and perfectly unknowledgeable students is trivial to predict.

Thus, the limit for prediction should primarily derive from the fraction of a data stream during which students are in this transitional phase where they are intermingling correct and incorrect responses. This can be viewed as the amount of entropy in the data, and this entropy can and does vary from dataset to dataset. As such, we believe that a method

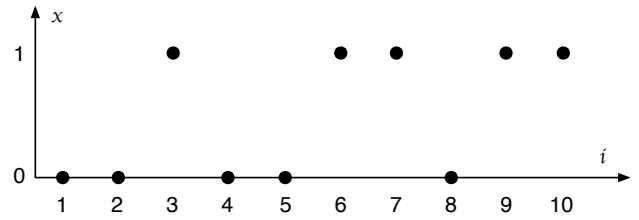

Figure 1: Illustrative example of the input to the next-item-correct prediction problem. For this example, $n = 10$ and $x_1, \ldots, x_n = 0, 0, 1, 0, 0, 1, 1, 0, 1, 1$.

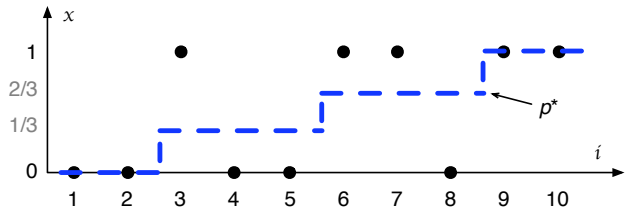

Figure 2: Results of the model-free bounding method when all items require the same knowledge component.

that can estimate the limits of predictability as a function of this entropy can serve as a less arbitrary estimate of the accuracy limit for next-item-correct prediction and serve as a useful means for characterizing and comparing datasets.

This paper is organized as follows. We first formalize the next-item-correct prediction problem in Section 2. We then describe our model-free bounding method in Section 3. We show experimental results of our method in Section 4. Finally, we discuss the limitations of our method in Section 5 and future directions in Section 6.

2. NEXT-ITEM-CORRECT PREDICTION

We formalize the next-item-correct prediction problem as follows. We are given a length-n sequence $x_1, \ldots, x_n$, where $x_i = 1$ if the student answered the ith attempted item correctly and $x_i = 0$ otherwise, as shown in Figure 1. Given this information, we want to produce n reals $p_1, \ldots, p_n$ where p_i is the probability of the student answering the ith attempted item correctly. Typically models are required to produce $p_1, \ldots, p_n$ in order and they are only allowed to look at $x_1, \ldots, x_{t-1}$ when producing p_t, as future observations should not be available during prediction. Some of the notable models for this task are Bayesian Knowledge Tracing (BKT) [3], Performance Factor Analysis (PFA) [8], and Deep Knowledge Tracing (DKT) [10].

In efforts to improve their performance, many models use the *knowledge components* required by each item, denoted as $\vec{s}_1, \ldots, \vec{s}_n$. Each $\vec{s}_i$ is a d dimensional vector where d is the number of knowledge components in the corresponding dataset. Each entry of $\vec{s}_i$ is typically boolean, indicating whether the item requires the corresponding knowledge component. The entries of $\vec{s}_i$ can be real valued as well, indicating the degree of mastery required on each component in order to answer the item correctly.

With the ground truth $x_1, \ldots, x_n$ and predictions of a model $p_1, \ldots, p_n$, a performance metric $\mathcal{L}$ is typically used to measure how good the predictions are. The most widely used metrics for this task are root mean squared error (RMSE) and area under the curve (AUC) [9]. Log likelihood (LL) has also been proposed [9] though it has not been widely used on this task. This paper will use average LL instead of LL since the former does not depend on the size of the data. Models with better $\mathcal{L}(p_1, \ldots, p_n; x_1, \ldots, x_n)$ are to be preferred. The meaning of "better" depends on the metric; larger values are better for average LL and AUC while smaller values are better for RMSE.

3. MODEL-FREE ACCURACY BOUNDS

The core idea of our method is that the probability of a student correctly answering items that require the same knowledge components should be non-decreasing over the short term. More precisely, if the current item is no more difficult than a previous item that requires the same knowledge and there hasn't been sufficient time or interference for forgetting to occur, the student's probability of getting the current item correct should be at least as high as the previous item.

This idea is illustrated in Figure 2, where the dashed line segments correspond to the probability of the student correctly answering each item. One could interpret this sequence as having three phases: (1) items 1 and 2 as a region of unknowing where the student gets every item incorrect, (2) items 3 through 8 as a region of learning where correct and incorrect responses are interleaved, and (3) items 9 and 10 as a region of mastery where the student gets every item correct. Even though the second region includes both correct and incorrect responses, we are interpreting those merely as events from an underlying probability distribution and that probability of correct responses is non-decreasing throughout the sequence.

Based on this idea, our proposed bounding method finds correctness probabilities for each item $p_1^*, \ldots, p_n^*$ that optimize $\mathcal{L}(p_1^*, \ldots, p_n^*; x_1, \ldots, x_n)$ subject to the constraint that the p_i^* sequence is non-decreasing on appropriate item sequences. These p_i^* provide the best local estimate of the likelihood that a student will get an item correct given an assumption that only learning is occuring. To do better, one would have to predict the precise sequence of correct and incorrect responses and we believe that this problem is akin to predicting the precise sequence of heads and tails from repeated flips of a coin. As such, we expect this to be a practical bound to next-item-correct prediction.

We refer to this method as being "model free", because it does not rely on any parameterized model of student behaviors and does not require training. Instead, the p_i^* values are derived directly from the sequence $x_1, \ldots, x_n$ and, therefore, can be potentially applied on any dataset.

3.1 Single knowledge component case

Before diving into the case where multiple knowledge components are involved, we first explain our method in the

simplest case where the sequence of items require the same knowledge component. In this case, since all of the items are equivalent in terms of knowledge components, the aforementioned constraint is equivalent to constraining $p_1, \ldots, p_n$ to be non-decreasing. Thus our method reduces to solving the following numerical optimization problem to obtain $p_1^*, \ldots, p_n^*$:

$$\begin{aligned} \text{optimize: } & \mathcal{L}(p_1, \ldots, p_n; x_1, \ldots, x_n) \\ \text{subject to: } & 0 \leq p_i \leq 1 \text{ for all } i \\ & p_i \leq p_j \text{ for all } i < j. \end{aligned} \tag{1}$$

This numerical optimization problem can be solved efficiently by an interior point method if (1) $\mathcal{L}$ is convex and smaller $\mathcal{L}$ is better, or (2) $\mathcal{L}$ is concave and larger $\mathcal{L}$ is better. Out of the three metrics mentioned previously, average LL and RMSE satisfy this criterion while AUC is not even continuous (and hence not convex or concave). Thus this formulation as a numerical optimization problem is only applicable when $\mathcal{L}$ is average LL or RMSE. There are various tools that can solve this sort of numerical optimization problem. In our implementation we used used Matlab's *fmincon* with L-BFGS as the Hessian method.

To give a sense of what this method produces, Figure 2 shows as the dashed line the values $p_1^*, \ldots, p_n^*$ that minimize RMSE for the given observed item responses $x_1, \ldots, x_n$ (solid black dots).

3.2 Partial order of items

In order to handle sequences of items with different combinations of multiple knowledge components, we need to be able to compare the items and decide which previously attempted items provide information useful for predicting the outcome of the current item. The intuition is that if item a is the same difficulty or easier with respect to the required knowledge components than item b, then a student should do item a at least as well as item b. We compare items by defining a partial order $\preceq$ over the knowledge component vectors as follows:

$$\vec{s}_a \preceq \vec{s}_b \iff \vec{s}_{a,k} \leq \vec{s}_{b,k} \text{ for all } k, \tag{2}$$

where $\vec{s}_{a,k}$ is the kth coordinate of $\vec{s}_a$. This partial order essentially states that item a should be considered easier than or equal to item b if the required mastery level of each knowledge component of item a is less than or equal to that of item b. Intuitively, given $\vec{s}_a \preceq \vec{s}_b$, then a student should be able to answer item a correctly if the student can answer item b correctly.

Given this definition of partial order, we can induce a directed acyclic graph (DAG) on the set of items, where there is an edge from the jth item to the ith if and only if $i < j$ and $\vec{s}_j \preceq \vec{s}_i$. The intuition of the requirement $i < j$ is that being able to solve a "harder" item in the past implies being able to solve an "easier" item in the future. However, being able to solve a "harder" item in the future does not imply being able to solve an "easier" item in the past since the student might have learned a lot in between. To illustrate this, we show the DAG induced by a sequence of 6 items with 3 knowledge components in Figure 3. In such a DAG, an edge from the jth item to the ith means that the student should be able to do the jth item at least as well as the ith item.

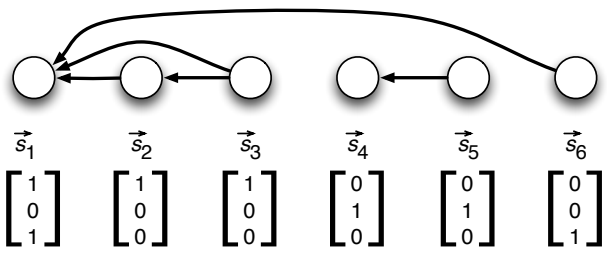

Figure 3: A directed acyclic graph induced by the partial order. An arrow from the jth item to the ith item means that the student should do the jth item at least as well as the ith item. There are two connected components in this induced graph, which are $\{x_1, x_2, x_3, x_6\}$ and $\{x_4, x_5\}$.

3.3 Multiple knowledge components case

Given the partial order on items as described above, we can generalize the non-decreasing constraints for a single knowledge component to handle any combination of knowledge components. Specifically, given $i < j$ and $\vec{s}_j \preceq \vec{s}_i$, the probability p_j of the student answering the jth item correctly should not be lower than the probability p_i of the ith item since the jth item is no harder than the ith item. That is, $p_i \leq p_j$ when there is an edge from the jth item to the ith item in the induced DAG on the sequence. Thus the optimization problem can be reformulated as

$$\begin{aligned} \text{optimize: } & \mathcal{L}(p_1, \ldots, p_n; x_1, \ldots, x_n) \\ \text{subject to: } & 0 \leq p_i \leq 1 \text{ for all } i \\ & p_i \leq p_j \text{ for all } i < j \text{ that satisfy } \vec{s}_j \preceq \vec{s}_i. \end{aligned} \tag{3}$$

This complicated optimization problem can usually be broken down into smaller ones by dividing the sequence $x_1, \ldots, x_n$ into shorter subsequences based on the connected components they belong to in the induced DAG. In the example depicted by Figure 3, there are two connected components which correspond to $\{x_1, x_2, x_3, x_6\}$ and $\{x_4, x_5\}$. We can then optimize on each subsequence separately.

Another trick to accelerate the optimization is removing redundant constraints since the partial order is transitive. For example, the constraint corresponding to the edge from $\vec{s}_3$ to $\vec{s}_1$ in Figure 3 can be safely removed since it is implied by constraints corresponding to $\vec{s}_3 \preceq \vec{s}_2$ and $\vec{s}_2 \preceq \vec{s}_1$.

3.4 Metrics that cannot be directly optimized

As mentioned before, our method is not applicable to AUC since it is not continuous. To compute a bound for AUC, we first solve the optimization problem by either maximizing average LL or minimizing RMSE. Once we obtained p_i^* for the entire dataset, we can calculate AUC using these p_i^*.

In general, we can always optimize on one metric $\mathcal{L}$ for p_i^* and evaluate the p_i^* with any metric $\mathcal{L}'$ even though the optimization is done with respect to $\mathcal{L}$. We refer to this as the bound obtained by optimizing $\mathcal{L}$.

4. EXPERIMENTAL RESULTS

We applied BKT, DKT, and our method to the Assistments 2009–2010 dataset. We chose this dataset because it has

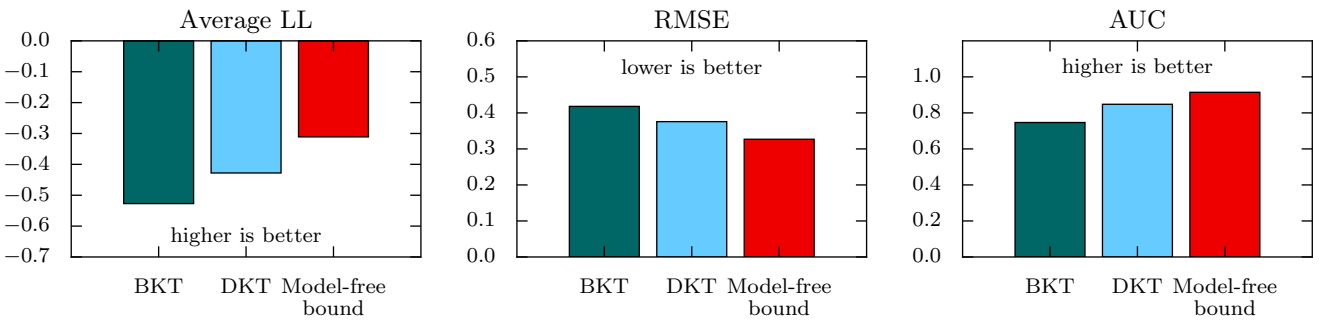

Figure 4: Results of applying BKT, DKT, and our method to Assistments 2009–2010 dataset.

relatively long sequences of attempts. We used the same train/test split for this dataset as in Khajah et al. [5]. We used the BKT implementation by Yudelson[1] [11] with the default parameters and Baum-Welch as the training method. We used Khajah et al.'s [5] implementation of DKT[2] with default parameters. We only applied our method to the test set for meaningful comparisons.

For the rest of this paper, we only report bounds obtained by maximizing average LL. Throughout our experiments, we found that the bounds for all of average LL, RMSE, and AUC obtained by minimizing RMSE differed by less than 0.5% from those obtained by maximizing average LL. In fact, it can be proved that minimizing RMSE and maximizing average LL will yield the same p_i^* in the single knowledge component case (Equation 1). See the Appendix for the proof.

We show our results on Assistments 2009–2010 for average LL, RMSE, and AUC in Figure 4. The performance of DKT is roughly half way between BKT and the bound produced by our method for all of the metrics. This suggests that the room for further improvements on Assistments 2009–2010 is limited.

5. LIMITATIONS

The major limitation of our method is its optimistic nature, meaning that it can produce a bound that is too loose. This optimism manifests in two ways: first, our method can predict the precise location of learning transitions, which will be difficult for any realistic model, and, second, more generally when the sequence of predictions to be made is short the model isn't significantly constrained.

5.1 Predicting Particular Events

The proposed technique appears to provide a reasonable bound of prediction performance when student behavior follows a non-instantaneous learning of a topic involving an interleaving of correct and incorrect responses as shown in Figure 1. However, when students transition instantly from consistently answering incorrectly to consistently answering correctly, the model will likely produce a bound that is too loose. Consider the item response sequences of two students shown in Figure 5. Both of these students only transition

[1]https://github.com/IEDMS/standard-bkt
[2]https://github.com/mmkhajah/dkt

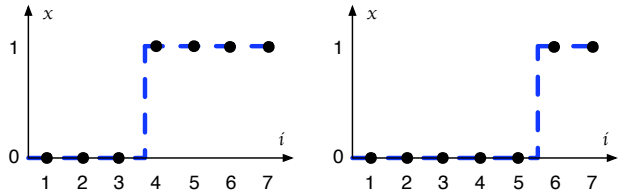

Figure 5: Two sequences that our method predicts perfectly. A real predictor, however, might have trouble predicting the precise location of the upward transition.

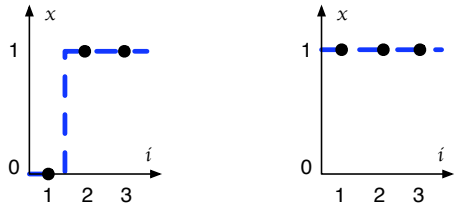

Figure 6: Our method can predict initial behavior perfectly in some circumstances. A real predictor, however, might have trouble predicting precisely which students would get a problem correct on their first attempt.

from incorrect responses to correct responses, meaning that the optimization is free to generate predictions that precisely match the data, resulting in 100% accuracy. A real model, however, must predict the point of the transition, knowing that after observing the first three incorrect responses it should predict correct for the first student's fourth attempt and incorrect for the second student's fourth attempt. While it isn't impossible to imagine that there are features to guide such a prediction, it is difficult to believe that it could be done consistently with 100% accuracy.

A special case of predicting such a transition is predicting whether or not the very first attempt is going to be correct. As shown in Figure 6, our method can perfectly predict whether or not a student gets their first attempt correct, provided the student gets all other attempts correct. A real system might be challenged to predict precisely which students would perform in this manner, although some knowl-

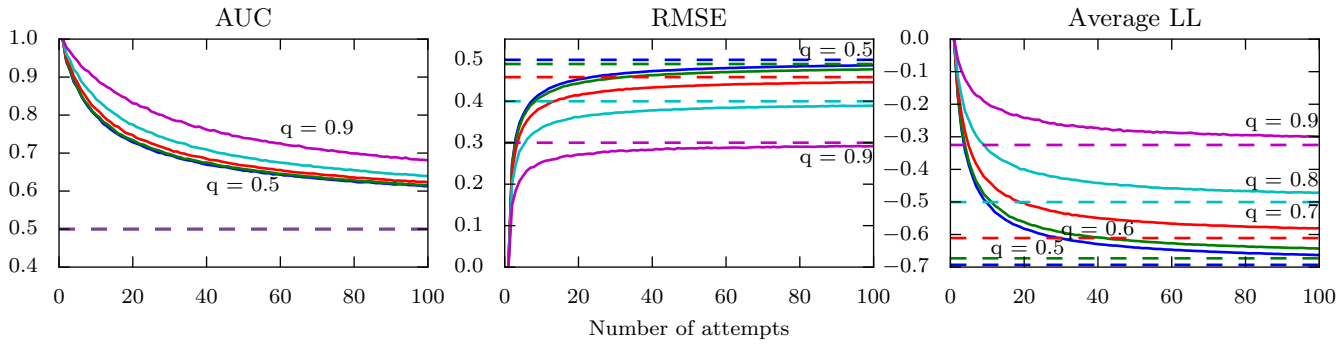

Figure 7: Upper bounds produced by our method versus theoretical bounds for attempt results that are i.i.d. with fixed q for various sequence lengths. The solid curves correspond to the results of our method and the dashed lines correspond to the theoretical bounds.

edge about the students will certainly enable such predictions to be performed at a rate better than just the average frequency that students get a given question correct on their first attempt. Nevertheless, these features of the data lead our system to be optimistic, and these features occur more frequently and have larger impact on short sequences.

5.2 Short Sequences

In general, our method struggles with short sequences, because the optimization is largely unconstrained. For example, consider the case where every student has made exactly one attempt. In such a case our method will always produce p_1^* that is exactly the same as x_1, which results in a trivial bound of 100% accuracy. However, as the sequence length increases, the constraints will generally prevent our method from being perfectly accurate, and thus it will provide a more useful bound.

To understand how the amount of optimism in our method depends on the sequence length, we used independent and identically distributed (i.i.d.) coin tosses to study this. Such sequences allow us to compute a theoretical bound that we can compare to the one produced by our method. When attempt results $x_1, \ldots, x_n$ are i.i.d. with probability q of being correct, the theoretical bound is $q \log q + (1-q) \log(1-q)$ for average LL, $\sqrt{q(1-q)^2 + (1-q)q^2}$ for RMSE, and 0.5 for AUC.

Specifically, we generated i.i.d. results with sequence lengths ranging from 1 to 100 and with q ranging from 0.1 to 0.9 and same $\vec{s}$ for every attempt. For each length, we generated 10,000 sequences and computed the bound for average LL, RMSE, and AUC using our method.

We plotted the bounds computed by our method and the theoretical bound in Figure 7. We chose to not plot the results for q from 0.1 to 0.4 in the figure since we found that q and $1-q$ yield the same results. The solid curves in the figure correspond to the results of our method for each q while the dashed lines correspond to the theoretical bound for each q. As the figure shows, our method starts off wildly optimistic when the sequence length is 1 and gradually converges to the theoretical bounds as the sequence length increases. At a sequence length of 100, the bounds by our method are close to the theoretical bound for average LL and RMSE but not AUC. These trends suggest that our method works reasonably well for average LL and RMSE when the sequence length is large enough, however it is too optimistic on AUC even with long sequences.

6. DISCUSSION AND CONCLUSION

In this paper, we presented a model-free bounding method to find the limit of the next-item-correct prediction task. The method assumes that forgetting is absent and uses the constraint that the probability of students correctly answering a set of similar items should not decrease as they practice more. We applied our method to the Assistments 2009–2010 dataset and found that DKT's performance on this dataset is fairly close to the bound produced by our method. This suggests that the room for improvement on this dataset is small.

The main shortcoming of our method is its optimistic nature. In other words, our method will produce a bound that is too loose, especially for short sequences. While we can conceive of many ways to potentially compensate for this optimism (motivated by the scenarios discussed in Section 5), we fear that any attempts we make to estimate compensation factors has the potential to yield a result that no longer serves as a bound (i.e., that a real implementation could potentially achieve a performance exceeding our "bound"). Furthermore, we view the parameter-free simplicity of our method to be one of its virtues, and it is not clear how to preserve that while introducing such compensation. The other shortcoming is that our method does not incorporate forgetting by default. However, this could potentially be incorporated by relaxing constraints when forgetting is suspected to have occurred.

The intuition behind our method is based on the reason why next-item-correct prediction is feasible. Since independent identically distributed (i.i.d.) coin tosses are inherently unpredictable, next-item-correct prediction is feasible only if there are regularities in the data. Learning is undoubtedly the most important regularity that we would like to observe in any educational system. Thus the difficulty of the next-item-correct prediction task depends on how much students' performance deviates from i.i.d. and shows non-decreasing behavior. Our method tries to capture such regularities due to learning.

7. ACKNOWLEDGEMENTS

This work was partially supported by NSF DUE-1347722, NSF CMMI-1150490, and the College of Engineering at the University of Illinois at Urbana-Champaign under the Strategic Instructional Initiatives Program (SIIP). The authors would like to thank Luc Paquette for useful discussions.

APPENDIX

To prove that minimizing RMSE is equivalent to maximizing average LL in the case of Equation 1, we first recall the concept of a *scoring rule* [4], which is a function that scores a predictive probability distribution P against an observation x_i drawn from a target probability distribution Q that we are trying to recover. In this context a larger score indicates a better P. In the case of binary variables with range $\{0, 1\}$, both P and Q are Bernoulli distributions and a scoring rule can be simply denoted as $S(p, x)$, where p is the probability of observing 1 in P and x is an observation drawn from Q.

A *strictly proper scoring rule* is a scoring rule such that the expected score over a set of observations drawn from Q is uniquely maximized when $P = Q$ [4]. The *quadratic score* and the *logarithmic score* are two commonly used strictly proper scoring rules. In the case of Equation 1, maximizing the quadratic score is equivalent to minimizing RMSE and maximizing the logarithmic score is equivalent to maximizing average LL.

In the binary case, a strictly proper scoring rule $S(p, x)$ has the Savage representation $S(p, x) = G(p) + G^*(p)(x - p)$ where G is strictly convex and G^* is a subdifferential of G [4]. Define the cost function $F(p; x_1, \ldots, x_n)$ by $F(p) = \frac{1}{n}\sum_{i=1}^{n} S(p, x_i) = G(p)+G^*(p)(\bar{x}-p)$ where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

LEMMA 1. *$F(p)$ has a unique maximum at $p = \bar{x}$ and is strictly quasiconcave, thus unimodal.*

PROOF. First observe that $F(p) = G(p)+G^*(p)(\bar{x}-p) \leq G(\bar{x}) = F(\bar{x})$ by the definition of the subdifferential, with equality if and only if $p = \bar{x}$. Thus $p = \bar{x}$ is the unique maximum.

To establish quasiconcavity, we will show that for any $\alpha \in (0, 1)$, $F(\alpha p + (1 - \alpha)q) > \min\{F(p), F(q)\}$. Let $r = \alpha p + (1 - \alpha)q$ and, without loss of generality, assume $p < q$, so either $p < r \leq \bar{x}$ or $\bar{x} \leq r < q$. In the first case:

$$\begin{aligned} F(r) - F(p) &= G(r) - G(p) + G^*(r)(\bar{x} - r) - G^*(p)(\bar{x} - p) \\ &> G^*(p)(r - p) + G^*(r)(\bar{x} - r) - G^*(p)(\bar{x} - p) \\ &= (G^*(r) - G^*(p))(\bar{x} - r) \\ &\geq 0. \end{aligned}$$

The last step is due to monotonicity of G^*, which states that $(G^*(r)-G^*(p))(r-p) \geq 0$, and because $(\bar{x}-r)$ has the same sign as $(r-p)$ we have $(G^*(r)-G^*(p))(\bar{x}-r) \geq 0$. This establishes that $F(r) > F(p)$ in the first case. Similarly, $F(r) > F(q)$ in the second case, thus $F(r) > \min\{F(p), F(q)\}$. □

For any solution to Equation 1, we can partition $p_1, \ldots, p_n$ into blocks (subsets) where each member of a block has equal value and no two blocks share a value. Because Equation 1 requires monotonicity, each block must have consecutive indices.

LEMMA 2. *If $\mathcal{L}$ is a strictly proper scoring rule, then every solution to Equation 1 consists of blocks of the form $p_i = \ldots = p_j = \overline{\{x_i, \ldots, x_j\}} = \sum_{k=i}^{j} x_k/(j - i + 1)$.*

PROOF. Consider any block $p = p_i = \ldots = p_j$ in a solution to the optimization problem described by Equation 1 when $\mathcal{L}$ is a strictly proper scoring rule. Because blocks have distinct values, p is locally unconstrained and so Lemma 1 implies $p = \overline{\{x_i, \ldots, x_j\}}$. □

Algorithm 1

1: $i \leftarrow 1$
2: **while** $i \leq n$ **do**
3: find the largest j with $i \leq j \leq n$ that minimizes $\overline{\{x_i, \ldots, x_j\}}$
4: $p_i, \ldots, p_j \leftarrow \overline{\{x_i, \ldots, x_j\}}$
5: $i \leftarrow j + 1$
6: **end while**

THEOREM 1. *If $\mathcal{L}$ is a strictly proper scoring rule, then Algorithm 1 gives the unique solution to Equation 1.*

PROOF. Let $p_1^*, \ldots, p_n^*$ be the output of Algorithm 1. Assume that $p_1, \ldots, p_n$ is a distinct solution to Equation 1. Let k be the first index for which $p_k^* \neq p_k$ and let $p_i^*, \ldots, p_j^*$ be the block with $i \leq k \leq j$.

If $p_k < p_k^*$, then monotonicity implies $k = i$. Let $\{p_k, \ldots, p_\ell\}$ be the following block, so $p_k^* > p_k = \overline{\{x_k, \ldots, x_\ell\}}$, which contradicts Line 3 in Algorithm 1.

If $p_k > p_k^*$, then $p_k^* < p_k \leq \overline{\{x_k, \ldots, x_j\}}$ because the optimization subproblems for blocks in $\{p_k, \ldots, p_j\}$ are locally unconstrained below. But by Lemma 2 we have:

$$\begin{aligned} p_k^* &= \overline{\{x_i, \ldots, x_j\}} \\ &= \frac{k-i}{j-i+1}\overline{\{x_i, \ldots, x_{k-1}\}} + \frac{j-k+1}{j-i+1}\overline{\{x_k, \ldots, x_j\}} \\ &> \frac{k-i}{j-i+1}p_k^* + \frac{j-k+1}{j-i+1}p_k^* \\ &= p_k^*, \end{aligned}$$

which is again a contradiction.

Note that Algorithm 1 does not depend on $\mathcal{L}$, so all strictly proper scoring rules give the same solution to Equation 1. □

REFERENCES

[1] R. S. Baker, A. T. Corbett, and V. Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems*, pages 406–415. Springer, 2008.

[2] J. Beck and X. Xiong. Limits to accuracy: how well can we do at student modeling? In *Educational Data Mining*, 2013.

[3] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-adapted Interaction*, 4(4):253–278, 1994.

[4] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

[5] M. Khajah, R. V. Lindsey, and M. C. Mozer. How deep is knowledge tracing? In *Educational Data Mining*, 2016.

[6] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Educational Data Mining 2014*. Citeseer, 2014.

[7] Z. A. Pardos and N. T. Heffernan. Kt-idem: introducing item difficulty to the knowledge tracing model. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 243–254. Springer, 2011.

[8] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis–a new alternative to knowledge tracing. In *The 14th International Conference on Artificial Intelligence in Education*, 2009.

[9] R. Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 7(2):1–19, 2015.

[10] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[11] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180. Springer, 2013.