Online Estimation of Student Ability and Item Difficulty with Glicko-2 Rating System on Stratified Data

Jaesuk Park
Knowre Korea Inc.
epark@knowre.com

ABSTRACT

We propose an adaptation of the Glicko-2 rating system in a K-12 math learning software setting, where variable time intervals between solution attempts and the stratification of student-item pairings by grade levels necessitate modification of the original model. The discrete-time stochastic process underlying the original system has been modified into a continuous-time process to account for the irregularity of intervals between solution attempts. Also, conceptual prerequisite relationships between items were used to provide initial rating estimates that allow for rating values to be meaningfully compared across grade levels. Fitting the model using real student learning data results in rating value distributions successfully exhibiting a gradation with the increase of grade level. A potential area of application in a personalized education setting is also briefly discussed.

Keywords

Item response theory, dynamic paired comparison model, stratified data, educational assessment, stochastic variance model

1. INTRODUCTION

We consider the problem of assigning appropriate curriculum levels in a large-scale K-12 math learning software to students who are substantially ahead or behind their peers. Previous studies have suggested the importance of matching learning content difficulty to a student's ability for positive student learning outcomes [3, 10, 16]. In light of this, students who are much farther ahead (e.g., gifted students) or behind their peers (e.g., students with learning disabilities) can benefit much from receiving a more tailored educational feedback, based on learner and skill models that can model their differences more effectively.

With the recent advances in computing devices, various approaches have been sought to harness the power of computing to model learners more accurately in educational contexts, as comprehensively overviewed in [2]. In one particular line of approach [11, 13, 12, 15], dynamic paired comparison models were used to quickly estimate student abilities and item difficulties in a scalable manner. In these adaptations, the players consist of students ("users") and units of learning task (e.g., problem items, assignments), and each solution attempt is conceptualized as a match between a student and a learning task, in which the winner earns 1 point and the loser earns 0 points (with no draw). The primary advantage of such models over traditional IRT methodologies is in their ability to compute ability estimates "on the fly" [11] while retaining a similar mathematical structure to IRT.

The problem occurs, however, when the dataset is *stratified*—i.e. when student-problem pairings can be grouped into distinct (or largely nonoverlapping) groups such that a problem's rating cannot be adequately adjusted by a student outside the group to which it belongs. In a K-12 math learning software, because students are only exposed to problems appropriate for their grade level, grade levels serve as strata. Consequently, we cannot adequately tell how a student would perform outside of their regular grade level just by looking at the student's rating value. See Fig. 1 for an illustration.

Ideally, we would not have this problem by gathering enough learning data from a large number of students for 12+ years, during which they would work through all curricula offered by the product in sequence. However, in a commercial educational software context where a user is not bound to use products from just one vendor, this is highly impractical.

Hence we raise a question: is there a way to enforce rating values to reflect the relative positions of the strata, despite the absence of sufficient overlaps in students/items among them? One possible strategy is to initialize the ratings differently for each stratum according to their relative positions, e.g., to initialize first-grade rating values to 100, second-grade rating values to 200, etc., and then let the dynamic paired comparison algorithm do the calibration within each grade level. But then how could we justify that the initial estimation done for all curricula is properly reflective of their actual difficulties relative to one another?

Here, the key insight is that the partial ordering of mathematical concepts due to prerequisite relationships provides a basis for the division of concepts into grade-level curric-

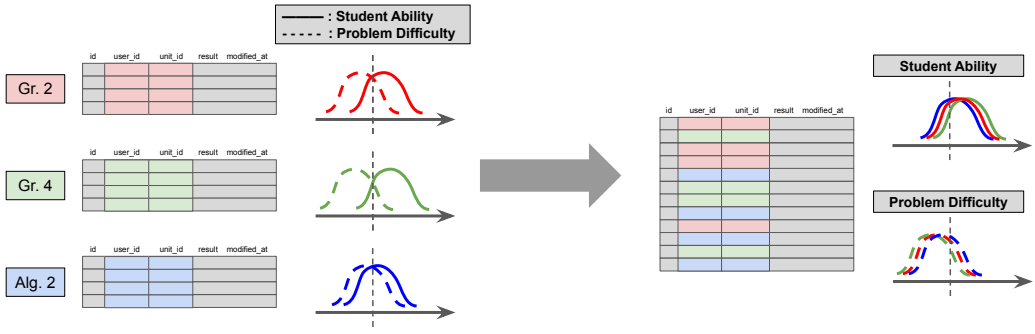

Figure 1: An illustration of the impact of data stratification on the rating interpretability. As a result of stratification, the distributions of rating values can overlap unreasonably much with each other, and the corresponding mean rating values may not align with the actual order of grade levels.

ula, which then in turn stratifies the learning data. In the K-12 math learning software used in our study, each problem item is conceptualized as a particular instantiation of a mathematical concept ("knowledge unit," or just "unit") with specific values. These mathematical concepts have prerequisite relationships defined among them, the collection of which can be represented as a directed graph. We attempt to employ these relationships to obtain statistically interpretable and contextually appropriate estimations.

Specifically, our contribution is twofold: 1) modification of a dynamic paired comparison rating system model to account for imbalance in rating update frequencies between students and items, and 2) use of prerequisite relationships between concepts for rating initialization to achieve rating comparability between curriculum levels. We aim to yield, from a stratified dataset, a set of ratings that can be meaningfully compared across grade levels: where students and items in a lower grade level would generally have lower ratings than those in a higher grade level.

The remainder of this paper is organized as follows. Section 2 presents our particular adaption of a dynamic paired comparison model, including the details for incorporating the conceptual prerequisite information into rating initialization. Section 3 describes the dataset used for evaluating our model and presents our results. Section 4 discusses the potential for applying our model to assign grade levels for students far ahead or behind their peers, lists some of the limitations of our work, and suggests a few possible directions for further research.

2. MODEL

The Glicko-2 rating system [7] falls under the family of dynamic paired comparison models, along with the Glicko rating system [6] (its predecessor) and the Elo rating system [4] (of which the two Glicko systems are extensions). Improving upon its predecessor, the Glicko-2 rating system models the change in variance of player strength as another stochastic process, thereby accounting for the possibility of sudden changes in strength. More specifically, the algorithm models the change in player strength per unit time with a normal distribution with variance equal to the square of the rating *volatility*, whose logarithmic change per unit time is itself normally distributed.

2.1 Continuous-time Glicko-2 Model

The original Glicko-2 system presented in [7] assumes the underlying stochastic processes to be discrete-time, where the overall measurement period is discretized into time increments called "rating periods." Within each rating period, the matches are assumed to occur simultaneously. However, because there is too much imbalance in the average number of matches between users and items, [7]'s recommendation of having 5-10 matches per rating period for every player is not feasible to implement in our application context. [15] has successfully worked around this limitation by constraining each rating period to contain only one match, but the workaround did not account for an increase in rating uncertainty due to the passage of time, which is a key feature of the Glicko rating system family. Here, we take the approach of modifying the Glicko-2 model under a continuous-time stochastic process framework, so that the model can account for rating uncertainty increase due to the passage of time without discretizing the measurement period.

Let $\theta_s(t)$ denote the ability estimate of user s at time t, and let $\beta_i(t)$ denote the difficulty estimate of unit i at time t. Then as a result of using continuous-time stochastic process framework, the model equations for latent trait parameters become

$$\theta_s(t) \;\sim\; \mathcal{N}(\mu_s(t),\; \phi_s^2(t)). \tag{1}$$

$$\theta_s(t{+}\Delta t) \,|\, \theta_s(t),\; \sigma_s^2(t{+}\Delta t) \;\sim\; \mathcal{N}(\theta_s(t),\; \Delta t\, \sigma_s^2(t{+}\Delta t)) \tag{2}$$

$$\log \sigma_s^2(t+\Delta t) \;|\; \log \sigma_s^2(t),\; \tau^2 \;\sim\; \mathcal{N}(\log \sigma_s^2(t),\; \tau^2) \tag{3}$$

for user ability estimates, and

$$\beta_i(t) \;\sim\; \mathcal{N}(\mu_i(t),\; \phi_i^2(t)) \tag{4}$$

for unit difficulty estimates. Here, as in [8], μ denotes rating, ϕ denotes rating deviation (RD), and σ denotes rating volatility. Note that the difficulty of a mathematical concept is expected to remain constant over time, so we do not impose any stochastic volatility assumption on $\beta_i(t)$.

As for the correctness probability (i.e., the probability of user s correctly answering an instantiation of unit i at time

t), the Glicko rating system family differs from the Elo rating system in its incorporation of rating uncertainty to calculate this quantity. We are generally interested in the correctness probability *before* the user s actually attempts unit i. However, the time elapsed between the user's last attempt and the current attempt can vary throughout the user's activity history, which also varies the amount of inflation to apply each time on the user's rating uncertainty, ϕ_s. Hence we apply equation (2) prior to calculating the correctness probability. Let t_s and t_i denote the last time user and unit latent trait estimates, respectively, were updated. Let $Y_{s,i}(t)$ be a Bernoulli random variable denoting user response correctness. Then the correctness probability is given by:

$$\Pr(Y_{s,i}(t) = 1) = E(\hat{\mu_s}(t),\ \hat{\mu_i}(t),\ \hat{\phi_s^2}(t) + \hat{\phi_i^2}(t)) \quad (5)$$

where $E(\mu_1,\ \mu_2,\ \phi^2) = \left[1 + e^{-g(\phi^2)(\mu_1 - \mu_2)}\right]^{-1}$ is the expected score function that accounts for rating uncertainty [7], and

- $g(\phi^2) = \left[1 + \frac{3\phi^2}{\pi^2}\right]^{-1/2}$,
- $\hat{\phi_s^2}(t) = \phi_s^2(t_s) + (t - t_s)\sigma_s^2(t_s)$,
- $\hat{\phi_i^2}(t) = \phi_i^2(t_i)$,
- $\hat{\mu_s}(t) = \mu_s(t_s)$, and
- $\hat{\mu_i}(t) = \mu_i(t_i)$.

Here, we use $\sigma_s^2(t_s)$ in place of $\sigma_s^2(t)$ to estimate $\hat{\phi_s^2}(t)$, although their equivalence only holds in expectation.

After user s finishes solution attempt for unit i with result $y_{s,i} \in \{0, 1\}$, the update equations for latent trait estimates are given as below, following [7]'s derivation of corresponding equations under the continuous-time framework:

$$\sigma_s^2(t) = \exp\left(\arg\max_{a(t)} p(a(t)|y_{s,i})\right) \quad (6)$$

$$\phi_s^2(t) = \min\left\{\phi_s^2(0), \left[\frac{1}{\phi_s^2(t_s) + \sigma_s^2(t)} + \frac{1}{v_s^2(t)}\right]^{-1}\right\} \quad (7)$$

$$\phi_i^2(t) = \min\left\{\phi_i^2(0), \left[\frac{1}{\phi_i^2(t_i)} + \frac{1}{v_i^2(t)}\right]^{-1}\right\} \quad (8)$$

$$\mu_s(t) = \mu_s(t_s) + \phi_s^2(t) \cdot g(\hat{\phi_i^2}(t)) \cdot (y_{s,i}(t) - E_s(t)) \quad (9)$$

$$\mu_i(t) = \mu_i(t_i) + \phi_i^2(t) \cdot g(\hat{\phi_s^2}(t)) \cdot ((1 - y_{s,i}(t)) - E_i(t)) \quad (10)$$

In these equations, we have

- $E_s(t) = E(\hat{\mu_s}(t), \hat{\mu_i}(t), \hat{\phi_i^2}(t))$,
- $E_i(t) = E(\hat{\mu_i}(t), \hat{\mu_s}(t), \hat{\phi_s^2}(t))$,
- $v_s^2(t) = \left[g(\hat{\phi_i^2}(t))^2 E_s(t)(1 - E_s(t))\right]^{-1}$, and
- $v_i^2(t) = \left[g(\hat{\phi_s^2}(t))^2 E_i(t)(1 - E_i(t))\right]^{-1}$.

Also, in equation (6), $p(a(t)|y_{s,i})$ is the marginal posterior density function for $a(t) = \log \sigma_s^2(t)$, approximated using the product of the following two normal density functions (here, $\varphi(z; m, \varsigma^2)$ denotes the normal density function with mean m and variance ς^2):

1. $\varphi(a(t); a(t_s), \tau^2)$, which comes from equation (3), and
2. $\varphi(\theta_s^*(t); \mu_s(t_s), \phi_s^2(t_s) + (t - t_s)e^{a(t)} + v_s^2(t))$, which is the normal approximation of the marginal likelihood distribution of $\theta_s(t)$, whose mode is denoted with $\theta_s^*(t)$.

The latter normal density function features the quantity $(\theta_s^*(t) - \mu_s(t_s))$, which is approximated in [6] using first-order Taylor expansion.

Finally, note that to prevent a rating deviation from becoming arbitrarily large, the quantity is constrained in equations (7) and (8) to never exceed the value for a brand new user/unit, just like how it was done in [5].

2.2 Initial Parameter Estimation

To address the stratification issue mentioned in the introduction, the user and unit ratings are differentially initialized based on their respective curricula. Instead of setting each curriculum's initial rating value arbitrarily, we want the values to reflect more closely our prior knowledge of the distributions of concepts within each curriculum.

We find this prior knowledge in our proprietary conceptual precedence graph, where units are represented as nodes (vertices) in a directed graph. Each edge (u, v) in the graph is interpreted as: "An instance of unit u is being used as a step in solving an instance of unit v." Hence unit u corresponds to a prerequisite concept that a user must have mastered before being able to successfully master unit v.

The key idea in our usage of the graph is that a question item (corresponding to a specific knowledge unit) that involves one or more steps to solve must in general be harder than any of the steps themselves. Hence we assign each unit with a non-negative integer value, which we call "depth," in such a way that for every edge, the tail node is assigned with a lower depth value than the head node. This way, a concept appearing in a higher grade level would in general correspond to a higher depth value (since they would generally incorporate lower-level curriculum concepts as prerequisites), making the depth values roughly signify how "in-depth" the corresponding concepts are. See Fig. 2 for an illustration.

We also seek to differentiate among units with no parents (i.e., concepts with no prerequisites) by imposing that the depth difference between a unit and its successor be as small in magnitude as possible, while still ensuring that every unit has a strictly greater depth value than any of its parents.

From a graph theory perspective, the problem of assigning depth values can be formulated as a variant of layer assignment problem on a directed acyclic graph $G = (V(G), E(G))$ with minimal dummy vertices, formally stated as the follow-

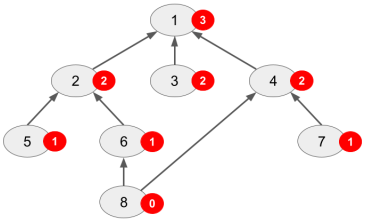

Figure 2: Illustration of assigning depth values to knowledge units in a simple conceptual precedence graph. Knowledge units are represented as nodes (gray ovals). On the right of each oval, a red circle shows the corresponding depth values assigned.

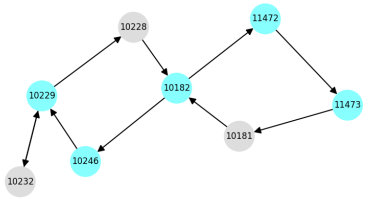

Figure 3: Three instances of simple cycles in the conceptual precedence graph used in our study, which all belong to one strongly connected component. We found that cycles exist mostly due to the presence of "gateway units" (shown in cyan ovals), whose main role is to select which concept to apply from multiple related concepts.

ing integer linear program (ILP):

$$\begin{aligned} \min \quad & \sum_{(u,v)\in E(G)} d(v) - d(u) & \\ \text{s.t.} \quad & d(v) - d(u) \geq 1 & \forall (u,v) \in E(G) \\ & d(v) \in \mathbb{Z}_{\geq 0} & \forall v \in V(G) \end{aligned} \tag{11}$$

(here, $d(v)$ denotes the depth value assigned to node v). For a general overview of the layer assignment problem and its variations, readers are referred to Section 13.3 of [9].

Two challenges arise in initializing rating values through solving the depth assignment problem. The first challenge is that our conceptual precedence graph could contain cycles, such as ones shown in Fig. 3. To address this challenge, we assign the same depth value to all units in the same strongly connected component (SCC), noting that any directed cycle is strongly connected. Implementationally, this corresponds to solving the ILP given in (11) on the conceptual precedence graph's *condensation*, which is a directed acyclic graph formed by contracting each SCC into one node.

The second challenge in assigning depths to nodes on the conceptual precedence graph is that the graph (and thus also its condensation) may consist of multiple weakly connected components (WCCs), which are subgraphs whose underlying undirected graphs are connected. The above ILP assigns depth values relative only to other SCCs in the same WCC, so additional steps must be taken to equate the depth value distributions for each curriculum across all WCCs. In particular, we label each SCC with the lowest-level curriculum that features at least one of its constituent units. Next, we take the smallest number of WCCs that together contain all curriculum labels. We call this collection of WCCs *reference WCCs*. Afterward, we offset the depth value for each SCC in every non-reference WCC to be at least the minimum depth value of all SCCs in the reference WCCs that are labeled with the same curriculum.

Once the adjusted depth values for all SCCs (and thereby all units) are thus computed, each curriculum's depth value is set to be the average depth value of all units in the curriculum.

Below is the summary of procedure for assigning depth $d(k)$ for each curriculum $k \in X = \{1, \ldots, K\}$:

1. Let $G = (V(G), E(G))$ be our conceptual precedence graph, which is a directed graph such that each node $v \in V(G)$ is associated with a curriculum $\chi(v) \in X$.
2. Condense G to yield a directed acyclic graph $C = (V(C), E(C))$.
3. Let $W_1, \ldots, W_n$ be WCCs of C, from largest to smallest.
4. For each $W_i = (V(W_i), E(W_i))$, solve the ILP given in (11) to yield pre-adjustment depth values $d_{init}(S)$ for each SCC S.
5. Label each SCC S with a curriculum
$$\chi_{min}(S) = \min_{v \in V(S)} \chi(v).$$
6. Let $A = \{W_1, \ldots, W_r\}$ be the reference WCCs (defined above), such that r is minimized; i.e., choose no more WCCs than necessary.
7. For each curriculum $k \in X$, let
$$d_{min}(k) = \min\{d(S) \mid \chi_{min}(S) = k, S \in \bigcup_{i=1}^{r} V(W_i)\}.$$
8. For each $W_j = W_{r+1}, \ldots, W_n$, adjust depth value $d(S)$ for each SCC $S \in W_j$ to be at least $d_{min}(\chi_{min}(S))$. However, do so in a way that the adjusted depth values still satisfy the constraints of the ILP given in (11).
9. We now have the adjusted depth values for every SCC $S \in V(C)$. For each SCC S, let $d(v) = d(S)$ for all $v \in S$.
10. For each $k \in X$, let
$$d(k) = \text{mean}\{d(v) \mid v \in V(G),\ \chi(v) = k\}.$$

We now give each user s or unit i associated with curriculum k as follows:

$$\mu_s(0) = \mu_{min} + \alpha \cdot d(k) \tag{12}$$

$$\mu_i(0) = \mu_{min} + \alpha \cdot d(k) \tag{13}$$

where quantities μ_{min} and α are hyperparameters to be optimized.

3. EVALUATION

We evaluate our model using a dataset consisting of student practice records from January 2016 to December 2019 through our adaptive software used in math learning centers located throughout the United States. Students are given problems to practice based on their current grade level and the content areas where they struggle. The data consists of 5,179,493 records of 10,194 users' combined attempts for problems associated with 7,513 knowledge units, ranging from Grade 2 concepts to Algebra 2 concepts. When a student gets a problem wrong in the first attempt, the student gets to make a second attempt for the same problem after being walked through the steps; in our analysis, however, only the first attempt's result was considered.

For the Glicko-2 model hyperparameters, we used the values suggested in [8]: 350.0 for the initial RD (in Glicko-1 scale; [8] shows how to convert between the two scales) and 0.06 for the initial user volatility. In the case of τ, for which a range of values is suggested, we used 0.5. The time elapsed from one attempt to the next, used in rating uncertainty inflation, is measured in days. Finally, through extensive simulations, we chose $\alpha \approx 0.2303$ and $\mu_{min} \approx -2.8782$, which, in Glicko-1 scale (on which the values were originally set), are exactly 40.0 and 1000.0, respectively.

Each unit's associated curriculum was based on the information provided in our content management system. For units appearing in multiple curricula, the earliest curriculum in the sequence was used. For users, due to the lack of availability of exact registration dates for all users at the time of the study, each user's curriculum was set as the curriculum associated with the first unit attempted by the user. The initial parameters for both users and units were then set following the procedure described previously.

3.1 Predictive Performance

To assess the predictive performance of our adaptation of the Glicko-2 rating system, we plotted the change in RMSE values for every 1,000 records over time (for the rationale behind the metric choice, see [14]). As the latent trait estimates are calibrated based on student practice records, we expect the RMSE across the entire system to decay over time. We see that this is exactly the case in Fig. 4, where the calibration curve for our model is also reported along with the reliability and resolution values.

We also report a convergent pattern in unit rating values and dynamically adjusting user rating values, analogous to the results obtained in [15], in Fig. 5.

3.2 Gradation of Unit Rating Distributions

We also plot the distributions of the final unit rating values for each curriculum. We expect that using a conceptual precedence graph to initialize rating values would cause the central tendencies of the rating distributions would show an upward trend as the curriculum level increases. As shown in Fig. 6, the final ratings computed without the graph-based rating initialization fail to show an upward trend in the mean rating values, whereas they do with the graph-based rating initialization. Also noteworthy is the complete disappearance of overlap in IQR between two curricula far

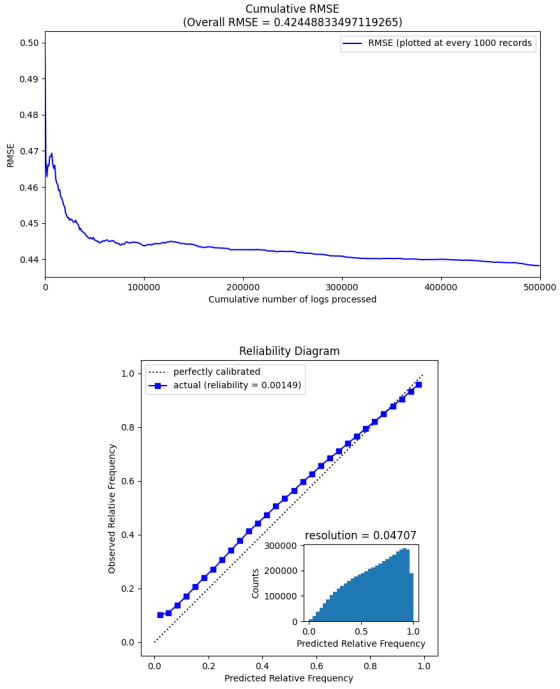

Figure 4: Top: Cumulative RMSE values calculated at every 1,000 records. For effective visualization, only results from the first 500,000 records were plotted. Bottom: Reliability diagram with sharpness graph inserted in the lower right.

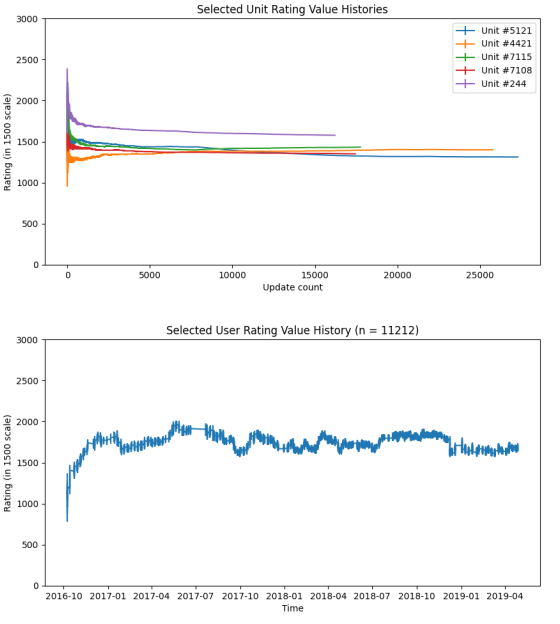

Figure 5: Rating values as a function of time for 5 most frequently attempted units (top) and for the user with the most number of attempts (bottom).

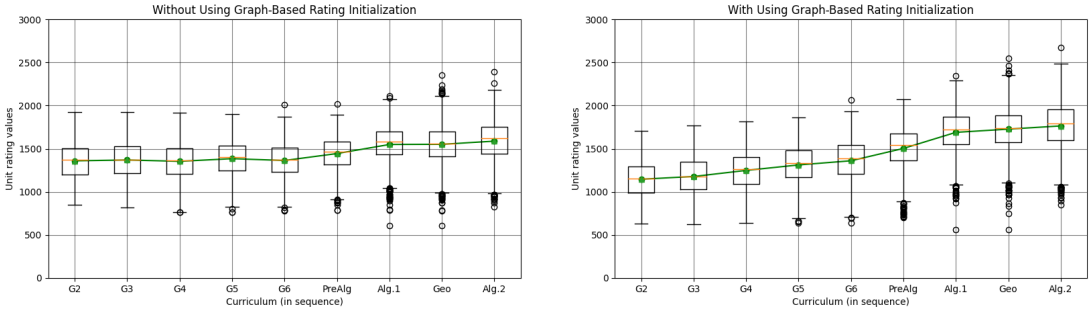

Figure 6: Final distributions of knowledge unit ratings. Orange bars indicate medians, green dots indicate means. Note that the rating values on the vertical axis are on the Glicko-1 scale.

apart from each other, such as Grade 2 and Algebra 2, upon using a conceptual precedence graph to initialize ratings.

4. DISCUSSION

We have used conceptual prerequisite relationships to give our model a better prior distribution— one that better reflects the stratified nature of student practice data. The depth values used to calculate the initial rating values, however, are still quite coarse estimates; for example, the difference in difficulty between a unit and one of its prerequisite units may not be even across the conceptual precedence graph. Nevertheless, we see that the distribution of the lowest-level curriculum (Grade 2 in our study) and that of the highest-level one (Algebra 2 in our study) show a substantially little overlap compared to when we used the initialization method of the original Glicko-2 system, which suggests that there was still a nontrivial improvement. Note that the separation of unit rating distributions between two adjacent curricula (for example, Grade 2 and Grade 3) are not well separated. This is expected, as we would not expect a huge jump in terms of curriculum difficulty from one school year to the next.

One interesting area of application of this framework is determining the appropriate grade level for students whose mathematical achievement levels are substantially ahead or behind their grade levels. With estimates of item difficulties that account for grade-level hierarchy, we can have a data-based justification that would allow gifted students to be placed at a higher-level curriculum that is neither too hard nor too easy for them. Likewise, we could allow for students lagging behind their peers to be placed at a lower-level curriculum, where they could ensure that their foundational understanding of lower-level mathematical concepts is firm before moving onto the next grade level. For this application, a separate round of validation with external measurements, e.g., standardized test scores, must first take place.

A well-known limitation of using the Glicko rating system family for educational applications is its inability to model multiple-choice item correctness probabilities. This is because the correctness probability of such an item has an infimum strictly greater than 0, making the corresponding probability distribution improper. Hence a natural future direction would be to address this limitation, e.g., by incorporating the particle-based method presented in [12].

Another potential threat to the validity of using the Glicko-2 model for student ability measurement is in its unidimensionality assumption. Part of the challenge of verifying whether the student response data can be modeled with a one-dimensional construct in a learning setting is that unlike in IRT settings, a student's ability is expected to change throughout the data collection period. An interesting future direction would be to investigate whether there is sufficient evidence to suggest that students' mathematical ability is multidimensional, and if so, how a model like the Glicko-2 rating system can be extended to reflect the multidimensionality; the degree to which the extension presented in [1] can be applied also remains to be seen.

Also, when assigning each curriculum with a depth value, the average depth values for all constituent units were calculated. In practice, however, as learning software product continues to expand, units can be added or removed, or their edge connections may change. Our current choice of taking an average makes the algorithm sensitive to changes in the conceptual precedence graph's internal connectivity structure. Median may be a more robust, and thus more practical, choice, though this may come at the risk of decreased differentiability across consecutive curricula.

5. CONCLUSION

We have presented an adaptation of the Glicko-2 rating system in a K-12 math learning software context. The stratified nature of student-item pairings has made effective discrimination of students and problems across grade levels challenging. We have shown evidence that by using the prerequisite relationships between concepts to initialize rating values, we can allow for the gradation of rating distributions from lower-level curriculum to the higher-level curriculum while ensuring that the prediction error for student response correctness still decreases over time. A potential area of application is for determining the grade level appropriate for students substantially ahead or behind their peers.

6. ACKNOWLEDGEMENT

I thank my boss Kurt Cho, who gave nudges in productive directions whenever I was stuck, and my colleagues Sunghwan Cho and Seunghun Lee, who worked on developing data pipeline infrastructure on which the proposed model can be deployed. Also, I thank everyone in my company, who patiently waited in support while the project was in the works.

7. REFERENCES

[1] L. Cai. Potential applications of latent variable modeling for the psychometrics of medical simulation. *Military Medicine*, 178(suppl_10):115–120, 2013.

[2] M. C. Desmarais and R. S. J. d. Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22(1-2):9–38, 2012.

[3] J. S. Eccles. Expectancies, values and academic behaviors. *Achievement and achievement motives*, pages 74–146, 1983.

[4] A. E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Publishing, 1978.

[5] M. E. Glickman. The Glicko system. `http://www.glicko.net/glicko/glicko.pdf`.

[6] M. E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

[7] M. E. Glickman. Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics*, 28(6):673–689, 2001.

[8] M. E. Glickman. Example of the Glicko-2 system. `http://www.glicko.net/glicko/glicko2.pdf`, 2013.

[9] P. Healy and N. Nikolov. *Hierarchical Drawing Algorithms*, pages 409–454. 08 2013.

[10] C. S. Hulleman, K. E. Barron, J. J. Kosovich, and R. A. Lazowski. Student motivation: Current theories, constructs, and interventions within an expectancy-value framework. In *Psychosocial Skills and School Systems in the 21st Century*, pages 241–278. Springer, 2016.

[11] S. Klinkenberg, M. Straatemeier, and H. L. van der Maas. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824, 2011.

[12] J. Nižnan, R. Pelánek, and J. Rihák. Student models for prior knowledge estimation. *International Educational Data Mining Society*, 2015.

[13] J. Papousek, R. Pelánek, and V. Stanislav. Adaptive practice of facts in domains with varied prior knowledge. In *Educational Data Mining 2014*, 2014.

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

[15] R. Reddick. Using a Glicko-based algorithm to measure in-course learning. *International Educational Data Mining Society*, 2019.

[16] S. Sampayo-Vargas, C. J. Cope, Z. He, and G. J. Byrne. The effectiveness of adaptive difficulty adjustments on students' motivation and learning in an educational computer game. *Computers & Education*, 69:452–462, 2013.