Elo, I Love You Won't You Tell Me Your K

Michael Yudelson[0000−0002−8280−0348]

ACT, Inc., Iowa City, IA 52243, USA michael.yudelson@act.org

Absrtract Elo is a rating schema used for tracking player level in individual and, sometimes, team sports, most notably – in chess. Also, it has found use in the area of tracking learner proficiency. Similar to the 1PL IRT (Rasch), Elo rating schema could be extended to serve the most demanding needs of learner skill tracking. Elo's advantage is that it has fewer parameters. However, the computational efficiency side of the search for the best-fitting values of these parameters is rarely discussed. In this paper, we are focusing on questions of implementing Elo and a gradient-based approach to find optimal values of its parameters. Also, we compare several variants of Elo to learning modeling approaches like Bayesian Knowledge Tracing. Our results show that the use of analytical gradients results in computational and, sometimes, statistical fit improvements on small and large datasets alike.

Keywords: Modeling Student Learning, Model Comparison, Elo rating schema.

1 Introduction

Computer-assisted testing and computer-guided learning rely on computational models of student knowledge and learning to produce personalized value for test-takers and learners. Models like 1PL IRT [11] and Log-Linear Test Model (LLTM) [21] were used and elaborated upon by the measurement community to compile test forms and compute student test scores. The field of computer-guided learning, most notably, intelligent tutoring systems, long relied on Bayesian Knowledge Tracing (BKT) [2] model for operational student-modeling or an approach in ASSISTments where three corrects in a row earn the student skill mastery [4]. Among the analytical models of learning that were used extensively, we could mention the Additive Factors Model (AFM) [1].

Elo recently rediscovered by learning analytics and educational data mining communities and several research investigations were published. While Elo is different from statistical models traditionally used in assessment and learning (often referred to as rating schema, not a model), it has highly desirable properties for these fields. First, Elo is designed to completely sidestep cold start problem and doesn't require substantial tuning (fitting) – known Elo variants all have under a dozen parameters. Second, Elo relies on local, often, asynchronous updates and that resonates well with computational issues assessment and learning models often have to combat with. Third, Elo is intuitively explainable – wrong

answer results in a decrement of student's ability ratings and vice versa, plus, the more unexpected the outcome, the more the update to the rating value is.

One of the shortcomings of Elo is that parameter fitting is largely done by hand-picking or grid search [13]. Our attempts to find traces of attempts to address Elo's parameter optimization resulted in no reference from the fields of assessment or learning. The only publication we found was from the field of biology where Elo was used for explaining behaviors of primates [5]. In this paper, we attempt to address the fitting of Elo parameters when applied to educational data and to work out analytical gradients for two forms of Elo rating schema. We then fit Elo on several sets of publicly available learning data and show that the use of analytical gradients follows the results when gradients are computationally approximated. Also, often, a computational improvement is observed.

2 Prior Work and Uses of Elo

Elo rating schema has long been used to rate chess players. In addition, Elo is also used for rating players in multiplayer competition in several video games [22], association football, American football, basketball [15], Major League Baseball, tennis [6], Scrabble, and other games. A Bayesian approach, based on Elo called TrueSkill^{TM} was developed by Herbrich and colleagues [7] to address performance in team sports. In biology, Elo has found use to explain the formation of dominance hierarchies of primates [5].

In education, there are several cases of successful use of Elo both as a theoretical and operational model. For example, members of Pelánek's research group published several works where variants of Elo rating schema were used in connection to learning Geography, specifically to track student recall of the shapes of maps of the Northern European countries [13]. One of the most at-scale operational uses of Elo rating schema in education is in the system Math Garden [8] that is widely used in a K-12 setting in the Netherlands. An Elo-based system of student ratings was used by Ivanovo State Power University, Russia to track student progress as they complete the courses overall, as well as the intermediate and partial exams within the courses [9], [24]. This approach called Developing Individual Creating Thinking (RITM in Russian transliteration) was implemented in 1992 and is still in use today.

3 Elo Rating Schema

Elo is a rating schema named after its inventor Arpad Elo [3]. In chess, where Elo found initial use, the modeled events are chess matches and the variables are opponent 1 ability and opponent 2 ability. After each match, the ratings of opponents' abilities are updated based on the outcome (a win of either opponent or a draw). In the fields of measurement and learning, an event is the student's opportunity to answer a question item correctly. The student is opponent 1, and the item is opponent 2. Sometimes, a set of skills relevant to the question item are used to collectively represent opponent 2. When applied to tracking learner

proficiency a standard version of Elo is often compared to a Rasch model that used in psychometrics. We will start by describing the Rasch model first and then focus on Elo.

3.1 Rasch Model

Rasch model [11], also known as 1PL IRT, captures test-taker performance with the help of two classes of variables: unidimensional abilities of test-takers, and unidimensional difficulties of test items. Both abilities and difficulties are thought of as stationary values that do not change over the time of assessment. Refer to Equation 1 and 2 for the formulation of the Rasch model. Here, θ_i – is the ability of student i, β_j – is the difficulty of item j, X_{ij} – is i^{th} student's response to item j, p_{ij} – is the estimate of the probability of student answering the item correctly, and m_{ij} – is the log-odds value of that probability.

$$p_{ij} = Pr(X_{ij} = 1) = \sigma(m_{ij}) = \frac{1}{1 + e^{-m_{ij}}} \tag{1}$$

$$m_{ij} = \theta_i - \beta_j \tag{2}$$

3.2 Student-Item Elo

A simple formulation of Elo capturing students and items is given in Equation 3. It is related to the Rasch model's formulation in Equation 1. In Elo, s_i – is the current logit rating of student's unidimensional ability and b_j – is the current logit rating of item's unidimensional difficulty. We are only defining Elo's m_{ij}, since the probabilistic form is the same as shown in Equation 1.

$$m_{ij} = s_i - b_j \tag{3}$$

If we are to draw comparisons between Rasch's θ_i and β_j and Elo's s_i and b_j, the former would be stationary values and the latter would be functions of time since, in Elo, s_i and b_j are incrementally updated as new data arrives. One may hypothesize that say, s_i could be asymptotically approaching θ_i. However, unlike θ_i, the distribution of s_i has not been theoretically described and s_i constantly changes which complicates such theoretical description. The same is true for b_j. Additionally, in the Rasch model, θ_i and β_j are parameters, while s_i and b_j in Elo are not. In some literature, for example [14], Elo-tracked student abilities and item difficulties are written as θ_i and β_j. However, in order to separate the meanings, we would use different notation.

As mentioned before, tracked Elo values are updated as new data points are observed. Refer to Equations 4-5 for the updating rules. Here, K is a sensitivity parameter controlling the magnitude of the update. Thus, the Elo variant as in Equation 3 has one parameter K. We will refer to this Elo version as **E1**.

$$s_i = \begin{cases} 0, \text{ if this is the first time we see data of student } i \\ s_i + K \cdot (X_{ij} - p_{ij}), \text{ otherwise} \end{cases} \tag{4}$$

$$b_j = \begin{cases} 0, \text{ if this is the first time we see data of item } j \\ b_j - K \cdot (X_{ij} - p_{ij}), \text{ otherwise} \end{cases} \tag{5}$$

We could modify the previously defined Student-Item Elo model by defining two sensitivity parameters: ${}_iK$ for updating student abilities, and ${}_jK$ for updating item difficulties. Here, the ${}_i$ nd ${}_j$ mean that the corresponding K values belong to student updates and item updates respectively. The corresponding changes are shown in Equations 6 and 7. This version of Elo we will call **E2**.

$$s_i = s_i + {}_iK \cdot (X_{ij} - p_{ij}) \tag{6}$$

$$b_j = b_j - {}_jK \cdot (X_{ij} - p_{ij}) \tag{7}$$

4 Gradients of Elo Parameters

4.1 Preliminary Definitions

We use $O = \{o_t\}$, to denote observations, where $o_t \in \{0, 1\}$ is the student's response to an item at some time t. Here, $t \in [1, T]$ is the time slice and it indexes the data of all students answering all items sorted by time. 0 and 1 denote incorrect and correct student responses respectively. Vector of Elo parameters is denoted as λ. An element of the vector is λ_m, where $m \in [1, M]$ and $\lambda_m \in (-\infty, +\infty)$.

We will be using maximum-likelihood estimation in our further work. For optimization, we are going to rely on negative total log-likelihood of data given parameters and will try to minimize that value. Total negative log-likelihood denoted as J is defined in Equation 8. In simple terms, the total likelihood of the data is the product of the probabilities of the actual observations given the parameters of Elo. Negative log-likelihood is the negative sum of the logarithms of the probabilities of actual observations. Here, p_t – is the probability (expected value) of the observation being the correct response at time t and is equivalent to p_{ij} in Equation 1. Also, m_t – a logit form of the expected performance – would be equivalent to m_{ij} from Equation 3.

$$J = -ln(L_{tot}) = -\sum_{t=1}^{T} (o_t ln(p_t) + (1 - o_t) ln(1 - p_t)) \tag{8}$$

4.2 General Partial Derivative

Partial derivative of J with respect to λ_m assumes the form shown in Equation 9. Depending on how m_t is defined in a particular variant of Elo, the $\partial m_t / \partial \lambda_m$ would change. As a simplification, we would write $o_t - \sigma(m_t)$ or $o_t - p_t$ as δ_t – the prediction error at time t and rewrite Equation 9 as shown in Equation 10.

$$\begin{aligned}
\frac{\partial J}{\partial \lambda_m} &= -\sum_{t=1}^{T} \left(\frac{o_t}{p_t} \frac{\partial p_t}{\partial \lambda_m} - \frac{1-o_t}{1-p_t} \frac{\partial p_t}{\partial \lambda_m} \right) \\
&= -\sum_{t=1}^{T} \left(\left[\frac{o_t}{p_t} - \frac{1-o_t}{1-p_t} \right] \frac{\partial p_t}{\partial \lambda_m} \right) \\
&= -\sum_{t=1}^{T} \left(\frac{o_t - p_t}{p_t(1-p_t)} \frac{\partial p_t}{\partial \lambda_m} \right) \\
using \frac{\partial p_t}{\partial \lambda_m} &= \frac{\partial \sigma(m_t)}{\partial \lambda_m} = \sigma(m_t)\left(1-\sigma(m_t)\right) \frac{\partial m_t}{\partial \lambda_m} \\
&= -\sum_{t=1}^{T} \left(\frac{o_t - \sigma(m_t)}{\sigma(m_t)(1-\sigma(m_t))} \sigma(m_t)\left(1-\sigma(m_t)\right) \frac{\partial m_t}{\partial \lambda_m} \right) \\
&= -\sum_{t=1}^{T} \left(o_t - \sigma(m_t)\right) \frac{\partial m_t}{\partial \lambda_m} \qquad (9)
\end{aligned}$$

$$\frac{\partial J}{\partial \lambda_m} = -\sum_{t=1}^{T} \delta_t \frac{\partial m_t}{\partial \lambda_m} \qquad (10)$$

4.3 Detailed Partial Derivatives

The Elo variant **E1** accounts for unidimensional student ability s_i and unidimensional item difficulty b_j. In order to bridge the notation defining Elo in Equations 4–7 to indexing data by time slice t, we define functions $g_i(t)$ and $g_j(t)$ that, for a given data point t produce the respective student and item indexes i and j.

Let's now define how the data points of the same student or item are counted. Function $c_i(t)$ and function $c_j(t)$ produce the count of data points before time t belonging to, respectively, student i and item j. Let's also define indexing functions $r_i(t)$ and $r_j(t)$ that, for a data point t, gives the time slice of the data point when a student or an item were seen last. Thus, for example, $r_i(t) < t$ is the prior data point corresponding to student $g_i(t)$. Refer to the first eight columns of Table 1 for an example that covers all of the indexes we talked about thus far. There, t, $g_i(t)$, and $g_j(t)$ – are given; the rest – follow from the definitions.

Given the above definitions, for Elo variant **E1** (simplest student-item Elo) the expected logit-scale value of student's performance is given in Equation 11a. Note that the expected value is defined by using prior estimates of student ability s_i and item difficulty b_j. The initial values of student ability and item difficulty are given in Equation 11c and Equation 11d for the top cases when the respective opportunity counts are 0's.

The rules of updating s_i and b_j upon processing data point t in the bottom cases of Equation 11c and Equation 11d, where the respective $c_\bullet$ counts are non-zero. Computation of the the gradient of the negative log-likelihood of the data given sensitivity K is in Equation 11e. An example of updating rating and gradient values for Student-Item Single Sensitivity Elo based on is in Table 1 in

$$
\begin{aligned}
i &= g_i(t), \text{ index of student for row } t \\
j &= g_j(t), \text{ index of item for row } t \\
r_i(l) &= r_i(g_i(l)), \text{ time student } i \text{ was seen prior to time } l \\
r_j(l) &= r_j(g_j(l)), \text{ time item } j \text{ was seenprior to time } l \\
c_i &= c_i(g_i(l)), \text{ count of times student } i \text{ seen prior to time } l \\
c_j &= c_j(g_j(l)), \text{ count of times item } j \text{ seen prior to time } l \\
m_t &= s_i - b_j && \text{(11a)} \\
\delta_t &= o_t - \sigma(m_t) && \text{(11b)} \\
s_i &= \begin{cases} 0 & \text{if } c_i = 0 \\ s_i + K \cdot \delta_t & \text{if } c_i > 0 \end{cases} && \text{(11c)} \\
b_j &= \begin{cases} 0 & \text{if } c_j = 0 \\ b_j - K \cdot \delta_t & \text{if } c_j > 0 \end{cases} && \text{(11d)} \\
\frac{\partial J}{\partial K} &= -\sum_{t=1}^{T} \delta_t \cdot \sum_{l=1}^{t} \left[(c_i > 0) \cdot \delta_{r_i(l)} + (c_j > 0) \cdot \delta_{r_j(l)} \right] && \text{(11e)}
\end{aligned}
$$

columns 9 through 19. If we are using Elo variant E2, and, instead of a single sensitivity K for updating tracking values for both students and items, we were to use separate sensitivities – ${}_iK$ for students and ${}_jK$ for items, the gradients would be as shown in Equations 12a–12d.

$$
\begin{aligned}
s_i &= \begin{cases} 0 & \text{if } c_i = 0 \\ s_i + {}_iK \cdot \delta_t & \text{if } c_i > 0 \end{cases} && \text{(12a)} \\
b_j &= \begin{cases} 0 & \text{if } c_j = 0 \\ b_j - {}_jK \cdot \delta_t & \text{if } c_j > 0 \end{cases} && \text{(12b)} \\
\frac{\partial J}{\partial_i K} &= -\sum_{t=1}^{T} \delta_t \cdot \sum_{l=1}^{t} \left[(c_i > 0) \cdot \delta_{r_i(l)} \right] && \text{(12c)} \\
\frac{\partial J}{\partial_j K} &= -\sum_{t=1}^{T} \delta_t \cdot \sum_{l=1}^{t} \left[(c_j > 0) \cdot \delta_{r_j(l)} \right] && \text{(12d)}
\end{aligned}
$$

5 Computational Validation

In order to give the analytical gradients of the described versions of the Elo rating schema approach, we have made comparative runs of Elo schema fitting

Table 1. An example of updating ratings and computing gradient of Student-Item-Single Sensitivity Elo where K=0.4. The total log-likelihood (the sum of J_t's) J = 5.768, and the gradient of the K is 0.777.

t	o_t	$s = g_i(t)$	$i = g_j(t)$	$c_i(t)$	$c_j(t)$	$r_i(t)$	$r_j(t)$	s_1	s_2	s_3	b_1	b_2	b_3	p_t	J_t	$\frac{\partial J_t}{\partial K}$
0								0.000	0.000	0.000	0.000	0.000	0.000			
1	0	1	1	0	0	0	0	-0.200			0.200			0.500	0.693	0.000
2	0	1	2	1	0	1	0	-0.380				0.180		0.450	0.598	-0.225
3	1	2	1	0	1	0	1		0.220		-0.020			0.450	0.798	0.275
4	0	2	2	1	1	3	2		0.016			0.384		0.510	0.713	0.051
5	0	1	3	2	0	2	0	-0.543					0.162	0.406	0.521	-0.386
6	1	1	3	3	1	5	5	-0.275					-0.105	0.331	1.106	1.180
7	0	3	1	0	2	0	3			-0.202	0.182			0.505	0.703	0.025
8	1	2	3	2	2	4	6		0.204				-0.293	0.530	0.634	-0.142

procedures. We relied on `R` statistical package and its base function `optim` that implements the BFGS algorithm [12]. Instead of the BFGS algorithm, we could have chosen gradient descent or conjugate gradient descent or any other method that would rely on log-likelihood and gradient. Instead of focusing on *the* algorithm, we picked *an* algorithm and controlled for that choice.

For all versions of the Elo, we implemented objective functions computing negative log-likelihood from the data given the parameter(s) and the gradient of the parameters. Since `optim` function relies on natively compiled code written in `C/C++`, we implemented the objective function (negative log-likelihood) and gradient computations in `C/C++` as well. Thus, the relative speeds of the core BFGS algorithm and the functions are comparable.

BFGS algorithm implemented in `optim` function could run with approximated gradients relying on the objective function alone or with the supplied gradient function. For each test case to be discussed below, we recorded the resulting negative log-likelihood, fit metrics, time, and the number of iterations it took the parameter fitting to complete. In all runs, the sensitivity parameter(s) K were seeded to 0.4.

To better position the results within the relevant literature, we compared the performance of the Elo models in question to Bayesian Knowledge Tracing (BKT) model. Since all of the data we will use comes from the Carnegie Learning Cognitive Tutor that relies on BKT, the choice is natural. To fit BKT models we used a package `hmm-scalable` [23] written in `C/C++` that is known to be efficient in dealing with large datasets of learning data.

6 Data

We used four datasets. Two are made available by LearnLab's LearnSphere repository [10] and two available as part of KDD Cup 2010 [19]. The first

LearnSphere dataset **D1** – Geometry Area (1996-97) – consists of 5,104 records belonging to 59 students working through a Geometry Area unit of Carnegie Learning Cognitive Tutor. Students there were interacting with 139 distinct items (problem steps).

The second LearnSphere dataset **D2** [18] has 128,493 rows belonging to 123 students working with a Geometry Area unit of Carnegie Learning Cognitive Tutor. Here, students were interacting with 16,485 distinct items (problem steps). The third dataset **D3** [16] has Carnegie Learning's Cognitive Tutor data collected in the 2008-2009 school year in Algebra I classrooms. This dataset had 8,918,055 transactions of 3,310 students working with 206,596 items (problems). Finally, the fourth dataset **D4** [17] has Carnegie Learning's Cognitive Tutor data collected in the 2008-2009 school year in Bridge to Algebra classrooms. This dataset had 20,012,499 transactions of 6,043 students working with 61,848 items (problems).

One could see that we used problem steps as items in datasets **D1** and **D2**, but problems as items in datasets **D3** and **D4**. There is a much larger ratio of unique problem steps to data points in the latter case and that is why we resorted to using problems. Even after the adjustment, the resulting item per datapoint ratios are rather different – 36.72, 7.79, 43.17, and 323.58 for datasets **D1**, **D2**, **D3**, and **D4** respectively. A different problem step to datapoint ratio is due to a greater variety of content units in datasets **D3** and **D4** that cover the whole year, while datasets **D1** and **D2** only cover one section of content.

7 Results

Table 2 is a summary of the comparative runs of fitting the two versions of the Elo rating schema and one regular BKT model to each dataset. The table is ordered by the dataset (**D1**, **D2**, **D3**, and **D4**), the Elo version (**E1** and **E2**), and BKT model comes after Elo models for every dataset.

The first thing to note is that both the negative log-likelihood and the reached parameter values are quite close across all 8 pairwise comparisons. The same is especially true for statistical fitness metrics – accuracy and RMSE – the difference is always in the third or fourth decimal digit. The second thing we can note is that the use of the analytical gradient results in longer run time for datasets **D1** and **D2**, but shorter time run for the datasets **D3** and **D4**. This could be due to the effect of the size – larger datasets do not incur as much relative computational overhead. When dividing the overall run time by the number of iterations[1] the relative speed of the analytical gradients is consistently higher.

If we look at single vs. double sensitivity Elo, we notice that, in terms of the negative log-likelihood, a 2-sensitivity model has a slight edge. However, in terms of fit metrics – accuracy and RMSE – the differences aren't so pronounced. In terms of time, not surprisingly, the 2-sensitivity Elo takes longer to fit.

[1] Since `optim` function does not output iterations explicitly, we have substituted iterations count with the sum of the number of times objective function and gradient were executed – both required a pass over the dataset.

Table 2. Comprative performace of approximated and analytical gradients when fitting the two Elo variants and BKT.

Model	Data	Grad.-s	Neg. LL	RMSE	Acc.	Param.(s)	Iter.	Tm., s	Tm./It.
E1	D1	approx.	2639	0.4139	0.7453	0.3583	19	0.022	0.0011
E1	D1	analyt.	2640	0.4140	0.7467	0.3701	60	0.035	0.0006
E2	D1	approx.	2634	0.4137	0.7443	0.2619, 0.4427	25	0.029	0.0012
E2	D1	analyt.	2634	0.4138	0.7437	0.2603, 0.4717	76	0.047	0.0006
BKT	D1	yes	2537	0.4034	0.7663	-	-	0.099	-
E1	D2	approx.	27930	0.2417	0.9299	1.0431	38	0.423	0.0111
E1	D2	analyt.	27957	0.2420	0.9298	0.9381	63	0.687	0.0109
E2	D2	approx.	27269	0.2412	0.9283	0.4128, 1.5169	50	0.738	0.0148
E2	D2	analyt.	27270	0.2411	0.9283	0.4188, 1.5333	137	1.339	0.0098
BKT	D2	yes	29921	0.2500	0.9291	-	-	0.504	-
E1	D3	approx.	3447761	0.3422	0.8538	0.1282	45	22.780	0.5062
E1	D3	analyt.	3450255	0.3422	0.8538	0.0986	49	17.404	0.3552
E2	D3	approx.	3437226	0.3417	0.8539	0.1965, 0.0340	72	40.827	0.5670
E2	D3	analyt.	3440697	0.3421	0.8540	0.1601, 0.0789	152	60.354	0.3971
BKT	D3	yes	3412619	0.3389	0.8572	-	-	46.237	-
E1	D4	approx.	7108867	0.3263	0.8653	0.1212	62	53.871	0.8689
E1	D4	analyt.	7108948	0.3263	0.8653	0.1171	47	38.136	0.8114
E2	D4	approx.	7101767	0.3261	0.8654	0.1697, 0.0734	77	98.708	1.2819
E2	D4	analyt.	7111965	0.3264	0.8652	0.1071, 0.1267	68	65.542	0.9638
BKT	D4	yes	6906909	0.3178	0.8722	-	-	110.052	-

Together with Elo performance, for every dataset, we included the performance of a fit BKT model. Across the four datasets, it is not possible to determine a clear winner. In some cases, BKT has the edge in terms of shorter running time but loses slightly on the accuracy. We were especially happy that Elo *holds its ground* well on the large datasets **D3** and **D4**.

8 Conclusions

In this paper, we have discussed an approach to finding optimal parameters for Elo rating schema using analytically derived gradients. To the best of our knowledge, this is the first attempt to derive analytical gradients for Elo and fit it as a machine learning model. We were primarily interested in the [relative] speed of the search for the best-fitting parameter and how close are the achieved log-likelihoods of the analytical and approximated gradient approaches. When comparing approximated and analytical gradients, it is expected to see differences in convergence and even statistical fit, the latter being of slightly elevated importance. The result we obtained should not be taken as a hard conclusion. In order to draw inferences, one should run series of cross-validations instead of a single fit of the modal to the whole dataset.

While we were fitting Elo parameters, we controlled for the kernel search algorithm – BFGS. Admittedly, different search algorithms (conjugate gradient

descent, Brent, L-BFGS, to name a few) could result in slightly better or worse performance. Although our brief experimentation with conjugate gradient descent did not show any difference in terms of run time and performance.

When it comes to a particular variant of Elo rating schema, we only considered student-item Elo with one or two constant sensitivity of the update (K). There exist far more complex and expressive variants of Elo (see, for example, [20] and [14]) where student tracked values are hierarchical and skill ratings are tracked instead of item ratings. Also, instead of the single sensitivity, authors sometimes use a form of an uncertainty function that diminishes the magnitude of the update to the rating as more data is used to re-compute it. Starting with the derivations in this paper, the analytical gradient approach we presented could be used to formalize those Elo variants as well.

A worked-out analytical gradient for a variant of Elo could be useful in several ways. One might think of an extension where each student receives an individualized weight (say, a multiplier) to go with the sensitivity parameter. Having worked out an analytical gradient, one might regularise these individual weights treating them as a random factor. Of course, individualized weights would have to change as a function of time just as student abilities and item difficulties do in Elo.

Also, Elo functionality could be employed for infusing the self-adjusting nature of tracked ratings onto other models. For example, an iBKT model [23] is not operationalizable to this day since student-level parameters need to be re-fit frequently using a lot of data. Treating student-level features as ratings updated using Elo-like procedure could make such Elo-infused iBKT operationalizable by definition.

References

1. Cen, H., Koedinger, K., Junker, B.: Comparing two irt models for conjunctive skills. In: International Conference on Intelligent Tutoring Systems. pp. 796–798. Springer (2008)
2. Corbett, A.T., Anderson, J.R.: Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction **4**(4), 253–278 (1994)
3. Elo, A.E.: The rating of chessplayers, past and present. Arco Publishers (1978)
4. Feng, M., Heffernan, N., Koedinger, K.: Addressing the assessment challenge with an online system that tutors as it assesses. User Modeling and User-Adapted Interaction **19**(3), 243–266 (2009)
5. Franz, M., McLean, E., Tung, J., Altmann, J., Alberts, S.: Self-organizing dominance hierarchies in a wild primate population. Proceedings of the Royal Society B: Biological Sciences **282**(1814), 20151512 (2015)
6. Glickman, M.E.: Parameter estimation in large dynamic paired comparison experiments. Journal of the Royal Statistical Society: Series C (Applied Statistics) **48**(3), 377–394 (1999)
7. Herbrich, R., Minka, T., Graepel, T.: Trueskill^{TM}: a bayesian skill rating system. In: Advances in neural information processing systems. pp. 569–576 (2007)

8. Hofman, A., Jansen, B., de Mooij, S., Stevenson, C., van der Maas, H.: A solution to the measurement problem in the idiographic approach using computer adaptive practicing. Journal of Intelligence **6**(1), 14 (2018)
9. Ivanovo State Power University: Ritm-rating. a system of tracking student ratings. http://ritm.ispu.ru/old/help
10. Koedinger, K., Baker, R., Cunningham, K., Skogsholm, A., Leber, B., Stamper, J.: A Data Repository for the EDM community: The PSLC DataShop. CRC Press (2010)
11. Van der Linden, W., Hambleton, R.: Handbook of Modern Item Response Theory. Springer, New York. (1997)
12. Nash, J.C.: Compact numerical methods for computers: linear algebra and function minimisation. CRC press (1990)
13. Nižnan, J., Pelánek, R., Rihák, J.: Student models for prior knowledge estimation. In: Proceedings of the 8th International Conference on Educational Data Mining. pp. 109–116. ACM, New York, NY, USA (2015)
14. Pelánek, R.: Applications of the elo rating system in adaptive educational systems. Computers & Education **98**, 169–179 (2016)
15. Silver, N., Fischer-Baum, R.: How we calculate nba elo ratings. Archived from the original on May 21, 2015
16. Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G., Koedinger, K.: Algebra i 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge. http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp (2010)
17. Stamper, J., Niculescu-Mizil, A., Ritter, S., Gordon, G., Koedinger, K.: Bridge to algebra 2008-2009. challenge data set from kdd cup 2010 educational data mining challenge. http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp (2010)
18. Stamper, J., Koedinger, K.: Human-machine student model discovery and improvement using data. In: Proceedings of the 15th International Conference on Artificial Intelligence in Education. pp. 353–360. Springer, Berlin (2011)
19. Stamper, J., Pardos, Z.A.: The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics. Journal of Learning Analytics **3**(2), 312–316 (2016)
20. Von Davier, A., Deonovic, B., Polyak, S., Woo, A.: Applications of the elo rating system in adaptive educational systems. Fronteers in Psychology **(in press)** (2019)
21. Wilson, M., De Boeck, P.: Descriptive and explanatory item response models. Springer-Verlag (2004)
22. Wood, B.: Enemydown uses elo in its counterstrike:source multilplayer ladders. Archived from the original on June 12, 2009
23. Yudelson, M.V., Koedinger, K.R., Gordon, G.J.: Individualized bayesian knowledge tracing models. In: In: Lane, H.C., and Yacef, K., Mostow, J., Pavlik, P.I. (eds.) Proceedings of 16th International Conference on Artificial Intelligence in Education (AIED 2013). pp. 171–180. Springer-Verlag, Berlin-Heidelberg (2013)
24. Нуждин, В. Н., Шишкин, В. П.: РИТМ в вопросах и ответах. Министерство науки, высшей школы и технической политики Российской Федерации, Комитет по высшей школе, Ивановский энергетический институт, Иваново. (1992)