# Leveraging Skill Hierarchy for Multi-Level Modeling with Elo Rating System

**Michael Yudelson**
ACT, Inc.
Iowa City, IA
michael.yudelson@act.org

**Yigal Rosen**
ACT, Inc.
Iowa City, IA
yigal.rosen@act.org

**Steve Polyak**
ACT, Inc.
Iowa City, IA
steve.polyak@act.org

**Jimmy de la Torre**
University of Hong Kong.
Hong Kong
j.delatorre@hku.hk

## ABSTRACT

In this paper, we are discussing the case of offering retired assessment items as practice problems for the purposes of learning in a system called ACT Academy. In contrast to computer-assisted learning platforms, where students consistently focus on small sets of skills they practice till mastery, in our case, students are free to explore the whole subject domain. As a result, they have significantly lower attempt counts per individual skill.

We have developed and evaluated a student modeling approach that differs from traditional approaches to modeling skill acquisition by leveraging the hierarchical relations in the skill taxonomy used for indexing practice problems. Results show that when applied in systems like ACT Academy, this approach offers significant improvements in terms of predicting student performance.

### Author Keywords

Online learning; assessment; skill acquisition; multi-level modeling.

### ACM Classification Keywords

J.1. Education; K.3.1. Computer-assisted instruction (CAI); I.2.6. Knowledge acquisition; I.2.6. Parameter learning.

## INTRODUCTION

Computer-based and computer-assisted educational systems have penetrated our daily lives. Computer-based delivery of one-size fits all content, assessment, or learning is no longer an option. Many fields of study are focusing on approaches to efficiently represent student knowledge. The central issues for these approaches are the accuracy and validity of the estimates. As the demand for the volume and the quality of the learning and assessment content is met, the question of serving the most appropriate content is crucial. The earlier we know the needs of the user, the earlier we can start helping them more efficiently.



A student that just started using an adaptive problem practice technology has to commit to it for a significant amount of time before their user model saturates and stabilizes. However, not all students can stay long enough for the algorithms to get over the cold start problem. In some systems, students tend to skim and the evidence of student knowledge spreads thinly across the domain.

We present an approach to saturating the diagnostic student profiles faster by utilizing the hierarchy of skills that practice problems are indexed with. We have obtained performance data of students working with a free to use ACT Academy that lets students practice their knowledge with retired ACT Test items. ACT Academy is already backed up by a diagnostic model that is the basis for the supplementary content recommendation. We have devised a series of computational experiments to determine whether, in the absence of the prior estimate of student's mastery of a skill, it is possible to abstract to a parent, coarser-grained skill in order to make a better than default judgment about student's performance. By modeling student skill at all levels of the skill hierarchy we show that it is possible to improve the accuracy of predictive modeling statistically significantly.

## ACT ACADEMY AND RAD API

The ACT offers a free practice and assessment platform, ACT Academy (www.act.org/academy), that helps students review the skills that are assessed on the ACT college readiness assessment. ACT Academy delivers of short 5-10 item practice quizzes as well as full-length practice tests that users can select in a self-directed manner. Test items on the quizzes are quality ACT Test items from previous years as well as items licensed from ACT partners. ACT Academy was launched in March 2018 and since then almost 100,000 students registered and gave it a try. In addition to self-assessment questions, ACT Academy offers curated text, video, and interactive resources on the topics covered in quizzes. Students can access these resources after completing and reviewing their mini-quiz results.

ACT Academy's is supported by Recommendation and Diagnostics (RAD) API. RAD API processes student responses to quiz items, maintains a diagnostic model of the learners and uses that data to generate personalized recommendations of remedial. RAD API receives reports of student performance and merges scores students received with question item metadata. After every question attempt, relevant student skills addressed in that question are updated

and a new personalized skill graph is generated for the student. A diagnostic model running in RAD API is based on the Elo rating system. It captures student skill abilities and skill difficulties. All of these values are tracked continuously as student performance data flows asynchronously.

**1.** Harry is paid a regular hourly wage of $12.50 per hour for working up to and including 40 hours in 1 week. For each additional hour he works in a week, Harry is paid twice his regular hourly wage. Harry worked 46 hours this week. What is his pay for this week?

(Note: Amounts are before taxes and benefits are deducted.)

**A.** $ 537.50
**B.** $ 575.00
**C.** $ 650.00
**D.** $ 787.50
**E.** $1,150.00

**H.A.MATH.OAF.QPEF.QG.L2.1**
Holistic Framework • Core Academics • Math • Operations, Algebra, and Functions • Quadratic and Polynomial Equations and Functions • Quadratic Growth • Level 2 - Create a quadratic function for data

**Figure 1. An example of a Math question and the Holistic Framework skill that indexes it.**

These diagnostic records of student skill masteries are used to produce recommendations when learners request instructional resources. RAD uses its hierarchical knowledge of the subject domain to inspect the category of knowledge and evaluates which skills/ skill areas would be the most helpful for the learner to review. Recommendations draw on the catalog of instructional content. After learners interact with the learning resources, they continue the lifecycle by continuing their progress with more test preparation and practice with ACT Academy quiz/test items.

## SKILL-LEVEL DIAGNOSTICS IN RAD API

### Holistic Framework

Act, Inc. developed a subject skill taxonomy called Holistic Framework covering domains of the ACT Test – Math, Reading, Science, and English [1]. Holistic Framework (HF) consists of over 4300 nodes. Every skill node of HF could be up to 8 levels deep. HF is the primary mean for indexing all new content ACT publishes including and not limited to ACT Test question items. See Figure 1 for an example of HF skill indexing of an ACT Test item.

### Elo Rating System

RAD API uses an Elo rating system to produce diagnostic skill mastery values from student performance. Elo, named after its inventor Arpad Elo, is a rating system that tracks rating values of two classes of variables for the modeled events [2]. In chess, where Elo found its first use, the events are chess matches and the variables are opponent 1 ability and opponent 2 ability. After each match, the ratings of opponent abilities are updated based on the outcome (a win of either opponent or a draw). When Elo is used in the educational domain, an event is the student's opportunity to answer a question item correctly. The student is opponent 1, and the item is opponent 2. Often, a set of skills relevant to the question item represent opponent 2. Student abilities can be represented hierarchically as a set of student-skill abilities together with an overall ability. For an extended discussion see an overview paper by Gřihák and colleagues [3]. Elo has a few desired properties. First, Elo predominantly uses local updates of the tracked values – student abilities, item or skill difficulties. Second, it requires minimal fitting or tuning. And third, student success or failure always results in a respective increment or decrement of their tracked ratings.

#### *Simple Student-Item Elo*

The simplest case of an Elo is the student-item parameterization There is student's unidimensional ability $\theta_i$ and difficulty of a question item $\beta_j$. A probability of student $i$ answering item $j$ correctly is computed as shown in Equations (1) and (2).

$$p_{ij} = {}^{1}/_{1 + e^{-m_{ij}}} \quad (1)$$

$$m_{ij} = \theta_i - \beta_j \quad (2)$$

#### *Student-Skill Elo*

Instead of tracking item difficulties in the Simple Student-Item Elo, one can replace them with skill difficulties if skill labels are available for all question items. This could be done for several reasons. One, if the data is coming from a system that has longer student exposure to skills, skills could be used as units of transfer to better track learning rather than using items that students interact with once or twice. Two, if the item pool is heterogeneous, less reliable, or extremely large and it is less efficient to track item properties. Equation (3) shows how to compute the probability of student *i* answering item *j* correctly, when skill difficulties are used. Variable $q_{jk}$ is an element of a Q-matrix – a matrix of 1's and 0's, where a value of 1 means that a skill is relevant to a question item.

$$m_{ij} = \theta_i - \sum\nolimits_k q_{jk}\beta_k \quad (3)$$

#### *Hierarchical Student-Student/Skill-Skill Elo*

The version of Elo that is used in RAD API is hierarchical. It tracks student abilities at two levels – overall, and per-skill. There is a global student $\theta_i$, as well as $\theta_{ik}$ values per each student-skill tuple. Skill difficulties are also retained in this approach. The form of the probability of correctness for the hierarchical Elo is given in Equation (4).

$$m_{ij} = \theta_i + \sum\nolimits_k q_{jk}\theta_{ik} - \sum\nolimits_k q_{jk}\beta_k \quad (4)$$

#### *Updates to Elo-tracked Values*

Values tracked by Elo (e.g., student abilities or skill difficulties) are maintained in the log-odds form. Initial values of all parameters are customary to be set to 0, before Elo has *seen* any data pertaining to those parameters. When a new data record arrives, special rules are used to update tracked values. Equations (5) and (6) show examples of these

rules. Here, $K$ is a sensitivity parameter which, in this case, is constant. $C_{ij}$ is actual correctness of student's response (a value of 0 or 1), and $p_{ij}$ is the prior estimate of the probability of correctness as it was defined in Equation (1).

$$\theta_i = \theta_i + K \cdot (C_{ij} - p_{ij}) \quad (5)$$

$$\beta_j = \beta_j - K \cdot (C_{ij} - p_{ij}) \quad (6)$$

The difference between updating student and item parameters is the sign in front of the actual/expected value difference. When more student-level and environment-level parameters are used, for example, student-skill ability Elo and skill difficulty respectively, the sign is set in a similar manner. In Equations (5) and (6) single sensitivity $K$ was used. One could use separate sensitivities for updating tracked parameters for students and items. There are also other ways to define sensitivity. An example of an alternative definition we used in our work is given in Equation (7). Here, $K$ is redefined as a ratio, where the denominator – $n_i$ – is a number of prior data points used to re-estimate student ability $\theta_i$, and $a$ and $b$ are parameters.

$$\theta_i = \theta_i + \frac{a}{1 + bn_i} \cdot (C_{ij} - p_{ij}) \quad (7)$$

Hierarchical Student-Student/Skill-Skill Elo used in RAD API has 3 classes of tracked values: student abilities, student-skill abilities, and skill difficulties. The starting values for all of them are 0 on the logit scale. For each class of the values, there are two hyper-parameters – $a$ and $b$ (see Equation (7)) – that control value updates. Thus, there is a total of 6 hyper-parameters in this version of Elo – quite a small number compared to other approaches.

*Propagation of the Student-Skill Ability Estimates*

RAD API explicitly tracks logit values student progress with leaf HF skills the question items were indexed with. However, HF skills are multi-level and, traditionally to ACT, students receive reports by subject (say Math) and area (one level below subject). In order to produce mastery values for higher-level HF nodes, an average is taken of the sub-nodes that are relevant to items a student has taken. Given the size of the HF taxonomy and the limited number of items each student might have attempted; this sort of propagation upward is an approximation over inherently sparse data.

## PROBLEM STATEMENT

In the ACT Academy, when a student starts using a system that can offer dozens of question items per skill, a cold start problem arises. Given the size of Holistic Framework and other skill taxonomies and skill schemas known in the field, it is common to expect a certain period in the beginning when a student has to put faith in the diagnostic and recommendation power of the algorithms. We would like to make this period of uncertainly shorter, if possible, even if the skills practice is not focused. We are going to leverage the tree structure of the skills, in our case – the HF taxonomy – and track student mastery at multiple levels in order to reduce the diagnostic sparsity early on.

## APPROACH

We will create a multi-level modeling version of the Student-Student/Skill-Skill Elo variant we described ago and would perform a series of comparisons to rank it against a non-multi-level version of the Elo. For simplicity, we will refer to them as Regular and Multi-level Elo model from now on.

### Elo Implementation

Although Elo is a rating system, it could be treated as a machine learning model. A function could be defined that, given the student performance data and the hyper-parameters (of which our particular brand of Elo has 6), can compute a likelihood value. We used `R` software, its package `optimParallel`, and a custom objective function to find optimal values of Elo's hyper-parameters by running an L-BFGS-B algorithm. It is also possible to define a gradient for Elo hyper-parameters. However, we opted to let `optimParallel` package simulate the gradient.

### Multi-level Elo

The difference of the multi-level Elo from the regular version would be that, instead of updating student-skill ability for the leaf HF skill node only using Equation (7), we will be applying it to all of the ancestors of the skill node recursively. If, for example, a student answered a math question item from Figure 1, then student-skill ability valued would be updated for skills H.A.MATH.OAF.QPEF.QG.L2.1, H.A.MATH.OAF.QPEF.QG.L2, H.A.MATH.OAF.QPEF.QG, H.A.MATH.OAF.QPEF, and so on until root skill H.

In the cases, when a lower-level student encounters a skill they have not seen before, multi-level Elo, instead of using a starting value of 0 logits as student-skill mastery, would make a walk up the skill hierarchy in search of a higher-level skill that has been estimated before. The search continues until an ancestor skill with at least one prior attempt is found.

This approach is inspired and semantically similar to the-so-called *accordion procedure* piloted in cognitive diagnostics [5]. There, authors, in the situation when skills are too numerous for a reliable assessment, resorted to higher-lever parent skills in order to improve the accuracy and reliability of the psychometric models.

### Model Comparison

We used a combined 5 times 2-fold cross validation F-test to compare regular and multi-level Elo versions [6]. This approach was validated on multiple datasets and shown reliable model ranking results. The use of 2-fold cross validation defends against increased overlap of the training sets when the number of folds is 3 or more. This approach has low Type I error rate. Due to the nature of how Elo operates, we will only be using student-stratified cross-validation so that whole student records are placed in one or the other folding of the data. To assess model performance, we will use accuracy and root mean squared error (RMSE). Accuracy would let us know how often the model guesses the right question answering outcome (right or wrong).

RMSE would tell us how far numerically from the correct outcome our prediction was.

An issue of how the performance metrics are computed when learning data is being modeled has been previously addressed when discussing the accuracy of the Deep Knowledge Tracing approach. In [7], authors argued that, for the learning data, it makes more sense to compute model performance by-skill and report an average value and not compute an overall performance for the whole dataset. The two approaches to computing model predictive accuracy could result in polar outcomes. In order to avoid the conundrum, we would compute models' performance using both approaches and will refer to them as *by-row* (averaging model performance overall rows) and *by-skill average* (aggregating model performance within skills and then taking an average).

As with any educational technology, there is great variability in how much effort each user spends with it. As a result, there is often a *heavy tail* of users that spent very little time in the system and are, sometimes, considered separately or are not considered at all. To draw model performance comparisons for the higher effort and lower effort users, we will perform a median split using the number of rows per each user as the criterion. Thus, we will compare models for all data, data of high effort users, and data of low effort users.

## DATA

Data was collected by ACT Academy launched in March of 2018. By January 2019, when the dataset was extracted, over 3,340,000 attempts to solve over 1,000 distinct problems were done by 94,000 students covering over 280 distinct Holistic Framework skills. Students do not have long practice sequences with each skill. Out of about 2,175,000 student-skill combinations, in 69% of the cases skills are attempted only once. In 12% of the cases – twice. In 6%, 3%, and 2% of the cases they attempt skills for 3, 4, and 5 times respectively. Six and more attempts per skill are made in 8% of the remaining cases.

## RESULTSLS

Results of comparing regular (Student-Student/Skill-Skill) Elo to a multi-level version of the same model are given in Table 1. Rows show outcomes of the F-tests per {slice of data, type of comparison, metric} tuple. First of all, when inspecting by-row model performance metrics computations (unshaded table rows) none of the comparisons/F-tests come out significant whichever data slice we are taking and whichever metric (accuracy or RMSE) we are considering. Second, the multi-level model has an edge with respect to RMSE on all data and with respect to both accuracy and RMSE on the data of high effort students. There is no difference between models on low effort students' data. From these results, it could be concluded that multi-level Elo is superior to the regular Elo with a statistically tangible edge.

| Students | Metric comp.-n | Metric | Regular Elo | Multi-level Elo | p-value | Result |
|---|---|---|---|---|---|---|
| All | By-row | Acc. | 0.7179 | 0.7178 | 0.334 | n.s. |
| | | RMSE | 0.4338 | 0.4340 | 0.122 | n.s. |
| | Skill avg. | Acc. | 0.7298 | **0.7301** | 0.075 | . |
| | | RMSE | 0.4248 | **0.4246** | 0.000 | *** |
| High effort | By-row | Acc. | 0.7303 | 0.7303 | 0.481 | n.s. |
| | | RMSE | 0.4274 | 0.4275 | 0.126 | n.s. |
| | Skill avg. | Acc. | 0.7368 | **0.7371** | 0.041 | * |
| | | RMSE | 0.4208 | **0.4207** | 0.000 | *** |
| Low effort | By-row | Acc. | 0.6829 | 0.6828 | 0.107 | n.s. |
| | | RMSE | 0.4516 | 0.4518 | 0.123 | n.s. |
| | Skill avg. | Acc. | 0.6870 | 0.6863 | 0.380 | n.s. |
| | | RMSE | 0.4473 | 0.4474 | 0.438 | n.s. |

**Table 1. Comparison of Regular Elo and Multi-level Elo. We list averages across 10 values of 5 runs of 2-fold cross-validation. "n.s." means the comparison result is not significant. Boldface number highlights statistically**

## REFERENCES


1. Camara, W., O'Connor, R., Mattern, K., & Hanson, M. A. (2015). Beyond Academics: A Holistic Framework for Enhancing Education and Workplace Success. ACT Research Report Series. 2015 (4). ACT, Inc.
2. A. E. Elo. The rating of chess players, past and present. Arco Pub, 1978.
3. Gřihák, J, Pelánek, R., Niznan, J. (2015) Student models for prior knowledge estimation. In International Conference on Educational Data Mining (EDM 2015), pp 109–116.
4. Corbett, A.T., Anderson, J.R. (1995) Knowledge tracing: Modeling the acquisition of procedural knowledge. User Modeling and User-Adapted Interaction 4(4), 253–278.
5. de la Torre, J., & Sun, Y. (2018, October). The accordion procedure: A method for accommodating a large number of attributes in cognitive diagnosis modeling. Poster presentation at the annual Educational Technology and Computational Psychometrics Symposium, Iowa City, IA.
6. Alpaydm, E. (1999). Combined 5x2 CV F-test for comparing supervised classification learning algorithms. *NComputationtation*, *11*(8), 1885-1892.
7. Khajah M., Lindsey, R.V., Mozer, M. (2016) How deep is knowledge tracing? In *Proceedings of International Conference on Educational Data Mining,* pp. 94-101.