Comparison of Different AIED Models and Evaluation Methods

Sihan Ma

Nanjing University of Information Science and Technology Nanjing, China

* Corresponding Author Email: 202183020002@nuist.edu.com

Abstract. In the educational field, although machine which attempts to learn AI is still in their early stages, the approach has yet to show remarkable results when facing complex challenges without obvious cut-off points, such as grading students' papers or exploring enormous and complicated data collections. AI can also be used to create virtual learning environments, intelligent testing systems, and automated grading systems. AI in educational fields refers to the application of AI technology to enhance and support the studying processes, such as tracking students' behavior and constructing models that can accurately hypothesize students' achievements. It can include the use of AI-powered tutoring systems, personalized learning platforms, and data analysis tools that can help teachers and administrators better understand student needs and progress. This paper mainly concentrates on the field of artificial intelligence tutoring and summarizes the methods by which intelligent tutors assess student performance by comparing some student models and the input, output, and model forms of methods for evaluating fine-grained interactions of intelligent tutors. This article also provides basic information and new perspectives for studying which methods to use to model and evaluate tutorial learning.

Keywords: AI, Machine learning, Student models, Education.

1. Introduction

AI in Education (AIED) encompasses a wide range of technologies, including AI-driven, individualized education and dialogue systems, student writing analytics, intelligent agents in game-based environments, chatbots that are supported by students, and student-centered exploration learning. Additionally, it encompasses whole-school initiatives, individual student computer use, student use of mobile devices outside of the classroom, and other things [1]. The field of AIED combines innovation and derivation. On the one hand, it incorporates theories and practices from allied disciplines like education, cognitive science, and artificial intelligence. However, it also raises new, more complicated study queries: What is the nature-knowledge? How is it stood for? Which teaching techniques work best, and when should they be applied? What misconceptions do students have?

AIED has been researched for over 40 years to solve these problems. In the 1970s, the first AI systems to provide personalized learning guidance were created, including the revolutionary educational tool BUGGY, which is useful for teaching simple addition and subtraction. To predict the errors that students might make while studying mathematics, the system uses the incorrect question bank model. ITalk2Learn helps students learn the concepts and applications of scores in mathematics and builds a learner model to analyze learners' existing mathematical knowledge, cognitive needs, emotional states, and feedback. In 1977, Wescourt et al. designed the BIP system to assist basic language teaching. In 1977, MIT developed the WUMPUS game system for training in logic, probability, judgment theory, and geometry. In 1982, Sleman and Brown put forward the concept of a auxiliary system. In 1987, the prototype IDExpert system was developed to aid instructional design decisions. In 1992, Brusilovsky proposed the intelligent instruction system ITEM/IP. Brusilovsky et al. developed the first adaptive teaching system in 1996. In 2013, MIT's Ehsan Hoque et al. developed MACH, a social skills training system. At present, the AIED industry can be roughly divided into four types of application scenarios: intelligent tutoring systems, intelligent learning collaboration, intelligent virtual reality teaching environments, and model-based intelligent learning tools.

In order to better study student models and their evaluation methods, this paper chooses the LISTEN project as the research object. The goal of Project LISTEN is to create a cutting-edge literacy tool---an automated reading tutor. It can observe youngsters read aloud while stories display on a computer screen. This article will introduce the models and evaluation of student performance in LISTEN project, which strives to choose the best method for modeling and evaluating student and tutor learning.

2. Models

This section introduces the research and defines five key indexes of the student model [2], including whether or how to represent time, abilities, noise, underlying traits, and various performance-related factors. The research team tests it on fictitious and actual data to show the generative potential of a framework [3] [4]. Here also evaluates student models that track multiple subskills based on three attributes, including how they train their parameters, how they predict knowledge estimations to forecast student performance, and how they update those extrapolations in light of observed performance.

2.1. Five Key Indexes for Student Models

2.1.1 Time effect

This attribute can be classified as either time-invariant or time-varying depending on how they change over time. Time-invariant skills are those that do not change significantly over time and remain relatively stable throughout an individual's lifetime. Examples of time-invariant skills include basic math abilities, language proficiency, and certain personality traits. Time-varying skills, on the other hand, are those that can change significantly over time and may be influenced by various factors such as experience, education, and training. Examples of time-varying skills include technical skills in a particular field, leadership abilities, and decision-making skills. It is important to note that some skills may be both time-invariant and time-varying, depending on how they are measured and the context in which they are used.

2.1.2 Skill dimension

Skills can be classified as single skill and multiple skill in a step. In a step, a single skill refers to a specific task or ability that is being performed or demonstrated. For example, a step in a recipe that instructs the cook to chop an onion is a single skill. Multiple skills refer to multiple tasks or abilities that are being performed or demonstrated in a single step. For example, a step in a recipe that instructs the cook to chop an onion and mince garlic is an example of multiple skills.

2.1.3 Credit allocation

Credit refers to the division of blame for a step's success (or failure). In models, credit refers to the division of blame for a step's success (or failure) because it helps to determine which parts of the model are responsible for the output. This information can be used to improve the model by identifying and addressing any weaknesses or errors in the parts of the model that are not contributing as much to the overall performance. Credit assignment is important for machine learning models because it allows the model to learn from its mistakes and improve over time. By identifying the parts of the model that are responsible for errors, researchers can adjust or optimize those parts to improve performance. Additionally, credit assignment aids in preventing the model from being overfit to the training data, which can have a negative impact on how well it performs on fresh or previously unexplored data. For example, if a model is trying to predict a certain output and it makes a mistake, credit assignment helps to identify which parts of the model were responsible for that mistake. This allows the model to adjust those parts and improve its performance on future predictions.

2.1.4 High order

In models, high order refers to the use of complex or advanced mathematical techniques to analyze and model data.

In the context of student attributes as potential characteristics, high order methods may be used to analyze and model data related to student attributes such as demographics, prior academic performance, or other factors that may be related to student success. Static student attributes are characteristics that do not change over time. Examples include gender, race, socioeconomic status, and prior academic performance. These attributes can be used as inputs to a model and analyzed to identify patterns and relationships that may be related to student success. High order methods can be used to analyze and model these static student attributes in order to gain a deeper comprehension relating to student success. For example, a high-order method such as a non-linear regression analysis could be used to model the bridge between a student's prior academic performance and their likelihood of success in a future course. This can help to identify patterns or trends that may be missed by simpler or more traditional methods.

2.1.5 Noise

In models, noise refers to any random variations or errors in the data that are not related to the underlying relationship or structure being modeled. Noise can arise from a variety of sources such as measurement errors, inaccuracies in data collection, or random fluctuations in the data. How to depict contrasts between models or between what students know and how they behave. When integrating estimates from various skills (NIDO and DINO), add noise either beforehand or afterward. These techniques for modeling noise are known as slip or guess, NIDO (noisyinput, predictable output), DINO (predictable output, noisyinput)

The summary of five evaluation indexes of student models is shown in TABLE 1.

Table 1 Five Evaluation Indexes of Student Models

<table>
<tr><th>Models</th><th>Time Effect</th><th>Skill Dimension</th><th>Credit Allocation</th><th>High Order</th><th>Noise</th></tr>
<tr><td>KT [5]</td><td>Dynamic</td><td>Single Skill</td><td>Per Skill</td><td rowspan="9">No Latent Trait</td><td>Slip/Guess</td></tr>
<tr><td>CKT [6]</td><td>Dynamic</td><td rowspan="7">Multiple Skills</td><td>Product</td><td rowspan="4">NIDO</td></tr>
<tr><td>Weakest-KT (Blaming Weakest, Credit) [7]</td><td>Dynamic</td><td rowspan="2">Minimum</td></tr>
<tr><td>Weakest-KT (Renew Weakest Skill)</td><td>Dynamic</td></tr>
<tr><td>KT+NIDA [8]</td><td>Dynamic</td><td rowspan="2">Product</td></tr>
<tr><td>KT+DINA [9]</td><td>Dynamic</td><td rowspan="2">DINO</td></tr>
<tr><td>LR-DBN [10]</td><td>Dynamic</td><td rowspan="2">Sigmoid</td></tr>
<tr><td>LR-DBN Minus [11]</td><td>Dynamic</td><td>Single Skill</td><td>Slip/Guess</td></tr>
</table>

2.2. Comparison of Student Models for Tracing Multiple Subskills

2.2.1 Fit Parameters

The prior technique [12]modeled every set of subskills as a separate subskill, which avoided the issue of fitting parameters. The transfer of learning between groups is neglected, and the subskills of various groups are treated as independent skills. The majority of solutions [13] [14]simply assume that each subskill employed in a step is fully accountable for that step. By examining all the stages utilizing each subskill, they developed a unique KT (Knowledge tracing) model for each subskill. As a result, for each subskill it employs, the training data shows the identical observation stages. To estimate each subskill through model parameters, they just follow the identical training techniques that be similar with standard KT.

2.2.2 Predict Performance

Previous approaches to the multi-subskill problem

combine the probability of all the skills necessary for proper execution in various ways to forecast the performance of a step. One approach is to multiply them on the premise that their probabilities

are independent and then show them as a forecast of independent sub-skill performance. The weakest subskill [15] forecast, however, results from the weakest subskill option taking the least value.

2.2.3 Update Estimate

The previous norm was to use various methods to divide responsibilities among them in order to track the success or failure of the step when a step requires numerous sub-skills. The Update Weakest Subskill technique merely updates the subskill with the lowest probability in the step using the update equation mentioned above while leaving the other subskills alone. When a step fails, its variation treats other sub-skills as accurate.

The summary of some student models is reviewed in TABLE 2.

Table. 2 Summary of Some Models

Models	Fit	Predict	Update
CKT	Train separately. Assign each full responsibility.	Multiple skills expectations.	Update together. Equations define accountability.
Weakest-KT (Blaming Weakest, Credit)		Minimum expectations for subskills.	Update separately. Full responsibility for each.
Weakest-KT (Renew Weakest Subskill)			Update only the weakest skill.
LR-DBN Minus		Log-logistic on subskill expectations.	Update together. Regression using logs distributes blame.
LR-DBN	Train together. Regression using logs distributes blame.		
KT+NIDA/DINA [16]	Train together. Assign each full responsibility.	Multiple skills estimates.	Update together. Full responsibility for each.

3. Evaluation Methods

3.1. Three Evaluation Methods

3.1.1 Randomized Controlled Trials Analysis (RCT)

The main goal of randomized controlled trials is to assign trial procedures at random in order to strongly infer that variations in the trial protocols are the cause of notable disparities in results. Participants in randomized controlled trials are assigned at random to one of several possible therapies. To assess the efficacy of various interventions, RCTS compiles and contrast the outcomes for each condition. The reading tutors at Project LISTEN choose one approach out of many to teach students a specific word at random. For each student, the reading tutor chooses at random how to teach or practice a certain word in order to compare the outcomes using RCT analysis. Researchers or examples use how effectively students understand the term the following time they encounter it to evaluate the results of each such randomized study. By comparing the pooled results of each intervention, researchers can estimate their relative effectiveness against each other.

3.1.2 Learning Decomposition

The primary objective of this methods is to differentiate between various exposure kinds to provide empirical evaluations of various practices or styles of instruction when fitting a learning model to student success data. By describing the experience as a function of each type of exposure, or as a linear combination of them, earlier work has already implemented this principle in the form of a specific model, the exponential learning curve.

There will be optional instantiations as a result of optional model forms. The power law is an acceptable substitute for the exponential learning curve. Compared with power-law curves [17], exponential curves typically match specific performance data, despite the fact that power laws are frequently employed to match average performance data for data from numerous participants.

Another, more complicated scheme includes more parameters to fit and is presented recursively rather than in a closed form. It is founded on a broad theoretical framework that describes how people acquire and lose knowledge throughout time. The model's ability to suit the data is referred to as learning decomposition implementation.

3.1.3 Knowledge Tracing

In order to update the estimation of the state in accordance with the ongoing observation of students' academic performance, the core concept of knowledge tracking is to model the changes of students' academic performance and knowledge state during the process of skill acquisition. If the learner has mastered the skill in the specified phase, this node's value is true; otherwise, it is false. Because the node is concealed, it is impossible to see its value directly. In contrast, knowledge tracking updates this probability with observations of student performance every time a student encounters an opportunity to use a skill.

In a model, the following variables are used to describe it:

a. knew: likelihood that a student knew a skill before receiving instruction
b. probability_T_i: Likelihood that the one is a category i intervention
c. learn_T_i: Likelihood of learning the subskill through a type T intervention
d. forget_T_i: Likelihood of losing a known talent in the presence of a kind T_i intervention
e. guess: Likelihood of responsing truthfully without having the necessary knowledge
f. slip: Likelihood of giving an inaccurate response despite having the knowledge

The issue of contrasting the outcomes of various tutorial behavior types thus becomes a matter of contrasting the values of the a and b parameters for the various c's corresponding to those behavior kinds.

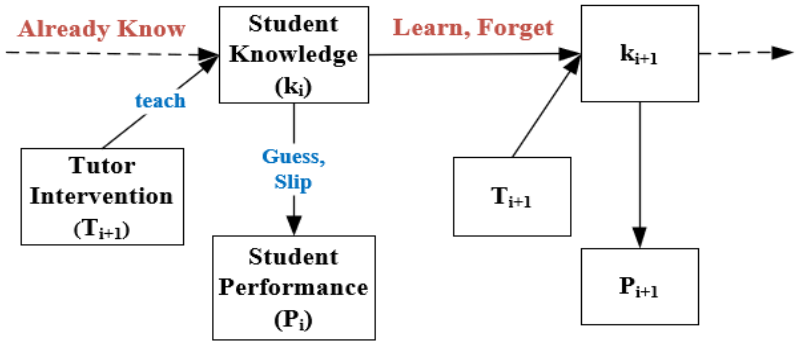


Fig 1 Knowledge Tracing Model

3.2. Comparison of information input

RCT analysis does not compare unobserved, undefinable, or veiled test outcomes because it only compares observable results. The student's reading ability is not visible when the reading tutor reads a word. The student's reading performance was unclear when the reading tutor offered assistance before the student accepted the word. The recency effect concealed pupils' reading abilities if they had seen the word earlier in the day.

Analyzing experiments' outcomes and practices are known as learning decomposition. The test junctions that aren't observed for performance or whose definition isn't muddled are excluded.

Knowledge tracing accepts incomplete and missing observations and models results and practices as identical performance nodes in a dynamic Bayesian network. The estimated chance that the learner learned the word is updated even if a word is veiled by tutor-initiated help, rendering the credit useless.

3.3. Expected output comparison

RCT analysis evaluated the experiment's results and determined the variables—primarily the settings of the experiment—that predicted or influenced the results. Using this method, therapies can be compared. By employing student status as a component in statistical tests, it can estimate student

abilities that affect outcomes; nevertheless, such assessments do not make use of randomized trial assignments. As a result, the RCT analysis was more equipped than the students to assess the intervention. Even while the review evaluated students' abilities as a factor, it did not update such assessments gradually based on ongoing observations of student performance. In addition, the review studied the complete set of trials at once.

The fundamental objective of the learning breakdown is to compare the relative effects of various exercise types, while it can also fit other parameters including students' baseline performance and learning pace. Unless the encounter is treated as a different type based on its immediate results, it forecasts a student's performance when confronted with a talent. However, this is dependent just on how often the person has learned the skill, not on how well they have previously worked at it. Learning decomposition can combine knowledge about previous performance by creating effective and ineffective interactions as double distinct forms of training. In fact, investigating that idea might be enjoyable. This does not change the issue that merely counting the number of various encounter kinds ignores the sequence and most recent effects. As a result, the performance predicted by the resulting model after a set of correct responses followed by five incorrect responses is the same as the performance anticipated after a set of incorrect responses followed by a set of correct answers. However, by treating spaced workouts and massive exercises as different categories, learning decomposition [18] can imitate certain short-term benefits.

Knowledge tracking was created with the express purpose of updating the instructor's assessment of the abilities each student has attained at each practice opportunity. You may evaluate the effect of each exercise on learning by accounting for the type of activity. The instructor can forecast how well the student will perform based on the most recent knowledge estimates (plus slippage and guesswork parameters) and make plans appropriately, such as selecting a question that is simpler to avoid a potential impending failure. Because the forecast takes into account performance over the preceding practice period rather than just the quantity of practice, knowledge tracking is a stronger predictor of performance than learning breakdown. Given that knowledge tracking bases its predictions on observed performance, for instance, if the data set contains N-1 sequences of practice chances by the identical ability from double distinct users, with the identical order of training varieties, then estimate that performance will be higher after success than after failure. The learning decomposition, in contrast, predicts the same performance in all scenarios because it is entirely based on the quantity and variety of practical chances, barring the inclusion of any student-specific characteristics or performance-based differentiation of prior encounters. Based on ongoing performance monitoring, Knowledge Tracking continuously updates its assessment of students' knowledge. It is responsive to students' performance in terms of order but not in terms of timing. By recognizing extensive and spaced experimentation as a different kind of practice, learning decomposition partially overcomes this problem. The settings of learning and forgetting can also be adjusted by knowledge tracking based on the interval between succeeding skill experiences. The time effects of this specific refining model are discrete for both techniques, for instance by categorizing pupils as experiencing subsequent skills on the same or different days.

3.4. Comparison of model forms (presentation, calculation and extension)

The RCT analysis has no local minimum to jam, no startup circumstances to be sensitive to, and no hidden variables to estimate. Even with tens of thousands of trials, suitable statistical tests can be run right away in common statistical packages like SPSS to compare the outcomes of various groups of trials. Adding outcome variables to tests and features that are broken down into or used as factors in statistical tests like the analysis of variance are examples of extended analysis.

Deconstruct estimated parameters that show the effect of each practice mode on an exponential model's time-varying effectiveness. Non-linear regression in SPSS11.0 lasted like almost a minute to train prescribed amounts of models (one for each word) into tenfold verb interactions in our experimental contrast of assessment options [19]. In order to further differentiate between different

exercise kinds or other performance-related factors, such as the impact of word length on reading comprehension, parameters must be added to the model.

Knowledge tracing, generalization to differentiate between practice kinds, estimation parameters, representation of each practice type's impacts on the knowledge's hidden state in dynamic Bayesian networks, and performance of these knowledge's effects on observed students. The BNT-SM [20] needs approximately 10 hours to learn and test the model on the aggregate of data, whereas learning the decomposition only occupies one minute. The process of expanding the model involves adding nodes to reflect facts and linkages to reflect assumed causality.

4. Conclusion

This paper first reviews the project briefly and then starts from the two topics of model and evaluation, respectively analyzing and comparing several student models and methods of evaluating student performance that can be applied to the project.

Depending on the student model utilized, we demonstrate significant differences in success predictors across the reading tutors in the LISTEN program. However, taught methods can easily be covered and overfit. The number of observations in each scenario also affects how well cross-validation estimations of learning strategy expectations perform. Future goals include proposing accurate models like the LR-DBN and potent classifiers, examining the process by which student model precision influences prediction success precision, and learning broader policies to increase coverage and reduce overfitting. Additionally, there are tests to see how well predicted success matches reality.

References

[1] Wayne Holmes, Maya Bialik, Charles Fadel. Artificial Intelligence in Education Promises and Implications for Teaching and Learning.

[2] Yanbo Xu and Jack Mostow. A Unified 5-Dimensional Framework for Student Models.

[3] González-Brenes, J.P. and J. Mostow. What and when do students learn? Fully data-driven joint estimation of cognitive and student models. In Proceedings of the 6th International Conference on Educational Data Mining, S.K. D'Mello, R.A. Calvo, and A. Olney, Editors. 2013, International Educational Data Mining Society: Memphis, TN, p. 236-239.

[4] González-Brenes, J.P. and J. Mostow. Dynamic cognitive tracing: towards unified discovery of student and cognitive models. Proceedings of the Fifth International Conference on Educational Data Mining 2012. Chania, Crete, Greece.

[5] Corbett, A. and J. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 1995. 4: p. 253-278.

[6] Koedinger, K.R., P.I. Pavlik, J. Stamper, T. Nixon, and S. Ritter. Avoiding problem selection thrashing with conjunctive knowledge tracing. In Proceedings of the 4th International Conference on Educational Data Mining. 2011: Eindhoven, NL, p. 91-100.

[7] Gong, Y., J.E. Beck, and N.T. Heffernan. Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. Proceedings of the 10th International Conference on Intelligent Tutoring Systems, 35-44. 2010. Pittsburgh, PA. Springer Berlin / Heidelberg.

[8] Beck, J.E., et al. Does help help? Introducing the Bayesian Evaluation and Assessment methodology. in 9th International Conference on Intelligent Tutoring Systems. 2008. Montreal.

[9] Hartz, S., A Bayesian framework for the unified model for assessing cognitive abilities: Blending theory with practicality. 2002, University of Illinois at Urbana Champaign: Unpublished doctoral dissertation.

[10] Xu, Y. and J. Mostow. Using logistic regression to trace multiple subskills in a dynamic Bayes net. Proceedings of the 4th International Conference on Educational Data Mining, 241-245. 2011. Eindhoven, Netherlands.

[11] Yanbo Xu, Jack Mostow. Comparison of methods to trace multiple subskills: Is LR-DBN best?

[12] Cen, H., K. Koedinger, and B. Junker. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. in Proceedings of the 8th International Conference on Intelligent Tutoring Systems. 2006. Jhongli, Taiwan.

[13] Cen, H., K.R. Koedinger, and B. Junker. Comparing Two IRT Models for Conjunctive Skills. in Ninth International Conference on Intelligent Tutoring Systems. 2008. Montreal.

[14] Koedinger, K.R., et al., Avoiding Problem Selection Thrashing with Conjunctive Knowledge Tracing, in Proceedings of the 4th International Conference on Educational Data Mining. 2011: Eindhoven, NL. p. 91-100.

[15] Gong, Y., J. Beck, and N.T. Heffernan. Comparing Knowledge Tracing and Performance Factor Analysis by Using Multiple Model Fitting Procedures. in Proceedings of the 10th International Conference on Intelligent Tutoring Systems. 2010. Pittsburgh, PA: Springer Berlin / Heidelberg.

[16] Studer, C. Incorporating Learning Over Time into the Cognitive Assessment Framework. Unpublished PhD, Carnegie Mellon University, Pittsburgh, PA, 2012.

[17] Heathcote, A., S. Brown, and D.J.K. Mewhort. The Power Law Repealed: The Case for an Exponential Law of Practice. Psychonomics Bulletin Review, 2000: p. 185-207.

[18] Beck, J.E. Using learning decomposition to analyze student fluency development. ITS2006 Educational Data Mining Workshop, 21-28. 2006. Jhongli, Taiwan.

[19] Zhang, X., J. Mostow, and J.E. Beck. A Case Study Empirical Comparison of Three Methods to Evaluate Tutorial Behaviors. 9th International Conference on Intelligent Tutoring Systems, 122-131. 2008. Montreal: Springer-Verlag.

[20] Chang, K.-m., J. Beck, J. Mostow, and A. Corbett. A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems. Proceedings of the 8th International Conference on Intelligent Tutoring Systems, 104-113. 2006. Jhongli, Taiwan.