Exploring Alternative Methods for Error Attribution in Learning Curves Analysis in Intelligent Tutoring Systems

Adaeze NWAIGWE[1], Kenneth KOEDINGER[1], Kurt VANLEHN[2], Robert HAUSMANN[2], Anders WEINSTEIN[2]
[1]Human Computer Interaction Institute,
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
anwaigwe@cs.cmu.edu, koedinger@cmu.edu,
[2]LRDC, University of Pittsburgh, PA, USA
[vanlehn, bobhaus, andersw,]@pitt.edu

Abstract. It is becoming a standard technique to use learning curves as part of evaluation of intelligent tutoring systems [1,2,3], but such learning curves require a method for attributing errors. That is, the method must determine for each error a student makes what "knowledge component" in the student model is to blame. To this point, alternative methods for error attribution have not been systematically investigated. We implemented four alternative methods for error attribution – two temporal heuristics and two location-temporal heuristics. We employed two evaluation standards – a statistical standard for measuring model fit and parsimony and the Kappa technique for measuring inter-observer reliability. We looked to see which method better met the "learning-curve standard" that is, led to better prediction of students' changes in error rate over time. Second, we asked if the codes generated by the methods better met the "human-match standard", that is, were they like error attributions made by human coders. Both evaluation standards led to better results for the location-temporal heuristic methods than the temporal heuristic methods. Interestingly, we found that two of the methods proposed were better at error attribution, according to the learning-curve standard, than the original cognitive model of the intelligent tutoring system. Overall, these results suggest that the heuristics proposed and implemented in this paper can generally aid learning curve analysis and the perhaps, more generally, the design of student models

Introduction

Increasingly, learning curves have become a standard tool for measurement of students' learning in intelligent tutoring systems. Slope and fit of learning curves show the rate at which a student learns over time, and reveal how well the system model fits what the student is learning. However, these learning curves require a method for attributing error to the "knowledge components" (skills or concepts) in the student model that the student is missing.

In order to facilitate learning curve analysis which can also be referred to as internal evaluation [11], we have proposed and implemented four automated heuristics for attributing error to the desired knowledge components, that is, the skills which the student is required to acquire, during the course of problem solving. These are two temporal heuristic models and two temporal-location heuristic hybrid models. We specifically addressed the following research questions:

1. Which heuristic better predicts students' changes in error rate over time?
2. Which heuristic produces error attribution codes that match those of human coders, best?

To address question 1, we applied the statistical component of learning factors analysis [4], while in answering question 2, Kappa analysis was used to compare codes from each heuristic to those produced by human coders.

Section 2 is a brief review of literature, section 3 describes the data source used in our analysis, while section 4 illustrates our methodology. Results, discussions and conclusions are presented in sections 5, 6 and 7, respectively.

1. Literature Review

The power relationship between the error rate of performance and the amount of practice is shown in equation 1 [5]. The relationship shows that the error rate decreases as the amount of practice increases. As previously mentioned, this function is known as a "learning curve".

$$Y = aX^{b} \quad \ldots\ldots..(1)$$

where
Y = the error rate
X = the number of opportunities to practice a knowledge component (e.g., skill, concept, rule, constraint)
a = the error rate on the first trial, reflecting the intrinsic difficulty of a knowledge component
b = the learning rate, reflecting how easy a knowledge component is to learn

A good cognitive model is expected to follow the power law. A cognitive model is a set of production rules or skills encoded in intelligent tutors to model how students solve problems. Productions embody the knowledge that students are trying to acquire, and enable the tutor estimate each student's learning of specific skills as the student works through exercises [12].

Cen and colleagues proposed a semi-automated method for improving a cognitive model called Learning Factors Analysis (LFA) [4]. LFA is a three-component system, implemented in Java, which combines statistics, human expertise and combinatorial search to evaluate and improve a cognitive model. The statistical component quantifies the skills and provides information on data fit – this is the component we make use of in this study.

While the power law model applies to individual skills, it does not include student effects. In order to accommodate student effects for a cognitive model that has multiple rules, and that contains multiple students, Cen et al extended the power law model to a logistic regression model (equation 2 below) based on the following assumptions:

1. Different students may initially know more or less. An *intercept* parameter for each student reflects this.
2. Students learn at the same rate. Thus, *slope* parameters do not depend on student. This simplification is to reduce the number of parameters in equation 2 and also because the focus is on refining the cognitive model rather than evaluating student knowledge growth [6,7].
3. Some knowledge components are better known than others. An intercept parameter for each knowledge component captures this.
4. Some skills are easier to learn than others. Thus, this is reflected using a slope parameter for each skill

$$\ln[p/(1-p)] = B_0 + \Sigma\, \alpha_j X_i + \Sigma\, \beta_j Y_j + \Sigma\, \gamma Y_j T_j \quad \ldots\ldots.(2)$$

where
p = the probability of success at a step performed by student i that requires knowledge component j
X = the dummy variable vector for students; Y = the dummy variable vector for knowledge components; T = the number of practice opportunities student i has had on knowledge component j; α = the coefficient for each student, that is, the student intercept; β = the coefficient for each knowledge component, that is, the knowledge component intercept
γ = the coefficient for the interaction between a knowledge component and its opportunities, that is, the learning curve slope

In this paper, we use the statistical component of LFA to quantify the skills and evaluate model fit and parsimony with respect to equation 2. Akaike Information Criterion (AIC) and Bayesian Information criterion (BIC) are two estimators for prediction risk [10] used in LFA and their formulae are shown in equations 3 & 4. Lower AIC & BIC scores, mean a better balance between model fit and complexity. BIC however, imposes a more severe penalty for complexity than AIC.

$$\text{AIC} = -2*\text{log-likelihood} + 2*\text{K}\ldots\ldots\ldots\ldots..\ (3)$$

$$\text{BIC} = -2*\text{log-likelihood} + \text{K} * \ln(\text{n})\ldots\ldots\ldots..\ (4)$$

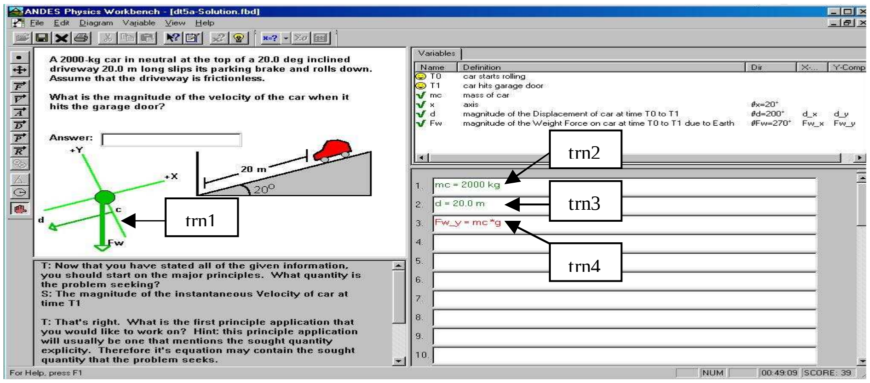
where
log-likelihood measures model fit,
K = number of covariates in equation 2, and measures model complexity,
n = number of observations

2. Data Source

To test the methods, we used data from the Andes physics tutor [8] collected at the US Naval Academy during its regular physics class (see figure 1). This data was collected as part of the Pittsburgh Science of Learning Center’s LearnLab facility that provides researchers, access to run experiments in or perform secondary analyzes of data collected from one of seven available technology-enhanced courses running at multiple high school and college sites (see http://learnlab.org). The data spanned 4 multi-step physics problems solved by 104 students and involved about 18,000 observed “transactions”.

In Andes, the student is required to define all variables before entering them in an equation. One method for defining variables is to draw a vector. In figure 1, the student correctly defines the vector, ‘F_w’, using a drawing (trn 1). The student also successfully defines the mass of the car and its travel distance (trns 2 & 3). However, in trn 4, the student writes a force equation, using the variable ‘g’ which the student is yet to define. This transaction is incorrect. Usually, Andes would indicate success at completing a transaction by turning the entry green. Entries for incorrect transactions are turned red.

The general problem of error attribution is to determine for each incorrect or hint transaction (solicited or unsolicited help from the ITS) what knowledge component in the overlay student model, is to blame. One common solution is that the intelligent tutoring system provides this information (as was sometimes the case in our data set). However, in some log data this information is not available and, furthermore, the error attribution method used by the ITS might not always be the best choice. An alternative strategy is to search in the log for a subsequent correct student action and attribute the error to the knowledge component coding of that correct action. If that subsequent correct action is the next one in time, then one is employing the "temporal heuristic” described below. However, sometimes students jump around in a tutor interface and the next correct action may not be related to the error – it might be better to search for the next correct action that is in the same interface location in which the error occurred. This “location heuristic” has been employed in past approaches [1, 2, 3] and is further described below.



A 2000-kg car in neutral at the top of a 20.0 deg inclined driveway 20.0 m long slips its parking brake and rolls down. Assume that the driveway is frictionless.

What is the magnitude of the velocity of the car when it hits the garage door?

Name	Definition	Dir	X-...	Y-Comp
☺ T0	car starts rolling			
☺ T1	car hits garage door			
✓ mc	mass of car			
✓ x	axis	θx=20°		
✓ d	magnitude of the Displacement of car at time T0 to T1	θd=200°	d_x	d_y
✓ Fw	magnitude of the Weight Force on car at time T0 to T1 due to Earth	θFw=270°	Fw_x	Fw_y

1. mc = 2000 kg
2. d = 20.0 m
3. Fw_y = mc*g

T: Now that you have stated all of the given information, you should start on the major principles. What quantity is the problem seeking?
S: The magnitude of the instantaneous Velocity of car at time T1

T: That's right. What is the first principle application that you would like to work on? Hint: this principle application will usually be one that mentions the sought quantity explicity. Therefore it's equation may contain the sought quantity that the problem seeks.

Figure 1. Example Screenshot of the Andes Intelligent Tutor

3. Methodology

The four heuristics proposed are the two temporal heuristics – TH_1, and TH_2 and two location-temporal heuristics LH_1 and LH_2. Each of heuristics is implemented in pure Java 1.5 and is described below.

3.1. Temporal Heuristic (TH) Methods

In seeking a knowledge component (KC) code for an error transaction for which the KC code is missing, the two TH methods generally assign blame to the first KC the student successfully implements. While the TH_2 method is implemented on all "incorrect" and "hint" transactions, that is, all error transactions, TH_1 is implemented on only error transactions for which the Andes tutor failed to ascribe blame to at least one or more KCs. When Andes performs blame attribution for an error transaction, it usually ascribes blame to the KC or set of KCs, in the problem's solution set, which it believes is responsible for the student's error.

To illustrate how TH_1 is implemented, the java program reads the first student/tutor transaction in an XML log file. If it is an error transaction which has a missing KC code, the program searches down the log till it finds the first correct transaction for the same problem. The KC code for this later transaction is then assigned as KC code to the afore-mentioned error transaction for which a code is sought. If no correct transaction is found, the TH method ends for that error transaction without returning a code. The TH sequence is implemented on all error transactions till the end of the log file is reached. Error transactions for which no KC code is found are excluded from the analysis.

3.2. The Location Heuristic (LH) Method

Let us assume a student makes an error at a certain interface location and a KC code to blame, is sought. When it can, the LH method uses the error location as a heuristic for deciding how to assign blame to the first KC, which the student successfully implements at that same location. While LH_2 implements this method on all error transactions, LH_1 implements the method on only error transactions for which Andes failed to ascribe blame to a KC. The step for the LH method is given in section 3.2.1. With respect to LH_1, the java program reads the first student/tutor transaction in an XML log file. If it is an error transaction which has a missing KC code, the program searches down the log till it finds the first correct transaction, implemented at the same location as the error transaction and for the same problem. The KC code for the correct transaction becomes the KC code for the error transaction for which a code is sought. If no correct transaction is found for the same location as the error transaction, the LH method ends for that error transaction and repeats its sequence again on the next error transaction till it reaches the end of the log file. Again, error transactions for which no KC code is found are excluded from the analysis.

3.2.1. Algorithm for the Location-Temporal Heuristic

1. Read student/tutor transaction. If error transaction, go to 2. Else, go to 3.
2. Does KC code exist? If yes, go to 3. If no, go to 4

3. Last transaction? If yes, STOP. If no, go to 1
4. Does record of actual interface location exist? if yes, go to 5. if no, go to 6
5. Find 1st correctly implemented KC at same location as error transaction. Acquire KC code. If successful, go to 3. If unsuccessful, go to 7
6. Find 1st correctly implemented KC. Acquire KC code. If successful, go to 3. If unsuccessful, go to 7
7. Return ‘no code found’. Go to 3.

3.3. Matching Human Coders

Two University of Pittsburgh physics domain knowledge experts each coded the same sample of error transactions, which were missing KC codes, and spanning four problems. Of these, only those transactions for which the experts had matching codes were used to evaluate the human-match standard. To compare how well codes from each heuristic matched those of the human coders, Kappa analysis was used.

Kappa provides a measure of the degree to which two judges, A and B, concur in their respective sorting of N items into k mutually exclusive categories. A 'judge' in this context can be an individual, a set of individuals who sort the N items collectively, or some non-human agency, such as a computer program or diagnostic test, that performs a sorting on the basis of specified criteria. The level of agreement is determined by the Kappa score. The closer the score is to 1, the better the agreement between pairs of codes [9].

Table 1: Results of the learning-curve standard

Criterion	LH_1	LH_2	TH_1	TH_2
AIC	6,444	6,285	6,759	6,653
BIC	7,578	7,414	7,894	7,781
Loglikelihood	-3,051	-2,972	-3,209	-3,155

Table 2: Results of the human-match standard

	LH_1	LH_2	TH_1	TH_2
Kappa Score	0.78	0.71	0.76	0.68
Asymp. Std. Error(a)	0.019	0.021	0.02	0.022
Approx. T(b)	66.7	61.7	63.5	58.2
Approx. Sig.	0.0	0.0	0.0	0.0
N of Valid Cases	527	527	520	521

a. Not assuming the null hypothesis

b. Using the asymptotic standard error assuming the null hypothesis.

4. Results

Table 1 shows the results obtained in fitting the data from each heuristic to equation (2). As can be seen, LH_2 was the leading performer in terms of data fit, with

AIC score of 6,285, BIC score of 7,414 and loglikelihood value of -2,972, where lower scores mean a better fit and correspondingly, a better cognitive model. Scores with a difference of greater than 6 are considered to be reliably different.

The better fit of LH_2 over LH_1 suggests that simply using the location-temporal method for error attribution may better characterize student learning patterns than using the intelligent tutor's attributions when present. For the temporal heuristic methods, we also see the simpler approach (TH_2) yielding a better fit than the tutor-informed approach (TH_1).

Table 2 shows the results of the human-match standard. To calculate Kappa scores, first, only observations where the two human raters agreed were selected. Then, the selected observations were matched against observations for each heuristic to compute Kappa scores.

As can be seen, the results are quite similar to the results of the learning-curve standard. In this case however, LH_1 and TH_1 provide a better fit than LH_2 and TH_2 respectively.

5. Discussion

Generally, the location heuristics has been shown to produce data with better fit to the learning curve standard and to human codes.

An interesting observation we made though, is that the heuristic models that coded all error transactions, that is, LH_2 and TH_2, produced better fitting data than LH_1 and TH_1 respectively, according to the learning curve standard. These results are shown in table 1.

Given that the human coders only provided a set of KC codes for each error transaction for which the Andes tutor failed to provide one, the results in table 2 are expected. While LH_1 and TH_1 coded the same transactions as the human coders, LH_2 and TH_2 coded more transactions since these methods coded all error transactions whether missing or not. As such, a greater mismatch was expected between the later heuristics and the human codes.

6. Conclusions

In this paper, we present and implement four automated heuristics for error attribution in order to facilitate learning curves analysis. We found that the location-temporal heuristics were better at predicting students' changes in error rate over time. The location-temporal heuristics also fit human codes better than the temporal heuristics. We intend to implement these heuristics on other datasets to test the generality of these results. The availability of datasets from the Pittsburgh Science of Learning Center's 'DataShop' (see http://learnlab.org) will facilitate the process of getting appropriate data.

Interestingly, we found in 2 of 4 comparisons (LH_2 > LH_1 and TH_2 > TH_1 for learning-curve standard) that two of the heuristic models proposed were better at error attribution than the original cognitive model of the intelligent tutoring system.

Overall, these results suggest that the heuristics proposed and implemented in this paper can generally aid learning curve analysis and the perhaps, more generally, the design of student models.

References

[1] Anderson, , J. R., Bellezza & Boyle, C. F., (1993). The geometry tutor and skill acquisition. In J. R. Anderson (Ed.) Rules of the Mind, Chapter 8. Hillsdale, NJ: Erlbaum.

[2] Martin, B., Mitrovic, T., Mathan, S., & Koedinger, K.R. (2005). On Using Learning Curves to Evaluate ITS. Automatic and Semi-Automatic Skill Coding With a View Towards Supporting On-Line Assessment. Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED2005). Amsterdam, IOS Press.

[3] Koedinger, K.R. and Mathan, S. Distinguishing qualitatively different kinds of learning using log files and learning curves. In ITS 2004 Log Analysis Workshop. (2004). Maceio, Brazil. pp. 39-46.

[4] Cen, H., Koedinger, K. & Junker, B. (2005). Automating Cognitive Model Improvement by A* Search and Logistic Regression. In Proceedings of AAAI 2005 Educational Data Mining Workshop.

[5[Newell, A. & Rosenbloom, P. (1981). Mechanisms of Skill Acquisition and the Law of Practice. In Anderson J., (ed.): Cognitive Skills and their Acquisition, Erlbaum Hillsdale NJ

[6] Pirolli P & Wilson M. (1998). A Theory of Measurement of Knowledge Content, Access and Learning, Psychological Review vol.105, 1, pp. 58-82.

[7] Draney, K., Pirolli, P. & Wilson, M. (1995). A Measurement Model for a Complex Cognitive Skill. In Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ

[8] VanLehn, K., Lynch, C., Schultz, K., Shapiro, J. A., Shelby, R. H., Taylor, L., et al. (2005). The Andes physics tutoring system: Lessons learned. International Journal of Artificial Intelligence and Education, 15(3), 147-204.]

[9] Utah State University (2003). Kappa Tool User's Guide, http://www.gis.usu.edu/~chrisg/avext/downloads/kappa.pdf

[10] Wasserman, L. (2004) All of Statistics: A Concise Course in Statistical Inference. Springer

[11] Igbal A., Oppermann R., Patel A., & Kinshuk. (1999). A Classification of Evaluation Methods for Intelligent Tutoring Systems. In U. Arend, E. Eberleh & K. Pitschke (eds.). Software Ergonomie - Design von Informationswelten, B. G. Teubner Stuttgart, Leipzig, pp. 169-181.

[12] Corbett A.T., Anderson, J.R., O'Brien A.T., (1995). Student Modelling in the ACT Programming Tutor. In Cognitively Diagnostic Assessment. Erlbaum, Hillsdale, NJ