Intelligent Review Model

Abstract— **This paper describes the Intelligent Review Model that helps identify the strengths and weaknesses of students who participate in online tutoring systems. It currently tailors the Intelligent Tutoring System, an online system used at the Electrical and Computer Engineering department at Georgia Tech, to provide students with comprehensive feedback about their in-class performance and to help them improve their conceptual knowledge by giving them plenty of questions, related to their areas of weaknesses, to practice. It uses student and class statistics, both obtained from data mining, to identify individual weaknesses and ultimately makes n binary trees for every student, which correspond to n critical concepts. Each of these trees subsume other related concepts, which are topologically ordered based on difficulty. These trees can be used by existing tutoring systems that organize questions by concepts, to help students redress their weaknesses by creating more targeted assignments and practice exercises. The purpose of this model is to provide the ability to tailor any online tutoring system by accounting for the differences in students performance and consequently increasing these systems effectiveness.**

I. INTRODUCTION

An Intelligent Tutoring System (ITS) is computer software designed to simulate human tutors behavior and guidance. It can assist students studying a variety of subjects by posing questions, parsing responses, and offering customized instruction and feedback. In the last 10 years, there has been a significant rise in such online education and tutoring systems. Many educational institutes have started using these systems, like My Pearson Lab and WebAssign, as supplements with introductory college classes to provide college undergrads easy access to labs and other online resources and enable them to have better grasp of core concepts. Though one-by-one, in-person tutoring is effective in guiding students through the learning process, these tutoring systems attempt to capture the best methods derived from the traditional human model and move beyond it to discover new strategies for teaching and learning. Most of them have been proven to be quite successful.

A few institutions have developed their own systems. Such as the Cognitive Tutor, developed at Carnegie Mellon University. This system has been widely implemented in several levels of math and science nationwide, from algebra and geometry for secondary schools to the Genetics Cognitive Tutor that helps Carnegie Mellon students understand such issues as gene interaction and gene regulation. The Andes Physics Tutor, developed at Arizona State University, supports students in introductory physics courses, while the Writing Pal has been tested extensively with secondary school students. Georgia Institute of Technology has their own version of ITS and is used by ECE undergraduates, who are enrolled in an introduction to Digital Signal Processing (ECE 2025/2026) course, as a required homework utility. The idea behind this system is to design, test and use systems to enhance student learning in Georgia Tech courses by applying techniques that include, but are not limited to video and data mining, artificial intelligence, machine learning, and human-computer interfaces. The system currently has 8 assignments that students must complete over the course of a semester. The first seven assignments are based on different concepts and are designed to test students knowledge about core concepts. These can be considered to be pretests for students, which help them identify their strengths and weaknesses in order to prepare them for midterms. The last assignment, assignment number 8, is a consolidation of all the concepts and can be perceived as a posttest for students. This is the assignment that students should ideally attempt while studying and preparing for the final exam.

Theoretically, one of the key features of any ITS is to interpret complex student responses and learn as they operate. The software is supposed to build a profile for each student and estimate students degree of mastery. These kind of systems should be able to alter their behavior in real time, following differences in individual student strategies or adjusting its knowledge base for more effective interaction with all the students. For an intelligent tutor, the goal is not merely to know that a response is incorrect but to recognize where in that response the student has gone wrong. Nonetheless, most of the Intelligent tutoring systems dont adhere to this notion and simply provide the same experience to all the users. For instance, the current implementation of ITS at Georgia Institute of Technology fails to provide a customized or unique experience to students. Though the system collects and mines data on every student, it does not use this data effectively to identify students strengths and weaknesses, and consequently does not tailor itself to help students improve. This limits the student experience as every user gets the same set of questions, irrespective of his or her strengths and weaknesses. Also, the current model used for posttest is ineffective. Even though the last assignment is based on the core concepts covered in previous assignments, it does not facilitate good reinforcement of concepts, as it has over 250 questions and students are not required to attempt all of them. As a result, there is no way to ensure that students completed and learnt from this assignment. Also, students often misuse the skip question functionality that is currently provided. Using unlimited skips, students, without incurring any penalty, often go through all the questions and only attempt the easy ones, hence, seriously undermining the purpose and efficiency of the system.

After identifying these areas of improvements, the Intelligent Review Model was conceived and developed to redress these issues. In its present implementation, the model successfully identifies students strengths and weaknesses, and tailors the last assignment of Georgia Techs ITS, which is the combination of all the concepts learnt throughout the semester, to provide students with comprehensive feedback and give them plenty of questions to address their weaknesses. It also addresses the problem of random question ordering and low student participation, by reducing the number of questions to 10 in the best case and 150 in the worst case.

II. INTELLIGENT REVIEW MODEL

A. System Requirements

The Intelligent Review Models requires an online system with a minimal number of data members. Questions must be tagged with concepts or other identifiers that show relationships between questions and subsequently can be used to classify questions. Users must belong to sets, referred to as class terms, and have their corresponding question performance scored.

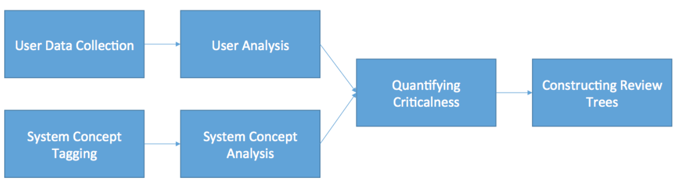

Fig. 1. Intelligent Review Model System Overview

B. System Analysis

1) User Statistics: One of the primary goals of the Intelligent Review Model is to provide a unique individual experience that addresses conceptual weaknesses of a given user. User statistics,

$$us[n, s] = \text{ proficiency of concept n for individual user s}$$

are calculated across all assigned questions and underlying concepts. A limitation of a purely individual user statistic approach are cases where a user fails to complete questions to on a specific concept. Class statistics,

$$cs[n, c] = \text{ proficiency of concept n for class term c}$$

were utilized to as a benchmark to fill in gaps of us[n,s]. User completion rate,

$$C[n, s] = \text{ fractional completion of concept n for user s}$$

and user incompletion rate

$$\bar{C}[n, s] = \text{ fractional incompletion of concept n for user s}$$

$$\bar{C}[n, s] = 1 - C[n, s]$$

are used to find the gaps of data in us[n,s] to replace with cs[n,c].

2) Internal Statistics: The requirement of the concepts bases to be able to make transitions in the Intelligent Review Model gives rise to internal analysis of concepts transitions. A modified transition matrix is defined using the conditional probability that concept $n_j$is tagged given that concept n_i is tagged. The transition matrix, tr, is defined as

$$\mathrm{tr}[\mathrm{n_i}, \mathrm{n_j}] = \Pr(\mathrm{n_j} | \mathrm{n_i}), \mathrm{tr}[\mathrm{n_i}, \mathrm{n_j}] = \begin{bmatrix} \Pr(\mathrm{n_1}|\mathrm{n_1}) = 1 & \cdots & \Pr(\mathrm{n_j}|\mathrm{n_1}) \\ \vdots & \ddots & \vdots \\ \Pr(\mathrm{n_1}|\mathrm{n_i}) & \cdots & \Pr(\mathrm{n_j}|\mathrm{n_i}) = 1 \end{bmatrix}$$

Along the main diagonal of the transition matrix, the conditionally probability of $n_i = n_j$ is equal to one. Unlike a traditional transition matrix based on states, concepts are not mutually exclusive such that the sum over any row or column of the modified transition matrix is greater than or equal to one [1]. The total transition potential, tp[n] is defined as

$$tp[n] = \sum_{n_j} tr[n_i, n_j] \geq 1$$

The total transition potential is utilized as the evaluation criteria of a concepts potential to facilitate a wide variety of transitions. Concepts with relatively high transition potential are strong candidates for bases of review trees because of their flexibility and stability.

C. Quantifying Criticalness

User statistics, class statistics, and total transition potential must be furthered processed to determine criticalness. An underlying assumption of the Intelligent Review Model, is for a system with sufficiently large number of total concepts ($n \geq 40$), us[n,s], us[n,c], and tp[n] are approximately normally distributed. The standard normal distribution is routinely used to process user statistics and class statistics. Markov models (transition matrixes) have also been quantified using standard normal distributions in real time systems [2]. The standard normal distribution with the underlying Z-score and cumulative distribution function (cdf) computations, support the transition from raw system data to quantifying criticalness as a function of standard deviation. The advantages of using the Z-statistic and CDF approach are the natural filtering of outliers in both directions, and the consistency of only a small subset of concepts receiving high criticalness scores. For user and class statistics, right-sided CDFs (cdf(-Z-score)) were used instead of the typical left-sided CDFs (cdf(Z-score)) to prioritize conceptual weaknesses over conceptual strengths for a constant user or class term. User statistics score,

$$uss[n, s] = cdf(-Z(us[n, s], n))$$

and class statistics score,

$$css[n, c] = cdf(-Z(cs[n, c], n))$$

and total transition potential score,

$$tps[n] = cdf(Z(tp[n]))$$

were calculated. Criticalness,

$$cr[n,s,c] = T.(tps[n]) + \bar{T}.(C[n,s].uss[n,s] + \bar{C}[n,s].css[n,c])$$

Criticalness, where

$$T \in [0,1]$$

and

$$\bar{T} = 1 - T$$

is the weighted average of user and internal analysis components. As T increases the upward stability of the binary trees increase and the customized user experience decreases. As T decreases the upward mobility of the binary trees decreases and customized user experience increases.

D. Constructing Review Trees

From the criticalness distribution, for a given student s in class term c, the criticalness scores are sorted into a concept priority list and a remaining question list is created. From the concept priority list binary review trees are built using Figure 1. Based on the size of the system, and the desired number of trees or desired depth, the total number of base concepts are selected. As concepts are selected they are removed from the concept priority list. For each base concept, questions are selected in parallel. For the first level of questions, only one question is selected for each concept. Every subsequent level of questions two questions are selected for each concept. Select questions returns the highest difficulty remaining question for the current concept and removes the assigned questions from the remaining question list. When select question has exhausted all questions for the current concept, a concept transition occurs. Select concepts returns the highest transition matrix value for the n_i = current concept, n_j are the remaining concepts in the concept priority list or the most critical remaining concept if tr[n_i,n_j] = 0 for all remaining concepts. The process of selecting questions and concepts continues until all questions are exhausted and the trees are fully built.

To simplify the explanation of building the model, imagine a set of three concepts, n_1-n_3, pre-sorted by criticalness with one base concept. On the first level of the tree, one question from n1 will be selected (n1 = current concept). On the second level of the tree, two questions from n1 will be selected. This process repeats over the binary tree until all the questions are exhausted for n1. After the questions for n1 are exhausted, tr[n_1,n_2] and tr[n_1,n_3] are compared. The higher transition value is the concept that is selected, $n\prime$ = n3 such that tr[n_1,n_3] $>$ tr[n_1,n_2]. Questions for n3 are selected until they are exhausted, at which point, n' = n_2,the only remaining transition. The remaining questions for n_2 are selected until the tree is completed.

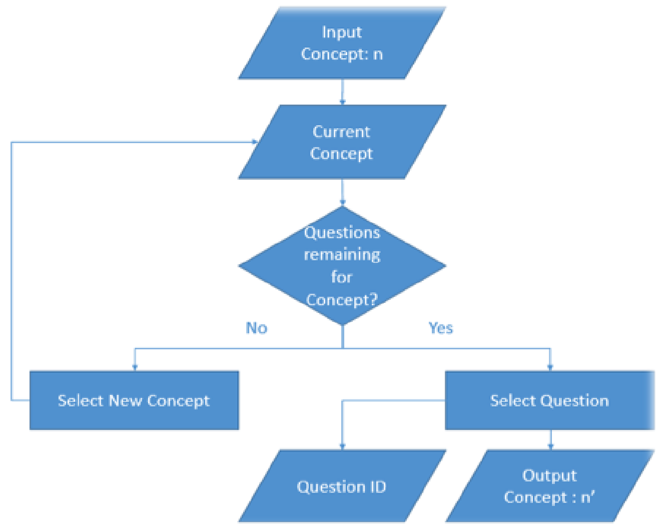

Fig. 2. Simplified concept review tree building algorithm

III. DISCUSSION

While developing the Intelligent Review Model, some major design decisions were taken to safeguard it from failure and make it compatible with real world systems. For instance, 0.5 was chosen as the desired value for T as it guaranteed a binary tree structure for the selected difficult concepts, while taking into account the strengths and weaknesses of every student. If value of T had been 0, the model would have discounted all user information and just built the nearly perfect transition trees based on the transition potential of concepts. Consequently, that would lead to a system devoid of any personal user experience and thus undermine the entire model. On the other hand, a value of 1 for T would have ignored transition potential and only accounted for students strengths and weaknesses. Though a tempting alternative, the resulting trees may have poor concept-to- concept transitions and structure, thus making them highly unreliable to be used for generating assignments. Also, it was decided that the model would generate n trees, based on n potentially critical concepts, to mitigate the possibility of failure. For instance, consider an online tutoring system with very similar and highly inter-related concepts. If the Intelligent Review Model were to be used on such a system, with relatively few concept bases, there exists a chance that all concept bases are from the same concept cluster. In such scenario, the Intelligent Review Model would quickly exhaust potential transitions, and lead to catastrophic transitional failure across all branches.

The preliminary implementation of the Intelligent Review Model has been tested on ITS and another team is currently working on designing a complimentary adaptive tree-like graphical user interface. The model is scheduled to go online during Spring 2015 to enable students enrolled in Introduction to Digital Signal Processing course in 2015 to use it to their advantage.

IV. RESULT

The Intelligent Review Model aims to address the problem of generic user experience in online tutoring systems. Utilizing different data mining techniques, the model is able to identify users strengths and weaknesses and consequently tailor assignments to give students a chance to address their weaknesses and effectively improve their conceptual knowledge. The purpose of this model is to provide the ability to tailor any online tutoring system by accounting for the differences in students performance and consequently increasing these systems effectiveness. In its current implementation in ITS, it effectively redresses the problem of low posttest completion rate and random question ordering and also provide students with comprehensive feedback about their performance.

APPENDIX

ACKNOWLEDGMENT

REFERENCES

[1] Constantino-Gonzlez et al. (2000) Constantino-Gonzlez, D. Daniel, Suthers Coaching Web-based Collaborative Learning based on Problem Solution Differences and Participation International Journal of Artificial Intelligence in Education, 13 (2000), pp. 324333

[2] Constantino-Gonzalez, et al. (2001). Coaching Collaboration in a Computer-Mediated Learning Environment. International Journal of Artificial Intelligence in Education. University Maastricht, Maastrict, the Netherlands.

[3] Hall and Gordon, (1998). A collaborative learning environment for data modeling. Proceedings of the Eleventh International Florida Artificial Intelligence Research Society Conference. 158-162.

[4] Hockemeyer and Albert, (1999). The adaptive tutoring system RATH. In: Auer, M.E., Ressler, U. (eds.) ICL 1999 Workshop Interactive Computer aided Learning: Toolsand Applications.Carinthia Tech Institute, Villach, Austria.

[5] Peter Brusilovsky, (1998). Adaptive educational systems on the World Wide Web. In: Ayala, G. (ed.) Proc. of Workshop Current Trends and Applications of Artificial Intelligence in Education at the 4th World Congress on Expert Systems, Mexico City, Mexico, ITESM, 9-16.

[6] Suraweera, 2001 Suraweera, Mitrovic (2001, 2004). An Intelligent Tutoring System for Entity Relationship Modelling. International Journal of Artificial Intelligence in Education.Volume 14, Numbers 3-4, 375-417

[7] Rodica Mihalca and Anca Andreescu, (2008). Knowledge Management in E-Learning Systems. Revista Informatica Economic nr.2 (46), 60-65

[8] A.T. Corbett, J.R. Anderson, K.R. Koedinger, (1997). Intelligent Tutoring Systems. Handbook of Human-Computer Interaction, Second, Completely Revised Edition, Chapter 37

[9] K. VanLehn, C. Lynch, K. Schulze, (2005). The Andes Physics Tutoring System: Lessons Learned. International Journal of Artificial Intelligence in Education, 15(3)

[10] Angela L. Barrus, Kent E. Sabo, Stacey S. Joseph and Robert K. Atkinson. Evaluating Adaptive, Computer-Based Mathematics Tutoring Systems: A Math Improvement and Feasibility Study

[11] Carole Beal, Sinjini Mitra and Paul R. Cohen Modeling learning patterns of students with a tutoring system using Hidden Markov Models