Graph-based Adaptive Diagnosis

Yajie Jessica Wang
Amit Sahai, PhD
UCLA

Abstract

The estimation of student ability is often based on numerical test scores. Though sometimes reflective of a student's capabilities, these scores reveal nothing about what the student is good at and what he or she needs to improve. The purpose of this research is to create a diagnostic test that uses more than just the ordinary means of assessment to test students in the area of math. By using a graph-based implementation that connects related subtopics or skills, we create and test a diagnostic system that is more effective in identifying students' areas of knowledge that are lacking in the general field of mathematics. After implementing the evaluation program in C++, we found that it correctly identities the boundary between topics that the student has mastered and those he or she has not. Though the system sometimes places the student in a region of mastery that is lower than where he or she actually is or has reported, we come to the conclusion that this situation only comes up when the student makes many mistakes (careless or otherwise), which is a reasonable interpretation of student ability. Because this model of evaluation offers both quantitative and qualitative results, we believe that its idea could very well benefit our current education system.

Introduction

Oftentimes students struggle in school and teachers are unable to find ways to help them improve. Though this problem might have to do with study skills and classroom size, we feel that it also has to do with the teacher's ability to identify students' problem areas. We understand that it is very difficult for teachers to personalize lesson plans and give all students individual attention. Thus, only in a very ideal situation would they be able to identify where students need help and provide lesson plans that are address everyone's needs. If teachers are able to identify what students need help on, learning and teaching can be made that much more efficient. One way of helping both teachers and students is to give students a more analytical test rather than a regular pen and paper examination. In an attempt to improve the testing methods employed by most education institutions, we want to create a program that implements an adaptive test based on graph traversal. We want to compare the results of our adaptive test with how well students are or feel they are doing in their mathematical studies. To create a test that is reflective of the student's capabilities, there are several issues we have to consider. First, how do we create a test that provides unambiguous results that we are confident in? Can we tell the difference between students that are guessing or actually answering correctly? How many questions can we ask before students get tired and yet derive qualitative results from their answers? How do we determine whether a student has mastered something – a point-based system or something more general? These are some of the questions we will try to address in our adaptive diagnosis.

Approach

This research is based on both adaptive and diagnostic testing. In the area of adaptive techniques, existing knowledge has already been developed in terms of the CAT (Computerized Adaptive Testing), a testing procedure that adapts to the ability of the examinee. CAT implementations vary from early pen-and-paper versions (flexilevel testing that only use the correctness of the current response) to more sophisticated, computerized tests that utilize the examinee's currently ability estimate as well as difficulty and discrimination of the questions. From a study comparing an IRT-based (Item Response Theory) CAT, which uses a mathematical model to estimate ability and select items, with the flexilevel CAT, it

was observed that if one only wished to rank order the performances of those who took the test, it would not matter if the examinee were given the flexilevel CAT, a conventional test, or the IRT-based CAT (De Ayala, Dodd, and Koch, 1990). However, this research aims to do more than just perform a perfunctory assessment of student ability and ranking.

In addition to being adaptive, our test also incorporates elements of diagnostic testing, whose distinguishing feature is self-referencing, meaning the student provides his or her own reference point (Bejar, 1984). Using techniques in the areas such as deficit assessment (weaknesses of the student) and error analysis (the kinds of errors students make), previous diagnostic procedures included asking students to identify as many other choices possible thought to be incorrect (Coombs, Millholland, and Womer, 1956). One previously published applications of diagnostic testing was by Thissen (1976), who created an implementation using Raven Progressive Matrices (which is comprised of matrices to test observational skills and clear-thinking ability). By using 570 junior high students as research subjects and conducting the procedure using a slide projector, Thissen was able to conclude from his analysis that such a model is primarily effective for individuals with lower ability levels or very difficult tests. Though our procedure is similar in that it also offers a test to students, the aim of our study is to find a method that would be effective regardless of the level of the student.

A more similar approach to the current research study is the adaptive diagnostic system for fractions developed by Marshall (1980). Marshall used nodes to represent each subskill or way to solve a problem and linked the subskills that could be used together. His test involves estimating performance on each subskill identified as necessary to solve a problem and developing a profile for each examinee. After estimating the individual's performance on the various subskills, the test generates specific problems to test these estimates. Our proposed implementation is similar to that of Marshall's but differs in the way the nodes are linked and traversed. We seek to find a new and innovative way to represent an adaptive assessment using graphs. If paired with a graph that accurately represents the relationships between mathematical subtopics, this test model might prove to be very efficient and insightful. Thus, our evaluation system might be better than those before because we take into account both student response and, like Marshall, topic relationships in our implementation Though much knowledge and theory exists in this field, not many have published implementations or research studies regarding diagnostic assessment. We feel that the scientific and education community can most certainly benefit from the work we are trying to achieve.

Implementation

Using C++, we created a Graph class that utilizes an Edge class and a Node class. The Graph class is responsible for reading input and output. When given the location of a folder, the program reads in two text files specifying the locations of all the library files and files containing the sample problems. It creates a vector of nodes that represent the various subtopics. Each node contains an array of edges as well as a vector of questions that it can ask. Nodes also contain data members that record a numerical skill level, a Boolean mastery marker, a weird-instance indicator, and a numerical count of questions answered correctly. In this implementation, we stress the idea that nodes are only connected if they are skills absolutely necessary for the next skill node. For example, one-digit addition is connected to two-digit addition. If a student cannot add one-digit numbers, it is highly unlikely that he or she will be able to add two-digit numbers. Nodes are divided into various regions, which are based on difficulty. Currently, the graph contains six regions: basic arithmetic, advanced arithmetic, algebra I, geometry, algebra II, and trigonometry. Each region contains two critical nodes (topics that require the largest number of other skills (nodes) in the same region or in other words, those that best summarize the region).

The traversal algorithm is organized into two parts: the first finds the boundary between two regions or the region that contains the boundary and the second explores the specified region using parallelism. Part one of the traversal algorithm starts off with the lowest (easiest) region. It moves to a higher or more difficult region if the current node is considered to be mastered. It moves to a lower region if the node is not mastered. Mastery is determined by the number of questions the student answers correctly: a topic is considered mastered if the student answers both questions correctly and incorrect if one or less is answered correctly

(or in the event that the student declares option e: he/she does not understand the topic/question at all). Traversal between regions is done mostly through critical nodes. Our reasoning is such: if the student is able to master the critical topics of the region, it is highly probable that they have mastered the entire region. There are several cases in which the current node has already been visited. First, the boundary might be already defined (and checked through a bound marker). In this case, the function returns the region below the boundary and sets the bound marker. Another case would be: both critical nodes of the region have been visited already, which would then prompt the function to find an "easy" question (nodes with no edges that move backwards) in the region. Evaluating the student on easy questions can reinforce the algorithm's conclusions on the student's skill levels or dispel uncertainty in its evaluation. If both easy nodes (2 defined specifically for each region) and critical nodes have been visited and results are inconclusive, the function will return the number of the region for the second part of the algorithm to explore.

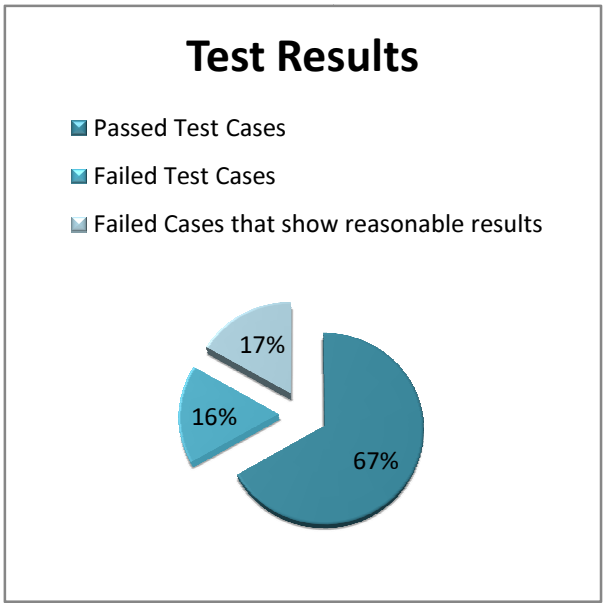
The second part of the algorithm uses several concepts: parallel traversal and trickle-down estimation. It first finds the deepest unvisited nodes (ones with no forward edges) and places them into a vector. Each node in the vector is visited and evaluated. If mastered, all lower connecting nodes are set to mastered as well (after all, these skills are necessary for the highest skill). If the skill is a weak area for the student, the algorithm will replace the node in the current position in the vector with insertions of all backward-connected nodes. Thus, the algorithm will traverse in a breadth-first fashion through the specified region. Other features include a weird marker that is set in the case that anything unpredictable is detected, such as an unmastered node in between two mastered nodes. After effectively exploring the region, the function returns the region explored as well as the condition of each of the nodes in the region: mastered or not mastered.

In testing our implementation, we ran into a few snags that caused us to change our original plans. We had initially intended on testing the program on students in middle school (those taking algebra I). To be able to do so (in accordance with UCLA policy), we applied for and attained approval to conduct a research study on seventh and eighth grade students at Pio Pico Elementary school. In this research study, students would first take the computerized adaptive test and then answer a survey that would gauge their skill levels. After approval, we proceeded to apply for permission from the LAUSD (Los Angeles Unified School District). Unfortunately, the LAUSD's rejection of our proposal occurred two to three within the completion of the school year and the end of the research seminar. Consequently, we were unable to rewrite our proposal and apply once more. Thus, the data generated in this research is solely based on logical reasoning, which might not be a precise portrayal of how children cogitate. Our new methodology includes testing a series of situations and verifying whether the results of each test case match our predicted outcomes.

Results

For the majority of the test cases, we were able to achieve results that matched our predictions. The first part of the algorithm almost always determined the predicted region correctly save for one situation: when the student makes many mistakes. In that circumstance, the algorithm underestimates the student's ability; that is, it believes the student's skill level is lower than it actually is. For example, if the testee, who is at an Algebra I level, answers 1/3 of all questions incorrectly, instead of placing the student in Algebra I, the system places the student in basic math. However, when the student only answers 1/5 of all questions incorrectly, the system is still able to produce accurate results. Here, we face the issue of whether to underestimate or overestimate the student. Our reasoning leads us to believe that it is more logical to underestimate students' abilities, especially if it is the analysis derived from answering many questions incorrectly. After all, if a student is answering that many questions incorrectly, it is probably adequate to say that the student doesn't know the topics as well as he or she thought or has trouble with careless mistakes. Through our testing, we found that the system handles student guessing rather well: it is able to produce the correct results when the student has a low guessing success rate (20%). Results are more inaccurate when the student successfully guesses at a higher rate (such as 25 – 50%) and cannot answer all questions from mastered regions correctly.

Test Cases	Times Tested	Times Passed
All questions incorrect:	1	1
All questions correct:	1	1
Understand up to Algebra:		
a) All questions right	2	2
b) Missing 33% of questions in mastered regions (analysis makes sense however)	2	0
c) Missing 20% of questions in mastered regions	2	1
Understand Algebra and Algebra II but not Geometry:	1	1
Guessing correctly:		
a) 50% of questions guessed correctly in unmastered region	2	2
b) 25% of questions guessed correctly in unmastered region	2	2
c) 25% guessed right with 20% of mastered questions answered incorrectly: (analysis makes sense however)	2	0
d) 25% guessed right with 33% of mastered questions answered incorrectly: (analysis makes sense however)	2	0
Difference between not knowing and guessing:		
a) Guessing/33% right	2	1
b) Option e	2	2
Region Testing:		
a) Everything not mastered	2	2
b) Everything mastered	2	2
c) Parallelism	3	3
d) Weird possibilities	2	0
Confidence matching:	6	4



In terms of efficiency, we found that the test is most efficient when all questions are answered correctly (then the test would, at worst, have to ask around 2 questions per region and 5 questions in one region, assuming the region has at most 5-7 questions without forward edges). The worst case occurs when the student goes through all the regions, the results are inconclusive, and he or she answers incorrectly in the region exploration algorithm. With the current size of the graph, the worst situation asks around 30 questions. On the average case, it asks about 20 questions, which is the size of a normal math test. Also, we must keep in mind that these questions are rather short and computationally-simple.

One component that we are unsure about, both implementation-wise and testing-wise, is the confidence estimation. Though we currently compare the numerical skill estimation with the more general mastered Boolean value to see whether our results are reliable, it is difficult to truly depend on those results, which might be biased. Without actual data from test subjects, it is difficult to calibrate exactly how many points are appropriate to subtract when questions are answered incorrectly. What would be the difference between receiving a

40 and a 60? Both mean F's in the customary way of grading but is that what we want to represent how well a student is doing in a topic? Currently, our results prove to be inconclusive, something that can hopefully be addressed in the future.

Conclusion

We come to the conclusion that our implementation is an effective model that, with a few more features, can become an insightful testing model. With an accuracy of 67% and 50% of the mismatches making logical sense, our diagnosis is able to identify the student's weak and strong regions relatively well. Its shortcomings include not returning definite matches that indicate the reliability of the diagnosis's results. If provided with future opportunities, there are several more tasks we would like to see done. Firstly, we would like to improve the implementation in the aforementioned area (increase confidence in the test's results) as well as create an interface that would display the results visually. We would also like to gain a wider range of test data by working with student test subjects. Copies of the source code for this research project can be obtained at:
www.brightenworld.com/jessica/sourcecode.zip
The executable and all necessary files to run the diagnosis can be obtained at:
www.brightenworld.com/jessica/research.zip

Acknowledgements

We thank Professor Amit Sahai for leading the CS 194 Seminar and making this research possible. I personally thank him for being my advisor and providing both sound and innovative advice. We also thank the North General Institutional Review Board of the UCLA OPRS (Office for Protection of Research Subjects) for approving our research study.

References

Bejar, Isaac I. "Educational Diagnostic Assessment." Journal of Educational Measurement 21 (1984): 175-189.

Coombs, C. H., J. E. Millholland, and F. B. Womer. "The Assessment of Partial Knowledge." Educational and Psychological Measurement 16 (1956): 13-37.

De Ayala, R. J., Barbara G. Dodd, and William R. Koch. "A Simulation and Comparison of Flexilevel and Bayesian Computerized Adaptive Testing." Journal of Educational Measurement 27 (1990): 227-239.

Marshall, S. P. "Procedural Networks and Production Systems in Adaptive Diagnosis." International Science 9 (1980): 129-143.

Thissen, D. M. "Information in Wrong Responses to the Raven Progressive Matrices." Journal of Educational Measurement 13 (1976): 201-214.