Closing The Loop on Test Creation

A Question Assessment Mechanism for Instructors

Titus Winters
Computer Science & Engineering Department
UC Riverside, Riverside, CA
92521
titus@cs.ucr.edu

Tom Payne
Computer Science & Engineering Department
UC Riverside, Riverside, CA
92521
thp@cs.ucr.edu

ABSTRACT

New accreditation requirements focus on education as a “continuous improvement process.” The most important part of such a process is that information gets fed back into the system to improve the quality of the output. This requirement is often interpreted to mean a feedback loop that iterates on offerings of courses or entire academic years. This paper provides a smaller and more immediate feedback loop. This technique gives instructors feedback on the quality of each question on a test or quiz, as well as a numeric score for the difficult of the question. A simple tool implementing this procedure can be used to help train instructors on which questions are difficult, as well as what types of questions are correlated with ability, and how to design a meaningful instrument of assessment. Performing this analysis at the end of a course offering could help demonstrate continuous improvement to accreditation committees. Performing this analysis immediately after the administration of a test or quiz can point out topics that the class as a whole have failed to understand, thus giving instructors more insight into student knowledge.

Categories and Subject Descriptors

J.1 [**Computer Applications**]: Administrative Data Processing—*Education*

1. INTRODUCTION

In the 1990’s the ideas of “Total Quality Management” and the “Continuous-Improvement Process” swept through the manufacturing and engineering communities. These concepts share a conceptually simple idea: rather than blindly continuing with the manufacturing process as usual, set goals and take measurements throughout the process, and alter the process itself to address discrepancies whenever the goals are not being met. From a system-design standpoint, the core concept is to introduce feedback mechanisms where once there were none.

The educational process is much like a manufacturing process: there is a continuous stream of outputs (students) for which there definite quality goals (what they should know or be able to do). EC2000[1], the ABET accreditation criteria under which many CS programs in the United States are now evaluated, requires that instructors no longer blindly proceed on gut-feelings regarding what worked and what did not, but instead move toward more quantitative measures of educational effectiveness. Total Quality Management is now being applied to students.

However, EC2000 does not give strict instructions on how to perform this task. In fact, it does not even give concrete examples of systems that meet these requirements. It falls to the instructors and curriculum committees for each ABET-accredited institution to come up with a system that works for them. In many cases, the feedback loops that are being introduced are high-level: the workings of each course as a whole are examined, commented on, and reported for the next instructor of the course in the hope that the same big mistakes will not be repeated. If there is sufficient “buy-in” on the part of the instructors involved, this has the potential to effect long-term change. Courses will become more similar from offering to offering, and better calibrated to the ability levels of the students coming into the course. Continuous improvement, if accepted, can make better teachers of all of us.

There exist other levels in which a feedback loop can be useful when teaching a course. This paper introduces an assessment mechanism suitable for driving low-level feedback loops. This answers the questions, “How difficult was question X?” and, “How much do scores on this question really tell me about what students know?” It produces numeric estimates of the difficulty and discrimination of a given question, on a known scale, that can be easily compared. Most importantly, it is instructor agnostic: this method in no way compares your style of questions against some theoretical perfect question or a perfect instructor. Questions are assessed relative to your own teaching style. Under certain assumptions, we can also show that these estimates are relatively stable across quarters, allowing more accurate tests to be generated from banks of existing questions.

The rest of this paper is divided up as follows: Section 2 briefly presents terms and notation we will use, as well as some basic concepts from educational statistics. We then build upon these concepts to show how to evaluate questions in Section 3. Examples from real student score data are presented in Section 4. Section 5 presents our final conclusions.

SIGCSE 2006 2006 Houston, Texas USA

2. NOTATION AND BACKGROUND

To generalize away from saying “test, quiz, or assignment,” we use the term *instrument* or *instrument of assessment* as shorthand for the graded work to be assessed. For simplicity, we assume that each question on the instrument is graded as either right or wrong, with no partial credit. The techniques presented here generalize to partial-credit questions with little difficulty.

Within the realm of educational statistics and computer-aided assessment, one of the most powerful tools is Item-Response Theory (IRT)[2], the power behind such tests as the computer-based GRE[4]. While IRT is primarily useful for situations where hundreds of students are attempting the same questions, some of the concepts from IRT are useful on classroom scales. IRT specifies the goal of assessment to be estimating (on some numeric scale) the level of aptitude for each student with respect to a certain topic. Obviously this value is not something that can be directly measured: no meter stick, balance, or mass spectrometer can tell how much a student knows about linked-lists. This score must be measured indirectly for each student, and is known as a “latent trait” or simply their current ability level for that topic. Each question is assumed to have a “characteristic curve,” which is a plot of the probability of getting the question right as a function of the student’s ability in the topic governing the question. We are not concerned with the specific equations describing these curves. Most of the common equations incorporate two parameters: β, the difficulty, and α, the discrimination. Intuitively, β governs what ability level the student must have to have a good chance of answering correctly. α corresponds inversely to the probability of someone with ability less than β getting the question correct or someone with ability greater than β getting it wrong. Thus β is how difficult the question is, and α is how meaningful the scores are.

IRT focuses on using question scores and known α’s and β’s to estimate student’s abilities. We already have a rough estimate of student’s ability: their (numeric) course scores. What we would like is a simple method to determine α and β for each question.

3. GOOD QUESTIONS

Given the vector of student scores for every question and the vector of course scores, we can easily evaluate which questions are actually correlated with high ability in the course, and estimate the difficulty of each question. We can then begin to learn which types of questions are most meaningful (have a high α) and which are not. This refinement of questions is based entirely on how your course is managed, thus allowing you to know how good a question is with respect to what you care about measuring in your students. Questions that are perfect for one instructor may not be perfect for another, although in general they should be correlated.

Knowing the β for the questions can give you insight into what the students are really understanding. If after evaluating a question the β seems too high, it indicates that the topic is not well understood by the students. Generally one can assume that a question with a high α and a β in the useful range (the range of course scores that would pass the course) is a “good question.”

3.1 Calculating α and β

To see how calculating the difficulty and discrimination parameters can be done easily, first assume β is known for each question. It is then possible to find the empirical discrimination for a question by evaluating how well that β separates students that got the question right from those that got it wrong.

There are a number of ways of evaluating this, and in general they give similar results in most cases. We have evaluated complex techniques like entropy-based measures, but in general the simplest techniques works well, with near-identical results. That technique is to calculate the percentage of scores that would be guessed correctly under the assumption that every student with ability under β got the question wrong and every other student got it right. If it really is the case that β is a perfect split point, then α will be 1, a perfect score. If correct and incorrect scores are evenly distributed on both sides of β then α will be .5, and if somehow the question were completely backward (only students with ability less than β answered correctly) this results in a score of 0. Nearly any similarity measure developed for data-mining / machine-learning “decision tree” algorithms can be adapted to work here. Interested readers are invited to peruse that literature to find other techniques[3].

Given this ability to produce α given the question scores, ability scores (course scores), and β, it is now easy to find the actual β. The best estimate of β is the value of β that maximizes α. Since α depends only on which scores were guessed correctly, it is sufficient to only loop through each distinct course score and evaluate on those split points, avoiding any gradient optimization methods. The procedure to produce α and β for each question can be implemented in 100-150 lines of code, and is available from the main author’s website.

3.2 Assumptions

In order for the above to be usable, several assumptions must be made. First of all, we are assuming that the question has some relevance to the general topic of the course so far. Nobody can track the true ability level of the students; the use of course scores is only an estimate. If those scores are highly uncertain (at the beginning of the course) or have nothing to do with the question, then this is not a valid assumption. This implies that each course is a single proficiency. This is clearly not usually true, but for any course that this is a bad assumption (for example, a course composed of two half-courses on different topics), it should be possible to use the scores from only the appropriate portion of the course. It would be ideal to automatically extract topic information from the score data and track student proficiencies on a fine-grained level (ability with linked-lists, rather than score in the Data Structures course), but this remains an area of future work.

The estimates of both parameters rely heavily on the distribution of the course scores. Student grades fluctuate significantly at the beginning of the course before the law of large numbers begins to stabilize each student’s grade toward its final value. Therefore, the values of α and β are going to be most accurate at the end of the course, and a final evaluation of which questions are worth keeping for next quarter is best done after the course has completed. It is useful to evaluate questions immediately after grading the instrument, especially to find topics that were tested but not fully understood, but such evaluations should be recognized as less-accurate estimates.

Independence of scores is also a concern. If the estimate of ability (course score) includes the score on the given question, then by definition there is some correlation between the two. To get the best estimates of α and β, it is best to provide ability estimates that do not include the score on that question or instrument. In practice, if each individual question has a very small effect on the total grade, then this effect is negligible and few questions need to be evaluated separately.

Finally, it should be pointed out that the course grade cannot take into account the validity of each question. If course scores are adjusted to de-emphasize low-discrimination questions and over-emphasize high-discrimination questions, then you are, on average, barring anyone from changing their overall course standing. If scores were re-weighted based on discrimination, then a student that initially performed poorly and studied extra hard would decrease the discrimination on those questions that they got correct due to extra study, and a similar argument can be made for good students slacking off. This is a technique for assessing questions, and the intent is for this to be utilized to make instructors more aware of the questions they are asking. If this is used to adjust scores, especially *as the courses progresses*, early course grades become more difficult to overcome for students that get off on the wrong foot.

4. RESULTS

This technique was used extensively during Summer 2004 in Introduction to Data Structures and Algorithms, where it provided very useful feedback to the instructor in evaluating the questions on three 10-Question quizzes, the 34-Question midterm, and the 34-Question final examination. Some of the intuitive insights it confirmed included:

- Do not test on specifics from the text: Questions that test not the subject, but merely a particular presentation of the subject have little assessment value if that presentation isn't a major component of the lecture. (Some instructors may choose to explicitly test this material to encourage students to do the required reading, but that was not this instructor's goal)

- Do not test things mentioned in passing: While some may feel that mentioning something once or twice during lecture is sufficient for the "good" students, this appears not to be the case. Iterators were discussed as an aside for 10-15 minutes the day before Quiz 1. This was not nearly enough for even the good students to pick up the concept. Although 47% of the class answered this question correctly, it is likely because they narrowed it down to 2 or 3 choices and guessed rather than knowing the answer, since this question had discrimination of 0.59, implying nearly zero correlation with general course performance.

The most useful feedback this technique provided was in finding questions with unexpectedly high difficulties. The most striking example was for the question, "What is the worst-case time to find an element in a binary search tree of n nodes?" While there is a mild "trick" in recognizing that a BST is not necessarily balanced, it was shocking to find that the difficulty score for this question was 87% (with a high α), meaning that it did a better job of differentiating A's from B's than anything else. This was certainly not intended to be a question splitting the A's from the B's. The fact that it was found to be quite difficult indicated that this aspect of binary search trees was a topic that needed additional coverage. Since the quiz was graded and evaluated the same day it was issued, the lesson plan for the next day was altered to account for the fact that the class had missed this important concept.

This technique can provide immediate feedback in two important ways. First, it allows instructors to determine which questions are good questions and can be used to train instructors not to ask overly vague questions. Second, and more importantly, it can provide concise feedback on what students are actually understanding and what they are only guessing. In this way, topics that were confusingly or incompletely covered can be immediately brought to the instructor's attention and remedied as quickly as possible. Both of these seem to be valuable tools in the effort to increase educational effectiveness.

4.1 Cross-Quarter Consistency

This technique also allows databanks of questions to be assembled along with an estimate of the difficulty associated with those questions. There are a few caveats here: the difficulties are likely very dependent on instructor, are obviously dependent on course, and are hopefully dependent on what time during the course the question was asked. For example, if one instructor uses a question on a quiz early in the quarter in one quarter and on the final exam in the next quarter, it is hopefully the case that the students will have solidified the skills necessary to answer that question, so the empirical difficulty will decrease. (This is a deviation from IRT, where question parameters are absolute but student abilities are generally increasing throughout the term.)

To demonstrate the validity of this statement, we have identified 20 questions that were repeated between Fall 2004 and Winter 2005. Of these, nearly half were dramatically too easy, with difficulty levels in the D- range or lower (in this range, the density of student scores is too low for predictions to be very accurate, without enormous class sizes). After ignoring anything with a difficulty less than 65%, we are left with 12 questions ranging from difficulties of 66% to 95%. On average the difference between difficulties between quarters is 5%, with a standard deviation of 4%, meaning that more than 2/3 of the time, a repeated question will have a difficulty within one letter grade when used under similar circumstances.

Table 1 provides empirical difficulty and discrimination values for several "good" questions, and Table 2 provides the same for some questions that are particularly bad. Note that for the "bad" questions it is often the case that the question was poorly worded or ambiguous in some way, or is a trick question. This matches our intuitive concept of discrimination perfectly: for questions that are subtle or textually confusing, the odds of a good student getting it wrong are much higher than they would be otherwise.

5. CONCLUSIONS

Every student attempt on every question is fundamentally testing two things: how good is the student at that point in time, and how good the question is. In normal teaching, we evaluate students continuously, and with the aid of the techniques presented here we can begin to evaluate the

Question	Difficulty	Discrimination
Quicksort has a worst case running time of $O(n \log n)$	80%	.72
Mergesort cannot be used efficiently in place	76%	.78
The following code will do what?	93%	.71
What is the representation of -1 in 4 bit 1's complement?	77%	.77
How many values can be represented by a 4 byte binary word?	95%	.81
AND and OR are two binary logic operators. How many binary operators can be defined?	85%	.72
What is the result of NOT(1000 AND (1100 OR 0101))	67%	.81

Table 1: Sample Difficulty and Discrimination Values for Good Questions

Question	Difficulty	Discrimination
(OS) Thrashing may not occur on a system with a two-level scheduling policy.	100.6%	.58
(OS) Which of these may block? (printf, strlen, malloc, free, getpid)	90%	.61
(OS) If *Foo* a* and *Bar* b* are pointers to objects allocated in a shared-memory space, and process A accesses a while process B accesses b without using mutual exclusion, there will be memory corruption (T/F)	77.5%	.67
(CS1) All return statements must return a value (T/F)	94%	.54
(CS1) What is the result of 20.0%8 in C++?	97%	.55

Table 2: Sample Difficulty and Discrimination Values for Bad Questions

questions as well. Evaluating educational effectiveness is an important concept in current accreditation practices, and we expect that importance to grow in years to come. The techniques presented in this paper provide a simple method to close the loop on the lowest-level of a course feedback process by reinforcing which questions are most meaningful. Building a catalog of good questions with known difficulties now becomes an easy thing for an instructor. Given the ability to estimate with confidence ahead of time how difficult a give question is, it is easier to build tests that more accurately assess the students. Most importantly, a quick analysis of the scores right after administering an exam can yield immediate feedback to the instructor regarding what topics are being misunderstood, allowing for quick alterations to the lesson plan to fix misconceptions before they get out of control. Continuously improving the quality of questions with these techniques is a low-effort, high-reward strategy that can help all of us.

6. REFERENCES

[1] Accreditation policy and procedure manual. http://www.abet.org/policies.html, November 2002.

[2] F. B. Baker. *Fundamentals of Item Response Theory*. ERIC Clearinghouse on Assessment and Evaluation, 2001.

[3] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 11(1):81–106, 1986.

[4] C. A. G. Wim J. Van der Linden, editor. *Computerized Adaptive Testing: Theory and Practice*. Kluwer Academic Publishers, July 2000.