A Content-Balanced Adaptive Testing Algorithm for Computer-Based Training Systems

Sherman X. Huang

Alberta Research Council
6815, 8th Street, Calgary, Alberta, Canada T2E 7H7

Abstract. There are two main obstacles that make use of adaptive testing in computer-based training difficult. One is the requirement of conducting a large-scale empirical study for item calibration. The other is the difficulty of generating content-balanced tests that meet the goal of the test administrators. In this research, we have developed a new adaptive testing algorithm, CBAT-2, to provide a solution for these problems and some other practical problems in adaptive testing. CBAT-2 generates questions based on the portion of the course curriculum that meets the goals of a test. It uses a simple machine learning procedure to determine the item parameter values.

1. Introduction

Testing is an important component of training. It is important for a computer-based educational system to contain a high-quality on-line testing component. An *adaptive test* is a computer-administered test in which the presentation of each test item and the decision to stop the test are dynamically adapted to the student's performance in the test.

Research in adaptive testing is usually oriented to large-scaled standardized tests designed by testing centres such as Educational Testing Service (ETS). The algorithms developed require a major empirical study in order to calibrate the test item pool (Wainer, 1990). Such an empirical study is rarely affordable for smaller schools or industrial organizations that offer on-line training for their students or employees. This difficulty prohibits adaptive testing algorithms from being used in computer-based learning environments.

Another major difficulty for applying adaptive testing algorithms is the content-balancing problem. Most adaptive testing algorithms are content-blind. Their question selection strategy does not take into account from which content areas in the curriculum the questions come. However, a test designer or instructor usually has a plan for the test to cover certain content areas. If the testing algorithm does not address the content-balancing problem, the test would not be able to meet the goals of the plan.

The goal of this research is to develop a *content-balanced adaptive testing* algorithm for computer-based training systems. The algorithm, called *CBAT-2,* is aimed at meeting the growing demand of computer-based education and just-in-time training provided by schools and industrial organizations. In particular, it generates tests that cover content areas in the test designer's plan, and eliminates the requirement for an empirical study to calibrate test items. From our experience in developing commercial adaptive learning software, we have seen the great potential of incorporating CBAT-2 into computer-based learning environments.

2. How an Adaptive Testing Algorithm Work

In general, the goal of an adaptive testing algorithm is to increase the efficiency and assessment precision of the test by selecting items that provide the most information about the student and terminating the test as the assessment reaches a precision criterion. An adaptive testing algorithm usually has three important components: a test item pool, an item selector and a proficiency estimator.

The test item pool contains items that may be selected for the test. Each item is characterized by a number of parameters. Three commonly used parameters, as described in the item response theory (IRT), are: the *difficulty level,* the *discriminatory power* and the *guessing factor* (Wainer, 1990). The difficulty level describes how difficult the question is. The discriminatory power describes how well the question can discriminate students of different proficiency. The guessing factor is the probability that a student can answer the question correctly by guessing.

At any time during a test, the algorithm has a temporary estimation for the student's proficiency (usually denoted θ). The item selector selects an item from the test item pool. The selected item is aimed at providing the most information about the student's proficiency. The selection is based on the item's three parameters and the temporary proficiency θ'. An ideal item should have a difficulty level close to the temporary θ', a high discriminatory power and a low guessing factor.

Once the student provides an answer for the selected item, the proficiency estimator calculates a new θ' and its confidence level, based on whether the student's answer is correct or incorrect, the old θ' and the item parameters. If the confidence level of the θ' reaches a designated level, then the test terminates. Otherwise the item selector selects another item for the student, and the test continues.

3. CBAT-2

CBAT-2 is aimed at providing a solution for the following problems in applying adaptive testing to computer-based training environments.

- *Content-balanced:* Ensure that the items selected for the test cover all content areas in the test plan. No content area is over-tested or under-tested.
- *Test item pool calibration:* Remove the requirement for a major empirical study to calibrate a test item pool.
- *Intelligent selection of test items:* Select test items that will provide the most assessment information to increase the efficiency and precision of the test.
- *Security:* Selected test items do not form a pattern. Having a selection pattern may increase the chance of guessing and cheating.
- *Questions in multiple content areas:* Allow a question to be associated with multiple content areas.
- *Two-level assessment:* Provide assessment information for each content area as well as for the global test.

3.1 Content Areas in a Curriculum and in a Test

A content-balanced testing system must be able to associate questions with content areas. This requires a representation of content areas. In CBAT-2, content areas in a course curriculum is represented by a directed acyclic graph called a *curriculum hierarchy*, as shown in Figure 1.

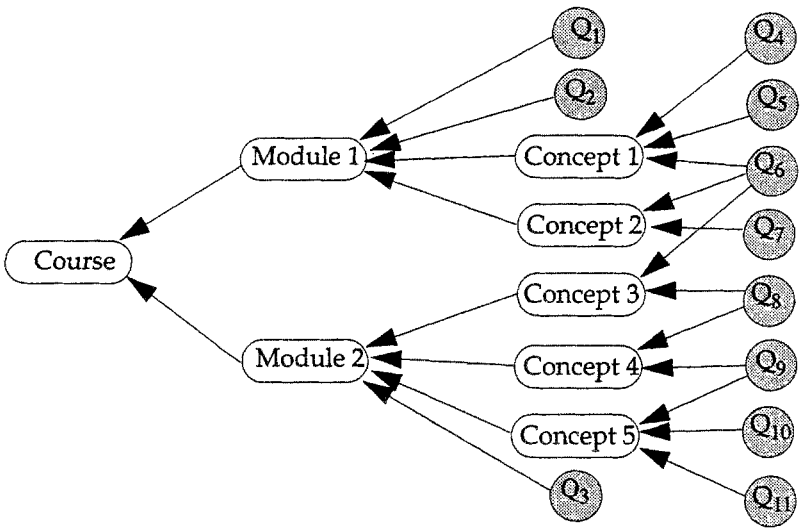


Fig. 1. Content Areas and Questions in a Course Curriculum

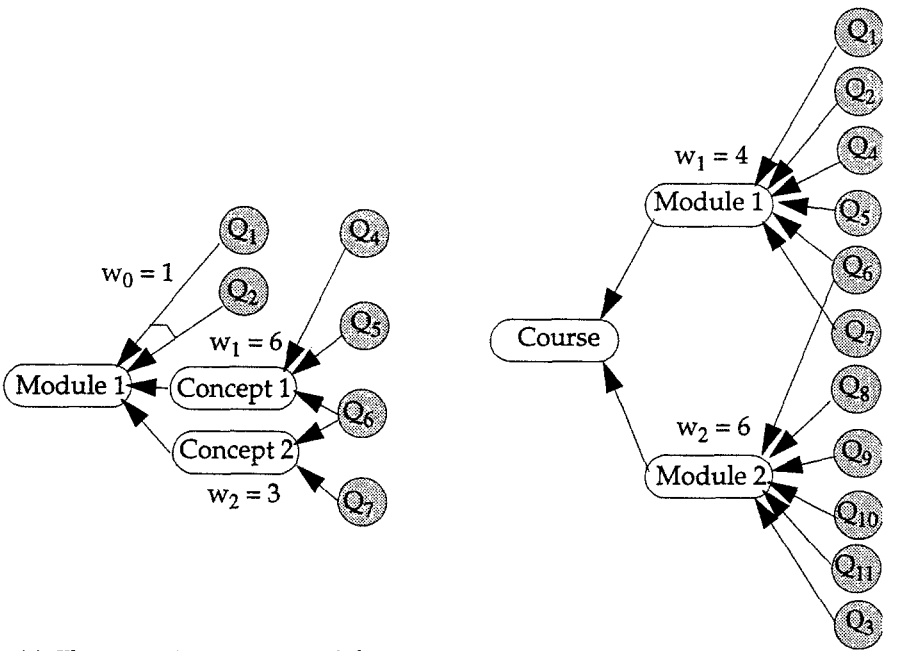
Each rectangle in the hierarchy, called a *component* in general, represents a content area at a certain level. Each component in the hierarchy has exactly one parent, except for the root that has no parent. The root of the hierarchy is the course. A component may have zero or many child components which represent its content sub-areas. For example, in Figure 1, a module is a sub-area of the course; and a concept is a sub-area of a module.

A question, represented in Figure 1 by a solid circle, may be associated with one or many components at any level of the curriculum. Questions associated with components at higher levels are those that require general knowledge. Questions associated with components at lower levels, in particular at the lowest level, are those that require knowledge of specific concepts and skills.

In CBAT-2, a test assesses a student's knowledge at two content area levels. For example, a module test covers a module, and does assessment for the concepts under the module. Only questions associated with this module or its concepts may be selected for the test. A course test covers the course, and does assessment for the modules under the course. All questions in the course may be selected for the test. However, the algorithm is sensitive to only the course-ship and the module-ship of the questions, it is not sensitive to the concept-ship of the questions. Questions associated with different concepts under a module are treated in the same way. They are also treated as the same as questions associated with general knowledge of the module. Figure 2 shows the system's two views of the curriculum in Figure 1. One view is in a module test for Module 1, and the other in a course test.

One might wonder why we don't do the assessment for content areas at all levels in one test. This is because such an overall test is usually too long. In practice, two-level information is usually sufficient. If assessment at every level is needed, then the student should take tests at all levels.

Thus, as a part of the initialization, CBAT-2 generates a *sub-curriculum* for each specific test. CBAT-2 also consults the test designer for the weight of each content area in the test. For example, in Figure 2-(a), general knowledge of the module weighs 1/10, Concept 1 weighs 6/10, and Concept 2 weighs 3/10. By default, all content areas in the test that have questions have the same weight.



(a) The system's view in a module test (b) The system's view in a course test

Fig. 2. Different views of the curriculum in different tests

3.2 Questions

Questions are located at the lowest level of the curriculum hierarchy. They have no children. Unlike a component which has only one parent, a question may have one or many parents. This reflects the fact that a question may require knowledge/skills from several content areas to answer. A question may have parents at any levels in the curriculum hierarchy. This allows general questions to be in a test. However, in practice, most questions' parents are components at the lowest level of the curriculum (e.g., concepts in Figure 1).

Questions in CBAT-2 are indexed by two parameters: a *difficulty level* and a *guessing factor.* The difficulty level describes how difficult a question is. It is comparable to the parameter b in IRT. The guessing factor describes the likelihood that a

question is correctly answered by guessing. It is comparable to the parameter c in IRT. The *discriminatory power*, parameter a in IRT, is not used in the current CBAT-2 because its values are usually difficult to calibrate and its meaning is difficult to be understood by test designers.

The guessing factor of a question is determined by the rate of the number of correct answers and the number of answers that the student may choose. For example, it is 0.5 for a True-Or-False question, 0.25 for a Multiple-Choices question with four mutual exclusive choices, and 0 for an Written-Answer question.

The value of the difficulty level ranges from 0 to 1. It is obtained by combining a designated initial value and historical information, using the following formula:

$$diff_i = \frac{20 \cdot init_i + \Phi_i}{20 + R_i + W_i}$$

where $init_i$ is the *initial difficulty* of question; the constant 20 is a normalization factor; R_i and W_i are the numbers of times that Q_i was correctly answered and incorrectly answered in the past, respectively. Φ_i is a difficulty accumulator for question Q_i such that every time a student incorrectly answers Q_i, the difficulty level equivalent to the student's temporary proficiency θ' is accumulated. Thus,

$$\Phi_i = \sum_{j=1}^{n} k_j \cdot f(\theta_j')$$

where n is the number of answers for Q_i in the past (i.e., $n = R_i + W_i$); θ_j' was the temporary proficiency of the student who gave the jth answer for Q_i; $k_j = 0$ if the jth answer is correct, and $k_j = 2$ if the jth answer is incorrect; $f(\theta_j')$ converts a θ value (-4 to 4) to a difficulty level (0 to 1). In the current implementation of CBAT-2, f is a linear function.

The initial difficulty has a value from 0 to 1. It is assigned by the test designer, based on his/her knowledge about the question. With the initial difficulty, it is no longer necessary to conduct an empirical study or to have historical information before adding questions to a pool. At the beginning, if there is no historical data, the initial difficulty determines the difficulty level. As the question is used in the tests, R_i and W_i become larger, and the difficulty level converges to $\Phi_i/(R_i + W_i)$. Thus, in some sense, CBAT-2 has a learning ability.

3.3 The Testing Algorithm

For each specific test, the test algorithm operates on the sub-curriculum of the test (see Figure 2) that is generated in the initialization of the test.

The algorithm consists of three procedures: a *question selector,* a *proficiency estimator*, and a *score and mastery decider.* Initially, there is an initial proficiency, θ_I, for a student, based on the test designer's knowledge about the student or a default value. The question selector selects a question based on θ_I. Once the student answers the

question, and the correctness of the answer is determined, the proficiency estimator calculates a new temporary proficiency, θ', for the student. It also calculates the confidence degree, v', for θ'. If v' has not reached a pre-designate confidence criterion v_0, then the question selector selects another question for the student. This goes on until v' passes the confidence level of v_0. Then the test stops. The temporary proficiency θ' becomes the proficiency θ. The score and mastery decider converts θ to a score that is comparable to the raw score in a conventional paper-and-pencil test, and determines whether the student is a master or a non-master.

This three-procedures approach is similar to the approach of Kingsbury and Weiss' (1979) Adaptive Mastery Testing (AMT), but each procedure is designed to meet the goals of CBAT-2. In the following subsections, we discuss these procedures in details.

Question Selection

The question selection procedure contains two steps. The first step is to decide which component (content area) the question comes from. This *working component* is randomly selected among a set of candidate components. A *candidate component* is a component under the sub-curriculum of the test such that the student's proficiency in this component has not been decided. However, the candidate components don't have an equal chance of being selected. The probability for a candidate component to be selected depends on its weight. The following formula is used to calculate the selection probability P_i for component C_i, where W_i is the weight of C_i.

$$P_i = \frac{W_i}{\sum W_j \mid C_j \text{ is a candidate component}}$$

The second step is to select a question among those associated with the chosen component. The question is selected based on the amount of information that a question may provide for the student's assessment. The information available, $I_i(\theta)$, from question Q_i, is calculated based on Birnbaum's (1968) logistic ICC (item characteristic curve) model. In this model, $I_i(\theta)$ is decided by the three IRT parameters (*a, b* and *c*) of the question and the student's temporary proficiency θ'.[1]

Questions in CBAT-2 does not have the parameter *a*. We use a constant, 1.2, for the parameter in the model because 1.2 is near the mean of the parameters in the question pools in Kingsbury and Weiss' (1979) study. For the parameter *b*, we use function $g(\text{diff}_i)$ to convert the value of the difficulty level, 0 to 1, of each question to a *b* value, -4 to 4. The function *g* is the reverse function of *f* (see section 3.2). It is currently a linear function. The parameter *c* directly uses the guessing factor of the question.

Once the $I_i(\theta)$ for each question is calculated, a set of questions with the highest $I_i(\theta)$ becomes the *candidate questions* among which a question is randomly selected for the

1. Due to the size restriction of the submission, we do not explore the details of the logistic ICC model. Interested readers may refer to the publication by Birnbaum (1968) and Kingsbury and Weiss (1979).

test. (The size of the candidate question set may vary with different applications.) This is different from AMT where the question that has the highest $I_i(\theta)$ is selected for the test. Our approach does not have the deterministic property that may cause a security problem.

Proficiency Estimation and Test Termination

Once the student has provided an answer for the selected question, and the system determined the correctness of the answer, the proficiency estimator is invoked to update the temporary proficiency, of the student. In CBAT-2, besides a temporary proficiency, θ', for the test (which is also the temporary proficiency of the parent component in the sub-curriculum of the test), there is a temporary proficiency, θ_i', for each child component C_i in the sub-curriculum of the test. Also, if there are questions directly associated with the parent component, CBAT-2 would create a component to represent the general knowledge of the parent component. This created component is treated in the same way as other child component of the parent component. It also has its temporary proficiency. For example, in the module test in Figure 2a, there are θ' for the module, θ_1' and θ_2' for Concept 1 and Concept 2, and θ_0' for the general knowledge of the module. In the course test in Figure 2b, there are θ' for the course, θ_1' and θ_2' for Module 1 and Module 2.

The proficiency estimator updates the temporary proficiency of the test and the components associated with the question that the student just answered. It also calculates the confidence level of each updated temporary proficiency. The proficiency estimator uses Owen's (1975) Bayesian updating procedure.[2] (This procedure is also used in AMT.)

There are two parameters that a test designer may set values to decide when a test terminates. One is the confidence level criterion for the proficiency estimation (say, 95%). We call it the *confidence criterion.* The other is the minimum number of questions that must be selected from each child component in the sub-curriculum (say, 2). We call this number the *MNQ.* Normally, the test terminates when two conditions are met: (1) the temporary proficiency of the parent component (or the test), θ', has passed the confidence criterion; (2) every child component has at least MNQ associated questions selected for the test.

Once a test terminates, the temporary proficiency of each component becomes its proficiency. However, the administrator must use with caution the assessment information for the child components because their proficiency may not have passed the confidence criterion. If the assessment of a child component, say C_i, is critical and must be precise, then a test on the sub-curriculum where C_i is the parent component may need to be conducted.

---

2. Again, due to the size restriction, we do not explore the Bayesian updating procedure here. Interested readers may refer to the publications by Owen (1975) and Kingsbury and Weiss (1979).

Scoring and Mastery Decision

The proficiency of each component in the sub-curriculum, in particular the proficiency of the parent component (or the test), may be converted to a percentage score that is commonly used in conventional paper-and-pencil tests. The conversion is based on the test characteristic curve (TCC) of a three-parameter logistic ogive used in AMT (Kingsbury and Weiss, 1979). We use the same conversion method as that in our question selection procedure (see section 3.3.1) to obtain the three parameters of each question.

If the test is a mastery test, then the score is used to compare with a master level set by the course administrator. A score above the master level classifies the student a master of the corresponding component or the test. A score below the master level classifies the student a non-master of the component.

4. Related Work

Much of the algorithm of CBAT-2 is based on IRT. Some techniques that we use come directly from AMT (Kingsbury and Weiss, 1979). However, like other IRT-based algorithms, AMT requires an expensive item calibration process to develop an item pool. This approach may work for large-scale institutional testing, but it doesn't seem to be appropriate for computer-based industrial training and local school education where the budget is normally small and the courses are diverse. CBAT-2 uses a simple machine learning approach to item calibration which requires no empirical study. It removes a main obstacle which prohibits small organizations from using adaptive testing.

AMT does not address the issue of content balancing, but Kingsbury and Zara (1989) recently developed a simple content-balanced testing algorithm. CBAT-2 is alike to their algorithm in that both deal with two-level content areas. However, rather than viewing a test as a stand alone activity, we place a test in the context of a course curriculum, which allows a test at the high-level of the curriculum to use questions in low-level content areas, and frees the test designer from having to develop a different pool for every test.

Wainer and Kiely (1987) has proposed the *testlet* approach for the content-balancing problem and the context effect problem. "A testlet is a group of items related to a single content area that is developed as a unit and contains a fixed number of predetermined paths that an examinee may follow. (P.190)" If an adaptive testing algorithm selects a testlet and a particular path of the testlet, then all items on the path will be presented in the test in the same order as they are in the testlet. The testlet approach relies on the test designer to ensure that the content areas are covered. This actually degrades adaptive testing to a mix of conventional fixed-item testing and adaptive testing. One may view a fixed-item test as a big linear testlet that contains all items in the test. Kingsbury and Zara (1989) have criticized the testlet approach, for it reduces the measurement accuracy and efficiency of adaptive testing.

To address the problem of requiring a demanding item calibration process in IRT-based algorithms, Welch and Frick (1993) have used an expert system approach to develop a series of *EXSPRT*'s. They showed a study of EXSPRT where a group of 38 student subjects were used for item calibration. Comparing to IRT-based algorithms that require 200 to 1,000 student subjects for item calibration, EXSPRT has made a significant advance. However, a study of 38 subjects may still be too difficult for most small

organizations, especially when the knowledge and experience of educational empirical study is required. Also, like most other algorithms, EXSPRT's do not address the content-balancing problem.

An EXSPRT is a mastery testing algorithm. A vital hidden problem in an EXSPRT is that the master level is decided in the empirical study that collects the historical data for its rules. Once the study is done, the test designers and administrators cannot change this master level freely. This is hardly acceptable for most organizations that live in a dynamic world.

5. Future Directions

Several things are in our future research agenda. We are planning to do an empirical study on real testing data provided by Alberta Education to compare the results of CBAT-2 and other adaptive testing algorithms as well as the conventional paper-and-pencil testing. We are doing research on student modelling (Collins et al., 1995). We intend to use the adaptive testing results as a source of input for our student modelling system that will provide assessment on the student's knowledge. We also intend to integrate CBAT-2 into a commercial learning environment, using it in real-world training.

Acknowledgment

Special thanks to Jason Collins, Jim Greer and Mike Dobson for their contribution in developing CBAT, an earlier version of CBAT-2, Gage Kingsbury for his valuable inputs to this research. Thanks to Janet McCracken and Chris Hughes for their comments on an early draft of the paper, and the Alberta Research Council for providing research funding.

References

1. Birnbaum, A. (1968): Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord and M. R. Novick, Statistical Theories of Mental Test Scores. Reading, MA: Addison-Wesley.
2. Collins, J. A., Greer, J. E. and Huang, S. X. (1996): Adaptive testing using granularity hierarchies and Bayesian nets. Proceedings of ITS '96.
3. Kingsbury, G. G. and Weiss, D. J. (1979): An Adaptive Testing Strategy for Mastery Decision. Research Report 79-5, Psychometric Method Program, Department of Psychology, University of Minnesota.
4. Kingsbury, G. G. and Zara, A. R. (1989): Procedures for selecting items for computerized adaptive tests. Applied Measurement in Education 2, 359-375.
5. Owen, R. J. (1975): A Bayesian sequential procedure for quantal response in the context of adaptive mental testing. Journal of the American Statistical Association 70, 351-356.
6. Wainer, H. (1990): Computerized Adaptive Testing: A Primer. Lawrence Erlbaum Associates, Inc., Publishers.
7. Wainer, H. and Kiely, G. L. (1987): Item clusters and computerized adaptive testing: A case for testlets. Journal of Educational Measurement 24, 189-205.
8. Welch, R. E. and Frick, T. W. (1993): Computerized adaptive testing in instructional settings. ETR&D 41, 47-62.