First International Conference
Modelling and Development of Intelligent Systems
Sibiu - Romania, 22-25 October, 2009

Adaptive Test Generation for E-Learning Systems

Ana-Maria Mirea[1], Constantin Teodorescu-Mihai, Doina Preda, Mircea Preda

Abstract

Computer based testing can provided knowledge evaluation that is tailored to the specific of each student with costs and quality performances that exceeds the human based testing. Usually, testing performed by the humans is a tradeoff between the involved costs and accuracy. Usually, an accurate test will take longer and if it is performed by a human examiner it will cost more. Also, longer tests can made the human examiners more prone to be affected by other factors like fatigue that can have an adverse impact on quality. Computer based testing can permanently adapt to the specific of the students, it is not affected by human specific factors and has low costs on long run.

1 Introduction

Testing is a major component of the learning process, it allows to evaluate the progress of learning. Classical testing involves human examiners and it is costly, subjective, error prone and not ready available in every situation. E-testing systems are permanently available for the students, are not biased by human factors like fatigue, provides fast and accurate results, the feedback for the students is comprehensive and a system can cover a large amount of topics being able to replace several human examiners. Also, e-testing systems provides auto evaluation possibility to the students.

Several major attempts to build Computer Adaptive Testing (CAT) systems can be mentioned. Their mathematical background is provided by the Item Response Theory (IRT) [1] that is concerned with the application of mathematical models to data from questionnaires and tests as a basis for measuring abilities, attitudes, or other variables. It is used for statistical analysis and development of assessments, often for high importance tests. The main assumption behind IRT is that the probability to obtain a correct answer to an exercise is a mathematical function of the characteristics of the person who takes the exercise and the features of the exercise itself.

In [4] it is described a web-based tool to assist teachers and instructors in the assessment process. The tests are generated according to teachers specifications and are adaptive, that is, the questions are selected intelligently to fit the students level of knowledge. In this way, are obtained more accurate estimations of students knowledge with significantly shorter tests. The knowledge level of the student is represented and measured by a single variable θ. Using as input data a set of responses of the students to a set of questions, the level of knowledge of the student is estimated using statistical methods. Then, the estimation $\hat{\theta}$ is used to determine the most informative item to ask next. These steps are repeated until some stopping criterion is met. Different statistical methods to estimate θ and to select the next best question to ask give different IRT models that are used in the article. By using applets to present exercises, rich interactions between students and testing system are possible. Some adaptive test generation methods involve costly time and human resources for preparation and have complex mathematical formulas. These methods are available only in large educational institutes and in professional testing centers. In [6], it

[1] All authors contributed equally to the paper.

is proposed a CAT method tailored for classrooms and for small business daily routine. The evaluation problem is considered as a sequential statistical hypothesis test where the hypothesis is *the topis is mastered*. As a solution is employed the Sequential Probability Ratio Test (SPRT), originally used in quality control but later reformulated for use in computerized testing of human examinees. The authors obtain a system that is easy to implement and does not require a large number of training sessions for parameter estimation. [7] provides an excellent coverage of computerized adaptive testing (CAT). It presents all necessary statistic and psihometric concepts and provides also practical foundations. [3] and [2] discusses the general problematic of adaptive educational systems. [5] presents the main learning styles that can be encountered at the students and the implications of these styles in building adaptive learning systems.

This paper formalizes an e-testing system with advanced capabilities that include adaptivity to each particular student and a database of exercises that is permanently evolving. The classic structure of the CAT methods is followed: an iterative algorithm where at each round, new exercises are presented to the examinee. If he performs well, more difficult questions will be presented, otherwise, if the performances are poor, simpler questions will be used. The proposed formalism is centered around the notion of difficulty degree and difficulty degree updating. Both an individual exercises and tests have attached difficulty degrees. The system maintains for each user the most appropriate difficulty degree. Initially, the difficulty degree for a student can be obtained by data mining methods starting from the difficulty degrees of similar users on a predefined initial value can be used. After each interaction between the testing system and user, the difficulty degree is updated. For a specified difficulty degree, the system will generate every time a different test. Other advanced features of the system include a permanently evolving exercises database, possibility to include short resumes and full lessons on covered topics and rich possibilities of interaction with the students. Exercises database evolves by generating new exercises using predefined patterns and by incorporating exercises proposed by students.

The main differences over the already developed CAT systems are:

- Most CAT methods estimate the knowledge level of the students but not the features of the exercises. The newly introduced method estimates the characteristics (difficulty) of the exercises and these estimations are permanently updated. The degree of mastery attained by a student on a topic is given by the difficulty of the exercises that were successfully resolved.

- Usually, the CAT systems are not able to provide detailed assessments on subtopics, they provide only a single estimation of the knowledge level. In the presented settings, the topic of the test can be divided in several subtopics and the degree of mastery on each subtopic is estimated.

- Measures of similarity are used to identify the exercises and students that are closely related. In this way, the system can transfer the learned knowledge between similar users and similar exercises and can substitute an exercise with another one.

The next section of the paper presents the theoretical model of the proposed adaptive testing system. After that, there presented implementation considerations, algorithms and further extensions of the system capabilities.

2 Specifying adaptive tests

In the following paragraphs, S will represents the set of the students that use the e-learning system and $s \in S$ is a particular student.

Let us consider Ex the set of exercises that are available for the testing and evaluation system. A test is a subset of Ex, $T \subseteq Ex$ that satisfies a specified set of restrictions. For example, the number of exercises from the test can be limited by a superior threshold, $Card(T) \leq M$, where $Card(.)$ is the cardinal function that provides the number of elements from a set and $M \in \mathbb{N}$.

A function $Diff : Ex \rightarrow \mathbb{R}$ will be used to evaluate the difficulty degree of the exercises from tests, where for an exercise $e \in Ex$, $Diff(e)$ represents the difficulty of the exercise e. Function $Diff$ can be extended to sets of exercises:

$$\begin{aligned}&Diff : 2^{Ex} \to \mathbb{R}\\ &Diff(T) = \sum_{e \in T} Diff(e)\end{aligned}$$

In accordance with the above definition, the difficulty of a test is the sum of the difficulties of the exercises that appear in the test. Alternately, the difficulty of a test can be the average difficulty of the exercises that form the test. The definition is as follows:

$$\begin{aligned}&Diff : 2^{Ex} \to \mathbb{R}\\ &Diff(T) = \frac{\sum\limits_{e \in T} Diff(e)}{Card(T)}\end{aligned}$$

The difficulty of a test will be limited by a constant $d_{\max}$, $\forall T, Diff(T) \leq d_{\max}$. $d_{\max}$ represents the maximum difficulty for a test.

2.1 Estimating difficulty of the exercises

A first estimation for the difficulty of the exercises will be provided by the human operator,

$$e \in Ex \underset{human}{\longrightarrow} Diff(e)$$

To simplify the task of the human operator, we can suppose that the first estimation of the difficulty degrees will use only a discrete set of qualitative values like *easy, average, high* and these values will be after that converted to numbers.

The first estimation will be subsequently refined using the performances of the students during tests. Let us consider $e \in Ex$ an exercise and $Result(e)$ a measure of the performance of a student for e where

$$Result(e) = \begin{cases} -1 & \text{if student failed to resolve the exercise} \\ 1 & \text{if student succeded to resolve the exercise} \end{cases}$$

Then, a new estimate for the difficulty of e can be obtained as follows:

$$Diff(e)^{new} \longleftarrow Diff(e)^{old} + \alpha(e) \cdot Result(e) \cdot Diff(e)^{old} \tag{1}$$

where $\alpha(e) \in [0, 1]$ is a parameter named *learning rate* for the exercise e. Usually, $\alpha(e) \to 0$ when $t(e) \to \infty$ where $t(e)$ represents the number of appearances in tests for the exercise e (over time the updates of the difficulty degree for the exercise e should be smaller because the already existent estimation will be accurate). A common formula for the learning rate is

$$\alpha(e) = \frac{\alpha_0(e)}{1 + \frac{t(e)}{TE}}$$

where $\alpha_0(e)$ and TE are constants.

2.2 Similarity measure between exercises

For adaptive tests generation it is important to establish how similar are two exercises. For example, if a student fails to resolve an exercise, similar exercises can be proposed to him in the subsequent tests to accurately measure the level of knowledge of the student in the area. Moreover, with a similarity measure between exercises, different tests can be generated for different students, tests that cover same topics and have similar levels of difficulty. In this way, the evaluation process will be consistent and in same time the students will have no possibility to know what exercises will appear in a test.

A measure of the distance between two exercises is introduced as:

$$d : Ex \times Ex \to \mathbb{R}_+$$

with the following properties:

$$\begin{aligned}&d(e_1, e_2) \geq 0\\ &d(e, e) = 0\\ &d(e_1, e_2) \leq d(e_1, e_3) + d(e_3, e_2)\end{aligned}$$

2.3 Classes of exercises

A category or a class of exercises is a subset of the full set of exercises $C \subseteq Ex$ that includes exercises with similar properties. Let us denote by $\mathcal{C} = C_1, ..., C_k$ the set of the all categories that are available for the system. It is possible that same exercise e to apart to two or more categories. This means that two categories can have a non empty intersection $C_i \cap C_j \neq \emptyset$. We will suppose that an exercise $e \in Ex$ will apart to at least one category, and, consequently,

$$\bigcup_{i=1}^{k} C_i = Ex.$$

Let us suppose that a distance measure can be devised between categories:

$$dc : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}_+,$$

where $dc(C_i, C_j)$ represents how different are the categories C_i and C_j. In these conditions, a distance between exercises can be devised based on the distance between categories:

$$d(e_1, e_2) = \max_{\substack{c_1, c_2 \\ e_1 \in C_1 \\ e_2 \in C_2}} dc(C_1, C_2) \tag{2}$$

2.4 Generating adaptive tests

Let us consider $T \subseteq Ex$ a test that was performed by a student s and $Success(T, s) \subseteq T$ and $Failure(T, s) \subseteq T$ represent the exercises that were successfully resolved and, respectively, failed. The level of knowledge of the student s assessed by the test will be defined as:

$$TrainingLevel(T, s) = \frac{Diff(Success(T, s))}{Diff(T)} \in [0, 1]. \tag{3}$$

Otherwise stated, the training level of a student s regarding to a test T is the ration between the complexity of the exercises that were correctly resolved and the total complexity of the test.

The difficulty degree of the next test can be established in accordance with the algorithm 1.

Algorithm 1: The algorithm used to adjust the difficulty degree of the tests in accordance with the student's performances.

```
SELECTING THE DIFFICULTY DEGREE OF THE NEXT TEST()
    if $TrainingLevel(T, s) > \beta_1$
        increase the difficulty level of the next test
    else if $TrainingLevel(T, s) < \beta_2$
        decrease the difficulty level of the next test
    else
        maintain the current difficulty level
```

Here, β_1 and β_2 are two constant thresholds that are used to trigger the update of the test's difficulty level for a student.

Difficulty level adjustment can be also done on each category. Let us define

$$TrainingLevel(T, s, C) = \frac{Diff(Success(T, s) \cap C)}{Diff(T \cap C)} \in [0, 1]. \tag{4}$$

to be the knowledge level of the student $s \in S$ for the category $C \in \mathcal{C}$ assest by the test $T \subseteq Ex$. The difficulty of the exercises selected from the category C can be updated using the mentioned algorithm.

2.5 Updating the difficulty degree

The algorithm from previous section involves update operations on the difficulty degree of the entire test or of a specified category. These update operations will be performed as is prescribed by the following guidelines. Let us consider $d_{\min}$ the minimum degree of difficulty and $d_{\max}$ the maximum degree of difficulty, and, d_{crt} the current one, $d_{\min} \leq d_{crt} \leq d_{\max}$. Three situations are possible:

- Difficulty degree should be increased. Then, the next difficulty will be selected using a Gaussian distribution with mean $(1-\gamma)d_{crt} + \gamma \cdot d_{\max}$ and variance 1. Probability density of this function is represented in the figure 1.

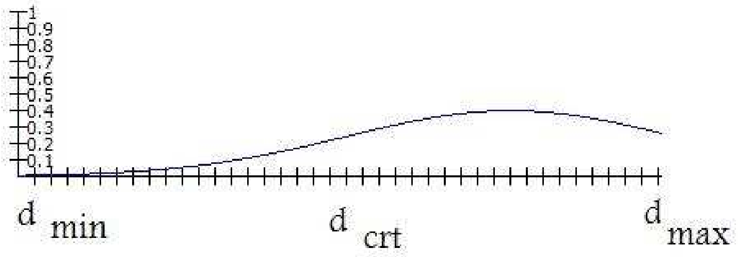

Figure 1: Density function for the Gaussian probability distribution used to select the newly increased difficulty degree with $\gamma = 0.5$.

- Difficulty degree should be decreased. Then, the next difficulty will be selected using a Gaussian distribution with mean $(1-\gamma)d_{crt} + \gamma \cdot d_{\min}$ and variance 1. The process is presented in the figure 2

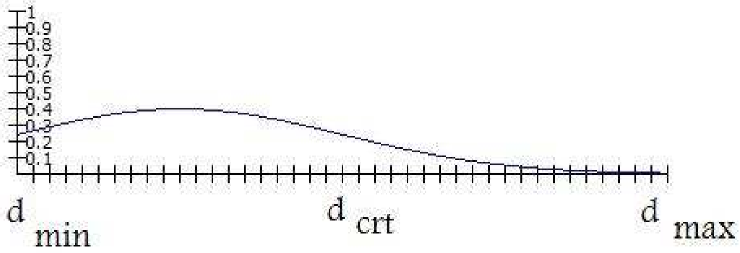

Figure 2: Density function for the Gaussian probability distribution used to select the newly decreased difficulty degree with $\gamma = 0.5$.

- Difficulty degree should be maintained. Then, the next difficulty will be selected using a Gaussian distribution with mean d_{crt} and variance 1.

Here, $\gamma \in [0, 1]$ is a parameter named *update rate.* γ can be a constant, for example, $\gamma = 0.5$, or, γ can have decreasing values, $\gamma \to 0$ when the number of difficulty updates goes to ∞.

2.6 Generating a test with a specified difficulty

Let us consider $\mathcal{C}(T) \subseteq \mathcal{C}$ the categories of exercises that will be considered for the test. If $Card(\mathcal{C}(T)) = 1$ then the test will involve only one topic (category). If $\mathcal{C}(T) = \mathcal{C}$ then the test will involve all topics that are available for the system. Let us denote by

$$P(C) = \frac{Card(C)}{\sum\limits_{C_i \in \mathcal{C}(T)} Card(C_i)}, C \in \mathcal{C}(T). \tag{5}$$

$P(C) \in [0,1]$, $\forall C \in \mathcal{C}(T)$ represents a discrete probability distribution over the set $\mathcal{C}(T)$ of the categories (topics) involved in the test.

If the d_{crt} is the difficulty degree of the test that should be constructed, then the algorithm 2 is used to achieve the desired result.

Algorithm 2: The algorithm used to create a test with a specific difficulty degree. The iterative procedure selects at each step a category and, then, an exercise from that category.

```
Constructing a test with a specified difficulty degree($d_{crt}$)
(1)  $T \leftarrow \emptyset$ (the initial test is empty)
(2)  while true
(3)      Select the category $C \in \mathcal{C}(T)$ of the exercise that will be added to the test
         in accordance with the probability distribution $P(C)$ (the categories with a
         larger number of exercises will have a greater probability to be selected).
(4)      Select $e \in C$ an exercise in accordance with the target difficulty $d_{crt}$ and with
         the current difficulty of the test $Diff(T)$.
(5)      $T \leftarrow T \cup \{e\}$ (the exercise is added to the test).
(6)      if $StopConditions(T, d_{crt})$ are satisfied
(7)          return $T$
```

When an exercise is to be selected from a category in order to be added at a test, there are two situations:

- The difficulty of the test is considered the sum of the difficulties of the exercises from test $Diff(T) = \sum\limits_{e \in T} Diff(e)$. A current difficulty level $d_{crt}(C)$ will be maintained for each category $C \in \mathcal{C}(T)$. The exercise will be selected using a Gaussian distribution with the mean $d_{crt}(C)$ and variance 1. In this way, all exercises from the category have a chance to be selected and the exercises with the difficulty around $d_{crt}(C)$ are preferred. Initially, the intra category difficulty $d_{crt}(C)$ should be low or medium low, because, pedagogy indicates that is recommendable to start with relatively low complexity exercises. $d_{crt}(C)$ will be updated after each test performed by the student in accordance with his results.
- The difficulty of the test is the average of the difficulties of the exercises from test $Diff(T) = \frac{\sum\limits_{e \in T} Diff(e)}{Card(T)}$. In this case, the exercise will be selected using a Gaussian distribution with the mean d_{crt} and the variance 1.

To define $StopConditions(T, d_{crt})$ same situations as above should be considered:

- The difficulty of the test is the sum of difficulties of the exercises that form the test. Then

  $$StopConditions(T, d_{crt}) = \begin{cases} true & \text{if } Diff(T) \geq d_{crt} \\ false & \text{if } Diff(T) < d_{crt} \end{cases}$$

  Otherwise stated, test building process stops when the difficulty of the current test exceeds the target difficulty d_{crt}. The employed algorithm is similar with the algorithm for the knapsack problem, exercises are added until the target difficulty, which is the capacity of the knapsack, is exceeded.
- Test difficulty is the average of the exercises difficulties. In these conditions, test building process stops when some test admissibility conditions are satisfied. Examples of such admissibility conditions are:

- Test should have at least Min exercises, $Card(T) \geq Min$.
- A test is admissible if it includes an exercise for each category, $\forall C \in \mathcal{C}(T)$, $\exists e \in T$ such that $e \in C$.

2.7 Adaptive test generation

Adaptive test procedure for a student is described in the algorithm 3 and it synthesizes the theoretical developments from the previous sections.

Algorithm 3: The adaptive testing procedure. At each step a test is generated in accordance with the recommended difficulty level for the student. The test is applied and after that the difficulty level is updated in accordance with the test results.

```
Adaptive test procedure for a student($s \in S$)
(1)    $d_{crt} \leftarrow d_{init}(s)$ the initial difficulty level for the student $s$
(2)    while stop conditions not satisfied for $s$
(3)        generate a test with the difficulty $d_{crt}$
(4)        apply the test to student and record the results
(5)        update the difficulty level $d_{crt}$ in accordance with the results.
```

Regarding the stop conditions for a student s several scenarios can be considered:

- The student s uses the system on regular basis to improve or maintain his knowledge level. In this case, the testing loop can continue forever. The student should have the possibility to reset the adaptive system in order to start the training again from the initial conditions.
- The testing system is used by an institution to evaluate the knowledge level of the student s. In this case, the system should stop when the variance on the last K tests is under a threshold $\delta(K)$. A low variance means that the difficulty level, which is optimal for the student s, was found and the system is stabilizing around this level. In this case, the average difficulty level on the last K tests can be used as a measure of the knowledge for s. A limit of the number of tests that can be applied to s, *MaxTestsNum* can be specified in this situation to cope with the cases when the difficulty level has strong oscilations for s.

3 Implementing testing systems with advanced capabilities

In this section, it will be discussed implementation details for the previously presented theoretical model and other features that can improve the e-learning system.

3.1 Personalizing the system for each student

Our system should consider each student as an individual entity with specific requirements. In order to do that, the system should store data to be able to differentiate between students. Maintained data will include:

- Personal information about student. Ideally, this information should allow to identify the similar students.
- Information about the performed tests. Minimally, this information will consist from the success percentages obtained at the tests. If the storage possibilities allow, the information can also include even the exercises that composed the tests, at the results for each exercise.
- Information on the evolution of the difficulty degree d_{crt} of the administrated tests. The intra category difficulty degree evolution $d_{crt}(C)$ can be also included for each category $C \in \mathcal{C}$.

3.2 Providing extensive feedback to the students

Students should receive complete statistics and graphical representations on their tests' performances. The evolution of the tests' difficultly degree should be also included. For each failure, the student will receive full explications on the topic.

3.3 Passing from a testing system to a learning system

The system can be extended with short summaries and even with full lessons on the covered topics. Different techniques can be used to ensure that the students really read and understand the presented information. A such technique can be for example to impose a minimum time of presentation of the information on screen. Also, each test can be prefaced by the summaries of the covered topics. Also, access to some tests can be allowed only to the users that completed a specified set of lessons and/or summaries.

3.4 A permanently evolving system driven by students

For some types of systems and students, it can be considered the possibility that the exercise database to be extended with the exercises proposed by examinees. The similarity measure between exercises can be used to automatically classify into categories the newly added exercises. Of course, the quality of these exercises is a concern and should be carefully assessed. Several methods to ensure the quality can be considered:

- All new exercises will be moderated by human administrators.
- Only the students with high performances can propose new exercises.
- Feedback from other users will be employed to measure the quality of the exercises.
- Social trust measuring methods can be employed, and only the trusted users can extend the exercise database.

3.5 Measuring the quality of the proposed tests

The system can request to users to provide feedback information about the quality of the exercises from tests and about the overall quality of the tests. The requested feedback should be carefully minimized in order to not place an unpleasant burden on the students. The exercises that in average have on long run low quality marks will be removed from the system.

3.6 Adapting to the learning styles of the students

The proposed model does not impose any restriction on the modality of presentation for the exercises. However, it can be supposed that the presentation with several alternate answers will predominate. Our system can accommodate variants of presentations that are predominant visual, auditive, practical or combinations between them. These styles can be easily applied to lessons and resumes but also can appear in the testing procedure. The system should establish the main learning style of a student using an initial set of questions or based on the student performances on different types of presentations. Ideally, the system should use for each student a variant of presentation that is suited to the student learning style.

4 Conclusion

The paper proposes a testing system that adapts the difficulty of the tests to the training level of the students. The system is highly extensive, it can adapt several learning styles, can be enriched with short summaries and full lessons and has an exercise database that is permanently evolving. The system always will present different tests to students even the difficulty level has not changed. The system is configurable, both administrators and students can decide how testing is performed and in particular when the testing procedure is finished. Future improvements will include:

- Automatic generation of exercises on specified topics.
- Automatic generation of summaries from full lessons.
- Rich interaction possibilities for the students (hyper-media).
- Improved data mining procedures to identify commonalities between students and exercises.
- Day time usage statistics to identify patterns in the results obtained by students.

References

[1] Frank B. Baker, *The Basics of Item Response Theory*, ERIC Clearinghouse on Assessment and Evaluation, 2001.

[2] Peter Brusilovsky. A Distributed Architecture for Adaptive and Intelligent Learning Managament Systems. *In Proceedings of the AIED 2003 Workshop Towards Intelligent Learning Management Systems*, Sydney, 5–13, 2003.

[3] Peter Brusilovsky. Adaptive navigation support: from adaptive hypermedia to the adaptive Web and beyond. *PsychNology Journal*, Vol. 2, No. 1, 7-23, 2004.

[4] Conejo, R., Guzman, E., Millan, E., Trella, M., Perez-De-La-Cruz, J. L., and Rios, A. SIETTE: A Web-Based Tool for Adaptive Testing. *Int. J. Artif. Intell.*, Ed. 14, 1, 29–61, 2004.

[5] Elvira Popescu, Philippe Trigano, Costin Badica. Towards a Unified Learning Style Model in Adaptive Educational Systems. *Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007)*, 804–808, 2007.

[6] Tao, Y.-H., Wu, Y.-L., Chang, H.-Y. A Practical Computer Adaptive Testing Model for Small-Scale Scenarios. *Educational Technology and Society*, 11(3), 259-274, 2008.

[7] Howard Wainer, Neil J. Dorans, Ronald Flaugher, Bert F. Green, Robert J. Mislevy, Lynne Steinberg, David Thissen *Computerized Adaptive Testing: A Primer*, Lawrence Erlbaum, 2000.

Ana-Maria Mirea
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, Craiova, 200585
Romania
E-mail: *ammirea@acm.org*

Constantin Teodorescu-Mihai
"Petrache Poenaru" High School
Department of Computer Science
Bălceşti, Vâlcea, 245400
Romania
E-mail: *constantintm@ymail.com*

Doina Preda
"Matei Basarab" High School
Department of Computer Science
Vasile Alecsandri street, 113, Craiova, 200463
Romania
E-mail: *dpreda@acm.org*

Mircea Preda
University of Craiova
Department of Computer Science
Al. I. Cuza street, 13, Craiova, 200585
Romania
E-mail: *mpreda@acm.org*