Department of Electrical, Computer, and Software Engineering

Part IV Research Project

Final Report

Project Number: 52

Computer

Adaptive Testing for

Educational Assessment

Oscar Li

Hajin Kim

Yu-Cheng Tu (Supervisor)

Andrew Meads (Co-supervisor)

13/10/2022

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Signature:

Name: Oscar Li

ABSTRACT: Assessment plays a vital role in education and it is therefore important to have tests that accurately measure a student's ability. Multistage testing is a form of computer adaptive testing that adjusts its difficult according to the user's ability with the goal of achieving greater measurement precision while also requiring fewer test items. Investigation into the relevant literature has uncovered a lack of research on multistage test implementation and its application in a high school curriculum. As a result, this project aims to compare the effectiveness of multistage testing against traditional fixed length tests by implementing an online testing platform targeted towards high school mathematics students in New Zealand. The application leverages popular web development frameworks and integrates them with the open-source multistage testing library, mstR. User testing has been conducted to evaluate the effectiveness of both tests as well as the usability of the application. Results from testing show that there is a positive correlation between multistage and fixed length test performance and most participants found that the multistage test was better suited to their ability. However, there were some limitations associated with testing such as the difference between the participant demographic and the intended demographic. The application has potential to be improved significantly by experimenting with the different models provided by mstR and adding new features to improve the testing experience. As a result, the final implementation serves as a solid starting point for further research in multistage testing.

Table of Contents

1. Introduction

Testing plays a crucial role in the educational process. Tests are used to attain an understanding of the student's current knowledge to inform the teaching and learning process as well as measuring performance after the student has completed a programme [1]. As a result, good test design is extremely important to get an accurate assessment of student ability. Test creators are given the difficult task of making a test that is comprehensive enough to give an accurate measure of ability, both in terms of content coverage and difficulty, while also trying to minimize the number of questions required. This has led to the emergence of Computer Adaptive Testing (CAT) which aims to achieve greater precision with fewer test items by utilizing algorithms derived from Item Response Theory (IRT). Such algorithms can select the most appropriate test item for a participant by predicting their ability based on their previous responses. Multistage Testing (MST) is a variant of CAT that administers and selects test items in stages or testlets.

A key goal of this project is to compare the effectiveness of MST against fixed length tests both in terms of estimation accuracy and enjoyability. Instead of focusing on institutionalized large-scale testing, this project also aims to explore the feasibility of using CAT to implement a testing platform at a small scale without requiring proprietary tools such as Xcalibre and Iteman. Implementing such an online testing platform involves creating a testing interface that users can interact with and integrating the application with opensource CAT frameworks such as mstR. Lastly, the application should be tested with real users to identify the effectiveness of MST and assess the overall usability of the application.

This report documents the research, design, and development of the testing platform. To provide the basis for project, a summary of the reviewed literature is included in the following section which focuses on item response theory, multistage testing, and their applications in educational assessment. Subsequently, the project requirements (Section 3), design (Section 4), implementation (Section 5), testing methodology (Section 6.1) and results (Sections 6.2 and 6.3) are discussed with key limitations identified (Section 6.3). Finally, the project goals and development are reflected upon in Section 7 before concluding with suggestions for future work in Section 8.

2. Literature Review

2.1. Item Response Theory (IRT)

Computer adaptive testing differs from traditional fixed length tests in that it adapts to the user's ability. The goal of this approach is to maximise assessment precision while requiring the same amount or fewer test items. Generally, CAT can be seen as being composed of six primary components: item response model, item pool, entry level, item selection, scoring method, and termination criterion [2].

2.1.1. Item Response Model

An item response model is used to identify the probability that a given test taker will answer a test item correctly. As a result, calculation using a model requires certain parameters such as item difficulty, item discrimination (how effective the item is at determining the ability of a test taker) and the chance of guessing a correct result. Most item response models are based on item response theory (IRT) and can be classified as one (1PL), two (2PL) or three (3PL) parameter logistic models [2]. In 1PL the only parameter considered is item difficulty, 2PL expands upon this by adding a discriminating factor and 3PL also adds another parameter to account for guessing. The models described are all dichotomous which are targeted at test items that only have two possible scores: correct or incorrect. There also exist polytomous models such as Rating Scale Model [3] and Partial Credit Model [4] which are used for items with more than two possible scores such as Likert-type items.

2.1.2. Item Pool

Based on the previous information about the item response model, it logically follows that a computer adaptive test must utilise a pool of questions that have been calibrated with the correct IRT parameters. Based on the literature, a pool of 100 items is usually sufficient to give acceptable results [2]. Additionally, it is highly important to have questions of varying difficulty and that the questions can discriminate between test takers of different ability.

2.1.3. Entry Level

Adaptive testing allows test takers to begin the test at different difficulty levels without having a significant effect on the measurement outcome [2]. This is because an adaptive test will gradually adjust to the test taker's ability level as they answer questions. However, to provide the most precise measurement within a certain number of items, test takers should begin at a level matching their ability [5].

2.1.4. Item Selection

Item selection plays a key role in CAT as the goal at each stage of the test is to select the item that will reveal the most information about the test taker's ability [2]. The two most used and researched item selection procedures are the Maximum Information [6] and Bayesian [7] procedures. Maximum Information seeks to find an item that will give the most information at the test taker's current estimated ability whereas Bayesian seeks to minimise the variance of the current estimated ability. Most CATs are conducted with a single procedure, however, there has also been research into using multiple procedures in the same test with promising results [8].

2.1.5. Scoring Method

A computer adaptive test must provide a new estimate of the test taker's ability after each item has been answered. Generally, this means a correct answer will result in an increased estimated ability and vice versa. However, the exact method used varies and can be done using either a maximum likelihood estimation or a Bayesian estimation method [2].

2.1.6. Termination Criterion

A stopping criterion is required to inform a CAT system when no further items are required to be administered to the test taker. Generally, the stopping criterion is specified by the test maker as an acceptable measurement error whereby the test concludes once the measurement error of the estimated ability falls within the specified range [2].

2.2. Multistage Adaptive Testing (MST)

Multistage adaptive testing differs from the traditional CAT, also known as the question adaptive model (QAM), in the way test items are administrated. While QAM selects a new test item after each item has been answered, MST instead does this in sets of questions known as testlets or stages [9]. Like QAM, the next set of questions administered to the participant in MST depends on their performance on previous item sets. Research has been conducted comparing the effectiveness of both approaches and results suggest that MST has greater measurement precision compared to QAM when using one-parameter and two-parameter logistic models and they performed equally for three parameter logistic models [10]. Another benefit of this approach is that test creators have more control over content balancing. The issue with content balancing occurs in QAM because examiners generally want tests to contain sufficient content coverage from multiple categories. In QAM, there is no guarantee that items will have balanced exposures without some constraints

being imposed which may limit the measurement efficiency [11]. This issue is mitigated by the coarse-grained nature of MST because examiners can decide the individual items within a set and hence have more control over content coverage. It also becomes easier to enforce limits on question set reuse and item overlap which improves test security [12]. Such characteristics make MST more favourable in practice as tests are expected to be secure while covering a wide range of topics. Consequently, the popularity of MST is reflected in the case studies discussed below

2.3. Multistage Testing in Educational Assessment

NAPLAN is a test administered by the Australian Curriculum, Assessment and Reporting Authority (ACARA) whose goal is to provide a snapshot of literacy and numeracy skills in Australian students [14]. The main interest in this case study lies in their transition from paper-based testing to computer adaptive testing. A key decision that had to be made was the choice between an item-level computer adaptive test or a multistage test. The key strength of traditional CAT is that it provides the highest possible measurement precision while using fewer test items. MST, on the other hand, allowed for more control over content coverage and requiring fewer test items to implement (as described in previous sections). As a result, the ACARA ultimately decided to use MST. A tailored test consisting of three stages and two branching points was proposed [15]. Pilot testing of the tailored test found that the standard error of measurement for ability estimates in students was significantly lower compared to fixed linear testing. This method also improved testing motivation of struggling students as poor performance at the first stage resulted in easier questions in later stages.

Another example is the Graduate Record Examinations (GRE) which transitioned to from CAT to MST in 2014 [13]. One motivation for this change was to improve the test taking experience. The original CAT did not allow participants to revisit previously answered questions whereas MST allows participants to freely move between questions within the same section. Another major benefit was the reduction in complexity and development cost when changing to MST as it required a smaller overall item bank to achieve the same content coverage. The final implementation of the revised GRE consisted of a 2-stage test with the first stage being known as a "routing test" whose results were used to decide the difficulty of the second stage test. The scoring model used was a 2PL IRT model which was based only on question difficulty and discrimination. The revised GRE introduced a wider variety of question types over pure multichoice; hence the effect of guessing was minimized and a 3PL IRT model was not required

2.4. Research Intent

Most of the research so far has been conducted by large institutions aimed at creating tests for thousands of participants and there is a relative lack of research discussing the feasibility of MST at smaller scales. Additionally, a significant amount of research on MST has been conducted on either primary school or university students. Based on these points, we aim to explore the feasibility of implementing an MST system targeting high school students using tools and software available to the wider population such as mstR. Our plan is to achieve this is by implementing a prototype using the IRT models described in the literature and measuring its effectiveness against fixed length tests.

3. Requirements

Before beginning development of the application, key requirements are established to guide the development process and ensure that the application successfully meets the goals of the project. The requirements can be separated into two categories; one category focuses on the test creation and the other category focuses on the features of the testing platform itself.

The literature review identified a lack of examples of CAT targeted at high school students. As a result, a requirement is to target high school students therefore the test content must contain questions that match the curriculum taught in New Zealand high schools. The chosen test domain is mathematics because all New Zealand high school students are required to study mathematics and the fact that the questions generally only have a single correct solution is ideal for the automated grading used in a computerized assessment. The next requirement is to have both a fixed length and multistage test covering the same content so that comparisons can be made and their differences analysed. Lastly, the multistage test needs to utilise CAT algorithms to differentiate between low, medium and high ability between each stage.

A major aspect of the project is the application that participants will use to take the test and the requirements are based on features of existing testing platforms such as Canvas and Inspera. A key requirement of this application is accessibility and users should be able to take the test on their own devices without needing to download additional software. Users should be able log in to the online platform with a unique ID and be administered the appropriate test. For this project, the application must also support both fixed length and multistage tests containing multiple choice and written answer questions. The application should have an intuitive testing interface with clear instructions that can easily be navigated by both adolescents and adults. Testing data should also be recorded and stored in a database so that the results are available later for analysis.

4. Design

4.1. Test Design

With the domain and target audience defined in the requirements, the next challenges involve the creation of a question/item bank to be used for testing and the actual format of the test itself.

4.1.1. Item Bank Creation

There are two approaches that were considered when creating an item bank. One approach involved creating novel questions specifically for the test whereas the other choice was to use past questions from similar tests. Both approaches had their own benefits and drawbacks. Creating an entirely new set of questions would be very time-consuming but would allow more control over the test content and difficulty. On the other hand, reusing questions would require much less time and the quality of the questions would likely be higher as they were created by a professional organization. Ultimately, the decision made was to reuse questions from a similar test due to the limited time frame of the project. The item bank for the project uses questions from the Trends in International Mathematics and Science Study (TIMSS) 2011 grade 8 mathematics assessment which contains a mixture of multiple choice and short answer questions targeted towards Year 9 students in New Zealand. A key reason for choosing this specific test is because CAT and MST require assigning parameters (such as item difficulty and/or discrimination) to each question and the data from the TIMSS assessment also detail the proportion of students that answered each question correctly. Using this data, an item difficulty parameter can be obtained by subtracting the percentage of New Zealand participants that answered correctly from 1 e.g. if a question was answered correctly by 30% of students, then the difficulty would be 0.7. This forms the basis of the one parameter (1PL) multistage test. A lack of other question data means that a 2PL, 3PL, or 4PL test is not supported as a value for discrimination and guessing cannot be derived.

4.1.2. Test Format

After creating the item bank, the next step is to create the format for the test. To compare the effectiveness of multistage testing with fixed length testing, a test had to be created using each paradigm.

For the fixed length test, the format is a 45-minute test containing 30 questions with a range of difficulties. This decision is based on the format of the original TIMSS mathematics assessment which is comprised of two booklets of 12-18 questions each to be completed in 45 minutes [16]. Both tests also use the same topic distributions as the TIMSS assessment which contains 30% number questions, 30% algebra, 20% geometry, and 20% data and chance.

The multistage test format is the primary interest of the study and contains two 15-minute stages containing 10 questions each. The purpose of having 20 questions in total is to identify whether the multistage test can achieve similar results to the fixed length test while requiring fewer questions. In the first stage, there is a single testlet which all participants will take whereas the second stage has three testlets. The difficulty of the questions chosen for the testlet in the first stage ranges between easy and medium questions. The three testlets in the second stage are separated by difficulty (containing easy, medium, and hard questions respectively) and a single testlet is chosen based on the participant's performance in the first stage. Questions chosen for the "easy" testlet typically have a difficulty of 0.4 and below, medium questions a difficulty of 0.4 to 0.7, and hard are 0.7 and above.

4.2. Software Design

4.2.1. Backend API

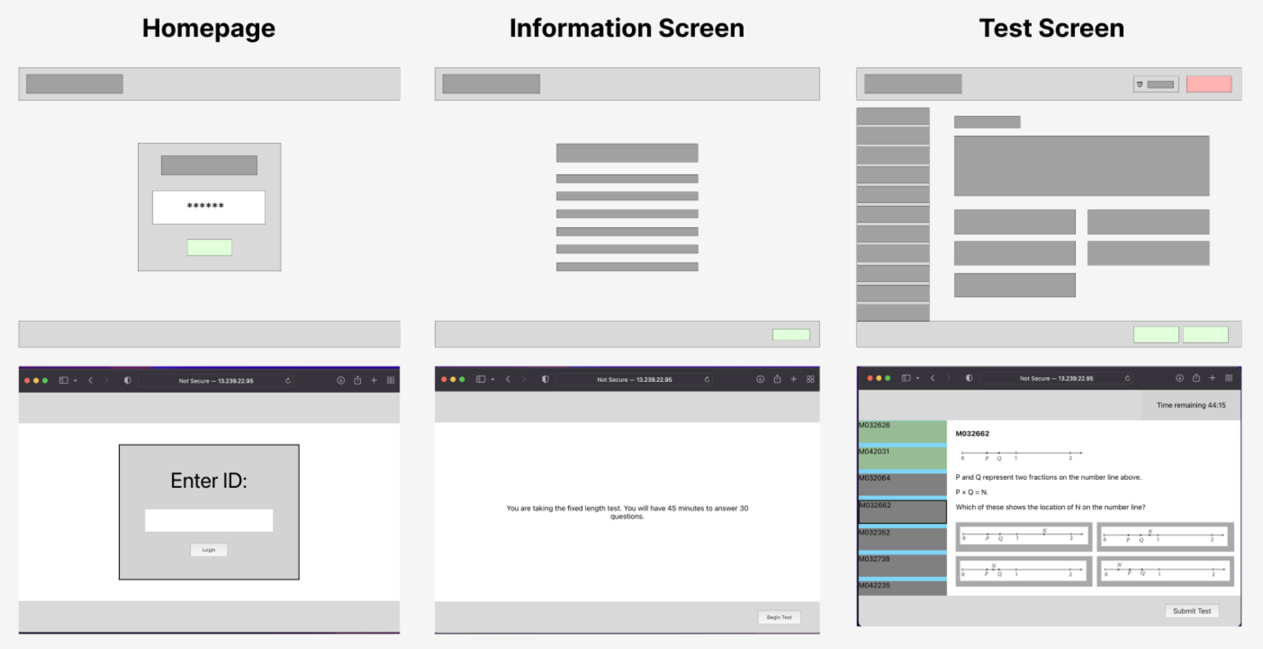
The application backend provides various REST API endpoints which implement functionality such as adding and retrieving questions and users as well as test submission with ability estimation. The API is separated into four main components, the Users API, the Questions API, the Fixed Length Test API, and the Multistage Test API. The Users API provides a /users endpoint that allows for creation and deletion of users by sending POST and DELETE requests as well as retrieval of individual user data by sending a GET request with the user ID appended to the path e.g. /users/12345. The Questions API provides similar functionality for questions. Trying to retrieve non-existent questions or users will result in a HTTP 404 Not Found response. The Fixed Length and Multistage APIs allow users to retrieve the questions specific to each test/testlet. Retrieving questions from the Multistage API requires specifying the testlet number (0-3) where 0 refers to the initial testlet in the first stage while 1, 2, and 3 refers to the three testlets (easy, medium, hard) in the second stage. The Test APIs also allow users to submit their test answers by sending a POST request to the respective endpoint. Error checking is also included to ensure that the answered questions belong to the correct test/testlet otherwise a HTTP 400 Bad Request response is generated. The Multistage API should also calculate an ability estimate for the user on testlet submission and select the testlet that the user should take in the second stage of the multistage test.

4.2.2. Database Structure

The database is a document store composed of two collections: Users and Questions. The User collection contains a document for each user which stores information such as their ID, fixed length test score, multistage test score, as well as the responses for each test. Additionally, the array shouldTakes and moduleNumber tracks the test that the user should take next. These fields are updated on test/testlet submission. The Question collection contains documents which store information about each question such as the ID, question type, difficulty, text and/or image content, correct answer, and the question options (for multichoice questions). A key difference between multichoice and short answer questions is the format in which the correct answer is stored. For multichoice questions, each question has four choices that each have their own ID and the correct answer is represented by the ID of the correct option. This format was chosen over storing the answer text as it allows options to contain images and/or text without requiring additional logic. This is different for short answer questions which only have a text answer therefore string comparison of answers is sufficient. Usability is also considered by making text answers case-insensitive to avoid punishing incorrect capitalization. It is also important to consider that a short answer question may have multiple equivalent correct answers (e.g. 100cm and 1m). However, a proper solution to the issue is beyond the scope of this project.

4.2.3. Frontend Design

As shown in Figure 1, the application frontend uses a simple design comprised of three main screens: the homepage/login, the information screen, and the test screen. The homepage contains a sign-in area that users can input their ID to log into the application. Once logged in, they will be shown an information screen describing the test that they will take and any other instructions. Once the user is ready to begin the test, they can click the "Begin Test" button at the bottom right of the information screen and they will be presented with the test screen. The test screen implements the most important functionalities of the frontend such as test navigation using the sidebar and answering questions either through multiple choice selection or as a written answer. Questions and multiple-choice options can be presented as an image and/or text. The test screen also implements a timer which automatically submits the test once the timer reaches zero.



You are taking the fixed length test. You will have 45 minutes to answer 30 questions.

P and Q represent two fractions on the number line above.

P × Q = N.

Which of these shows the location of N on the number line?

Figure 1. The Figma design and the corresponding screens in the implementation

In terms of UI flow, the fixed length test has a single instruction screen after the login screen then the test begins. After the user submits the test, they are shown another instruction screen which informs them that the test has finished and that the window may be closed. The multistage test is slightly more complicated as there are two stages of tests that the user must take. As a result, there is another information screen to separate the two stages.

5. Implementation

5.1. Technologies Used

Spring Boot was chosen as the backend framework for the project due to being simple to set up but also customizable enough to allow for integration with R using the package RCaller. Support for R integration is crucial as the application relies on the mstR package to provide the algorithms required for multistage testing. mstR was chosen as it is highly customizable, actively maintained and provides detailed documentation for each function.

The application uses MongoDB which is a NoSQL database. This was chosen over a SQL primarily due to having more experience with MongoDB from past projects. A key benefit of MongoDB over SQL databases is having flexible schema which allows changes to be easily made to the database design [18]. This was highly beneficial as it allowed for fast prototyping during the initial stages of development. The database is hosted remotely and can be accessed directly using the GUI provided by MongoDB Compass. This was especially helpful during debugging as having a live view of the database allowed for identification of any invalid behaviour caused by incorrect code.

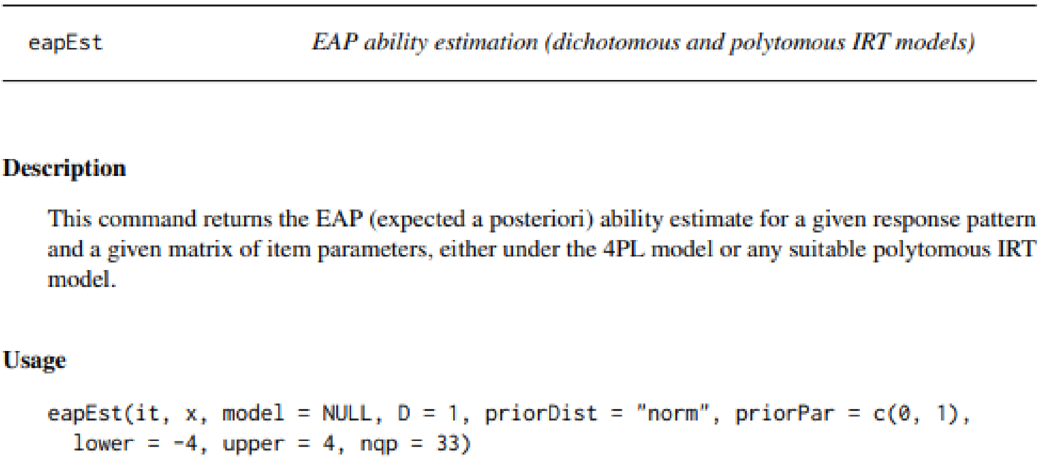
The application frontend is created using React. Application state is stored in the context which allows different components to read and manipulate state. The main advantage of this approach is that there is less code required to handle passing state between parent and child components. However, a drawback is the tight coupling between components and application state which reduces modularity. Another frontend feature is the use of the localStorage React hook to store the current test responses into browser storage so that users can resume the test if they accidentally close the window.

5.2. Multistage Testing Interface

For this project, the functions eapEST and nextModule from the mstR package were used for ability estimation and testlet selection, respectively. A benefit of these functions is that they are stateless which means they can be called as necessary by the backend without any side effects. As a result, integration with the Spring Boot REST API using RCaller is simple once the function parameters are chosen.

5.2.1. Ability Estimation

The eapEst function, shown in Figure 2, takes in the item parameters, item responses, IRT model, metric constant, prior distribution, prior parameters, lower and upper bounds of numerical integration, and the number of quadrature points.



eapEst *EAP ability estimation (dichotomous and polytomous IRT models)*

Description

This command returns the EAP (expected a posteriori) ability estimate for a given response pattern and a given matrix of item parameters, either under the 4PL model or any suitable polytomous IRT model.

Usage

```
eapEst(it, x, model = NULL, D = 1, priorDist = "norm", priorPar = c(0, 1),
  lower = -4, upper = 4, nqp = 33)
```

Figure 2. The documentation for the ability estimation function

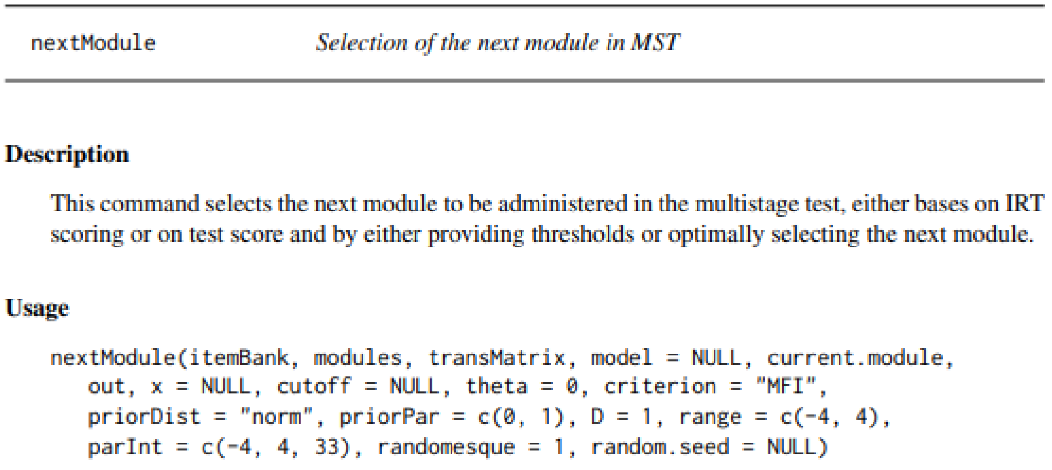
The item parameters are represented by a four-column matrix where each item is represented by a row in a matrix containing the item discrimination, difficulty, lower asymptote, and upper asymptote. Usage of all four parameters is required when using a 4PL IRT model. However, as stated earlier, the IRT model used for this project is 1PL which only requires the difficulty parameter hence the other parameters were kept as the default values i.e. discrimination = 1, lower asymptote = 0, upper asymptote = 1.

The second parameter is the choice of IRT model which is kept at NULL to represent dichotomous models which are used for items that only have two responses, correct or incorrect. The function also provides support for polytomous models for Likert-scale items but they are not used for this project.

As the theory behind MST and CAT extends far beyond the scope of this project, the remaining parameters are left as the default values as they were found to be sufficient for the needs of the application. The metric constant, prior distribution, and prior parameters represent the estimated proportion of abilities of the population before evidence from testing is considered. It is reasonable to believe that much like other human traits such as height or IQ, mathematical ability can also be modelled using a normal distribution hence the normal distribution is chosen for ability estimation with a mean of 0 and standard deviation of 1. The lower and upper bounds of numerical integration are –4 and 4 which represent the minimum and maximum ability estimates, respectively. The number of quadrature points is 33 which is the sequence of values between –4 and 4 separated by a step size of 0.25 which are used for the numerical integration in the ability estimation formula.

5.2.2. Testlet Selection

The other function used for this project is the nextModule function, as shown in Figure 3, which takes in the item parameters, the modules/testlets that the items belong to, multistage test structure, IRT model, current module, previously answered questions, module selection thresholds, selection method, as well as various values specifying the type and parameters of the prior distribution.



nextModule *Selection of the next module in MST*

Description

This command selects the next module to be administered in the multistage test, either bases on IRT scoring or on test score and by either providing thresholds or optimally selecting the next module.

Usage

```
nextModule(itemBank, modules, transMatrix, model = NULL, current.module,
   out, x = NULL, cutoff = NULL, theta = 0, criterion = "MFI",
   priorDist = "norm", priorPar = c(0, 1), D = 1, range = c(-4, 4),
   parInt = c(-4, 4, 33), randomesque = 1, random.seed = NULL)
```

Figure 3. The documentation for the module selection function

The item parameters are represented by the same matrix as in eapEst, however, the nextModule function also requires knowledge of which module(s) that each item belongs to which is represented using a binary matrix. The transition matrix is another binary matrix which specifies which modules that a module can transition to i.e. a value of 1 at row *i* and column *j* means that module *i* can move to module *j*. In this project specifically, the transitions that are allowed are from the initial module (module 0) to the easy, medium, or hard module (modules 1, 2, and 3). This information is used along with the current module and the ability estimate obtained from eapEst to select the next module to take.

The method for module selection is decided by the criterion parameter and can be either maximum Fisher information, maximum likelihood weighted module information, maximum posterior weighted module information, and more. Additionally, a cut-off can be supplied which overwrites the previous methods to allow for setting custom ability thresholds within which a module can be selected. For simplicity, the default method of maximum Fisher information is used.

Next are the parameters related to the prior distribution and these were kept the same as the values used in eapEst for consistency. This means the prior ability distribution is a normal distribution with a mean of 0 and standard deviation of 1. Lastly, the function also provides the option of only selecting the optimal module with a fixed probability by setting "randomesque" to a value less than one. This is used to prevent overexposure of a particular module. However, this option is kept at 1 for the project as the likelihood of participants sharing question information is very low.

5.3. Deployment

As the application had to be available online, an AWS Elastic Compute Cloud (EC2) instance was created with Java, R and mstR installed. A virtual machine such as EC2 is chosen over simpler platforms such as Netlify or Heroku because it provides the customizability and control required to integrate R into the Spring Boot backend. Initial setup involved installing Java, R and the mstR package as well as creating a private key for SSH access. Deployment is a manual process and involves building a jar file of the project which contains the Spring Boot backend bundled with the static JavaScript files for the frontend. The jar file is then uploaded to the EC2 instance via SCP and the application is launched.

6. Results

6.1. Testing Protocol

After the application was implemented, user testing was conducted to identify whether the project met the original goals of developing an online testing platform and evaluating the effectiveness of multistage testing. Testing was conducted on 15 participants whereby each participant would take the multistage and fixed length test consecutively. Participants were informed of the two different testing paradigms that the test domain but did not have any knowledge of the exact test content prior to the test. One of the potential sources of bias that was identified prior to testing was the practice effect. This is an issue because participants are required to take two tests in succession which means performance may increase in the second test merely from repeating the activity [17]. To mitigate this effect, roughly half of the participants sat the fixed length test first and the other half sat the multistage test first. After completing both tests, the participants were then asked to complete a Google Forms questionnaire discussing their testing experience and provide feedback.

6.2. Test Scores

The multistage and fixed length test scores for each participant were recorded and plotted as shown below. The fixed length score is calculated as the proportion of questions that the participant answered correctly and their multistage score is taken directly from the output of the ability estimation function in mstR. The graph shows that all participants scored highly on both tests which is unsurprising as the questions are targeted at 14 and 15 year olds whereas most of the participants are university students. It is also interesting to note that all participants were administered the hard test in the second stage of the multistage test. A key feature of the graph is the correlation between the two test scores. The trend line shows that there is a positive correlation between multistage test score and fixed length test score which is to be expected as both tests cover the same content and therefore performance in both tests should be correlated. However, the R-squared value of 0.28 shows that this correlation is weak. This combined with the fact that all of the data is clustered in a single region suggests that there is insufficient evidence to conclude whether the multistage test can achieve more accurate ability estimates than the fixed length test.

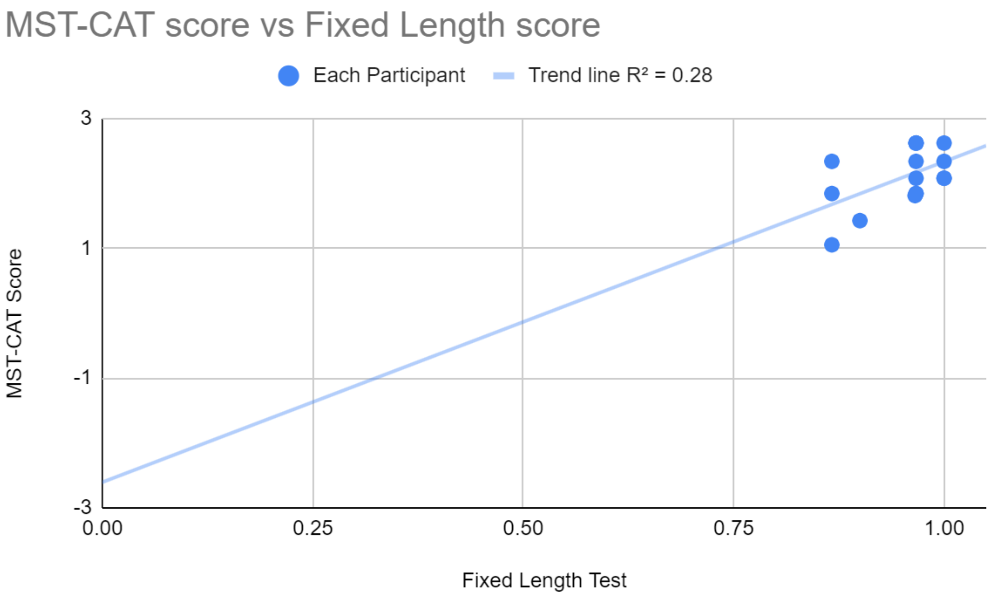


Figure 4. A plot comparing multistage and fixed length test scores

6.3. User Feedback

User testing also involved gathering feedback from each participant via a questionnaire. The graphs below show the responses to some of the Likert scale questions which used a scale from 1-5 where 1 means the participant strongly disagrees with the statement and 5 means the participant strongly agrees.

Part of the questionnaire is focused on the usability of the application and the responses show that the overall user experience of the application is positive. This can be seen in Figure 5 where 90% of users found it easy to navigate the platform and the user interface intuitive. A common theme in the written feedback is that the UI is simple albeit slightly bland and lacking some nice-to-have features such as rich text support and more ways to navigate between questions. This is to be expected as the application is still in early stages of development and only implements the features required initial user testing.

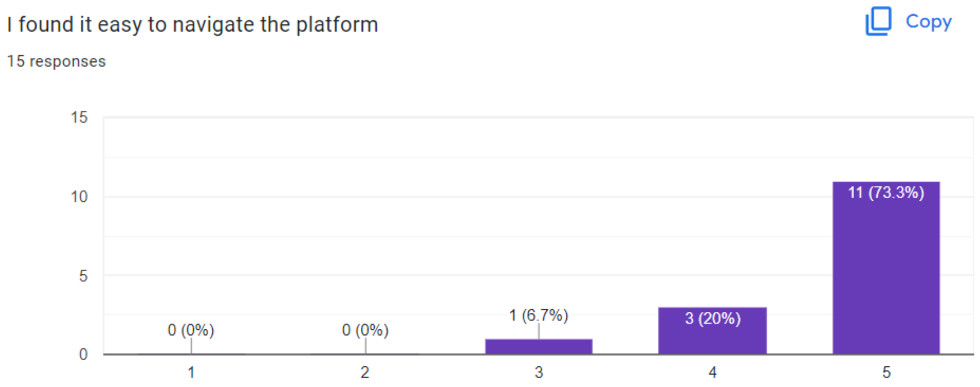


Figure 5. User feedback on ease of navigation

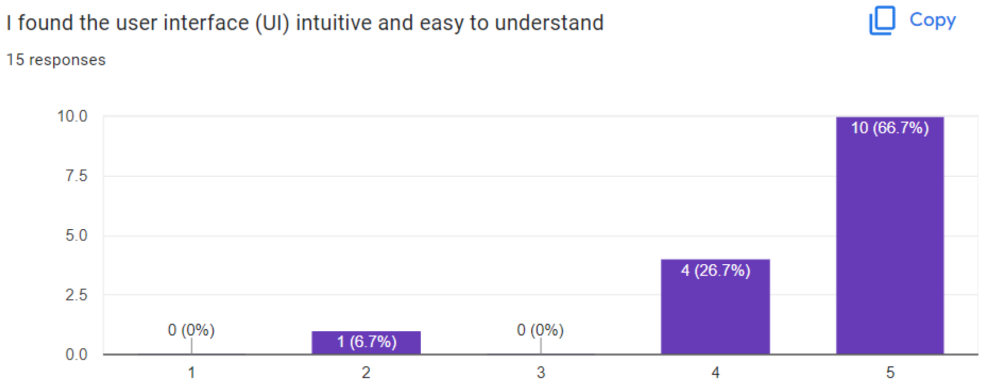
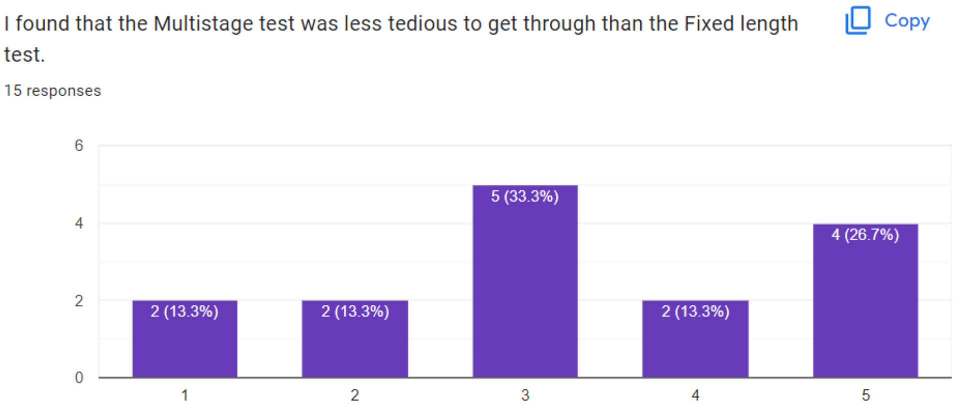


Figure 6. User feedback on user interface intuitiveness and ease of understanding

One question asked participants how they found the length of the two different tests. Figure 7 shows that user perceptions of test length were mixed and there is insufficient evidence to suggest that one test is more engaging than the other. This is surprising given that the multistage test had 10 fewer questions in total than the fixed length test. However, some participants noted that they preferred being able to complete the entire test at once over the disjoint nature of the multistage test.

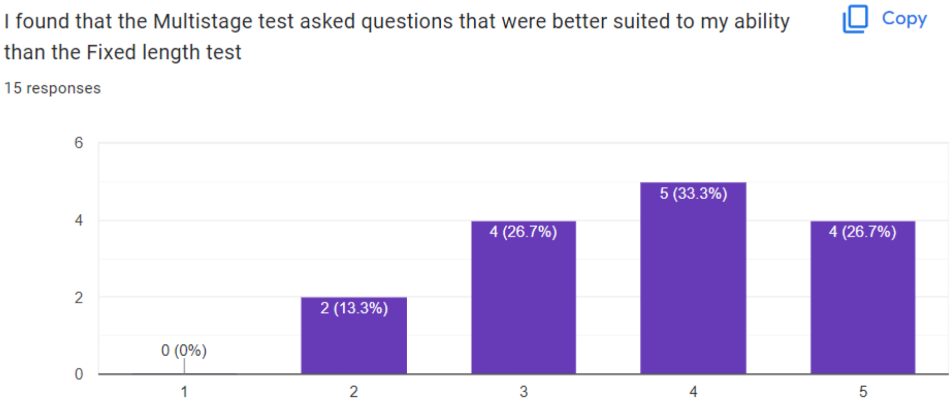


I found that the Multistage test was less tedious to get through than the Fixed length test.

15 responses

Figure 7. User feedback on perceived tediousness of both tests

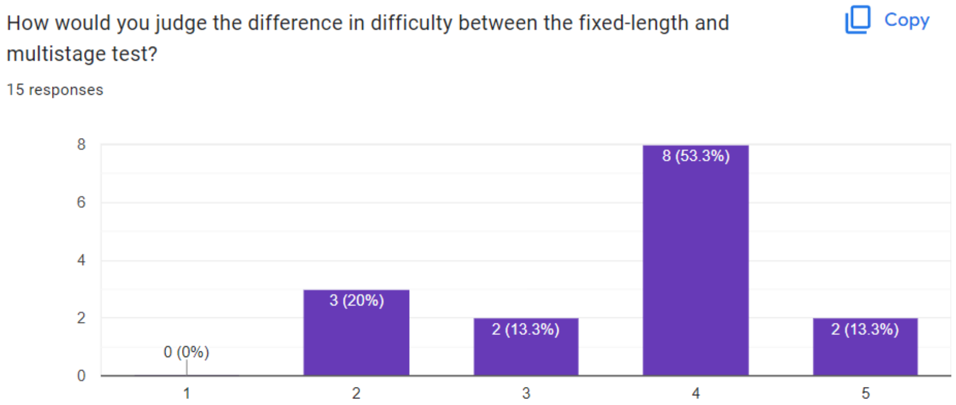
The questionnaire also requires participants to provide feedback on the perceived difficulty of both tests. Interestingly, Figure 8 shows that 60% of participants found that the questions in the multistage test were better suited to their ability compared to the 13.3% favouring the fixed length test. The feedback on test difficulty in Figure 9 also showed that over 60% of participants found the multistage test harder than the fixed length test. These results suggest that the fixed length test was too easy for the participants and this is also reflected in the test scores. This may also indicate that the multistage test is functioning as intended because high performing participants are given the harder testlet in the second stage.



I found that the Multistage test asked questions that were better suited to my ability than the Fixed length test

15 responses

Figure 8. User feedback on perceived appropriateness of both tests



How would you judge the difference in difficulty between the fixed-length and multistage test?

15 responses

Figure 9. User feedback on perceived difficulty of both tests. Note: a score of 1 meant the fixed length test was much harder and 5 meant the multistage test was much harder.

6.4. Limitations

Although measures were taken in the test design to increase the validity of the results, there are still limitations that exist and need to be addressed.

A major limitation of the study is the small sample size of only 15 participants which produced insufficient data to draw strong conclusions from. Additionally, there was a significant difference between the intended demographic and the actual demographic of the test participants. Because the questions were designed for Year 9 students (ages 13-14), they were far too easy for the actual participants of the test which were primarily university students with strong math backgrounds. This resulted in most of the participants achieving almost perfect scores and exceeding the level that can be accurately measured by the test. The difference between targeted demographic and test demographic also affects the validity of the usability feedback as each user's preferences are subjective and likely to vary between age groups [19]. Lastly, the test participants were not anonymous and many were friends with the researchers which could have resulted in them giving more positive feedback than anonymous participants.

Another limitation is that some of the questions had to be altered to conform to the format of the test. For example, some of the original long-answer questions had to be restructured into a multiple-choice question to allow for automatic marking. A consequence of these changes is that the true difficulty of the modified questions may no longer reflect the original difficulty parameter from the TIMSS question set and hence result in a less accurate ability estimate.

7. Discussion

One of the goals of this project is to evaluate that effectiveness of multistage testing against traditional fixed length testing. User testing shows that multistage testing can identify high ability as evidenced by the fact that all the participants with high fixed length scores were also administered the hard testlet in the second stage of the multistage test. This is also confirmed by user feedback from the questionnaire also shows that participants generally found the multistage test harder. However, the testing did not cover a large range of ability and hence conclusions cannot be drawn about the effectiveness of the multistage test at differentiating between different abilities. Interestingly, there was no noticeable preference for either test in terms of perceived tediousness even though the multistage test had 10 fewer questions than the fixed length test. This can be attributed to some users prefer being given the entire set of questions at once whereas others preferred having the test separated into multiple, shorter stages.

The second goal of the project is to assess the feasibility of implementing a CAT testing platform using widely available technologies. A basic testing platform has been successfully implemented using React, Spring Boot and mstR. One challenge in the project is combination of software development experience and knowledge of CAT/MST theory required to create the application. Having knowledge in both areas is rare and those wanting to create a similar application will most likely need to conduct research to fill the gaps in their knowledge. However, the abundance of free resources and open-source software available makes this an achievable task. The implementation for this project is an online web application that supports both fixed length and multistage testing. Users are provided with a simple and intuitive user interface to complete the tests and their scores are persisted in a database. User testing has been conducted on 15 participants, each of which had the opportunity to provide feedback on the application in the post-test questionnaire. Overall, users found the application easy to use albeit somewhat bland and there were no major bugs during testing. The application can therefore be considered a good starting point and, with more development time, has potential for further use in CAT research.

8. Conclusions

This project has contributed to the domain of computer adaptive testing and multistage testing. The lack research available in the applications of CAT in the high school curriculum has motivated the development of an online testing platform targeted towards Year 9 and Year 10 students in New Zealand. The application aims to compare the effectiveness of multistage testing against traditional fixed length test without requiring the use of any proprietary technologies. The implementation included designing a fixed length and multistage test and implementing an online testing platform to support user testing.

User testing was conducted to acquire both quantitative and qualitative feedback regarding the differences between the two tests and the application itself. Although the testing was conducted on subjects outside the targeted demographic, the results show that the multistage test can identify high ability users and there is a positive correlation between performance in the fixed length and multistage tests. Additionally, participants were asked to provide feedback on the testing platform. Feedback on the tests showed that users generally found the multistage test to be more difficult and better suited to their ability. Regarding the user experience, participants generally found the application simple and intuitive but would benefit from more features, some of which are covered in the following section.

9. Suggestions for Future Work

There are many opportunities for further exploration in this area of research. An example is acquiring ethics approval and conducting an official study on high school students in New Zealand. This would greatly improve the validity of the research and allow for more thorough analysis of results. The implementation also has potential to be improved significantly. One example is fine-tuning the parameters used in the multistage test. Currently, many of the parameters used in the mstR functions are the default values provided by the library and ability estimation would most likely be improved by experimenting with different IRT models and prior distributions. Lastly, the user experience of the testing platform would benefit from additional features such as support for mathematical symbols and improved automated marking that allows for multiple correct answers. With these improvements in mind, the application serves as a promising initial implementation with potential for further development and use in research.

Acknowledgements

The student would like to thank the project supervisors Yu-Cheng Tu and Andrew Meads for providing their support and guidance to make this project successful. The student would also like to thank their project partner, Hajin Kim, for their contributions to the research and implementation of the project.

References

[1] B. Ussher and K. Earl, "'Summative' and 'Formative': Confused by the Assessment Terms?" New Zealand Journal of Teachers' Work, vol. 7, no. 1, pp. 53-63, 2010.

[2] D. J. Weiss and G. G. Kingsbury, "Application of computerized adaptive testing to educational problems," Journal of Educational Measurement, vol. 21, no. 4, pp. 361–375, 1984.

[3] E. B. Andersen, "The Rating Scale Model," Handbook of Modern Item Response Theory, pp. 67–84, 1997.

[4] G. N. Masters, "A Rasch model for partial credit scoring," Psychometrika, vol. 47, no. 2, pp. 149–174, 1982.

[5] S. L. Wise and G. G. Kingsbury, "Practical issues in developing and maintaining a computerized adaptive testing program," Psicologica, vol. 21, pp. 135-155. 2000.

[6] D. J. Weiss, "Improving measurement quality and efficiency with adaptive testing," Applied Psychological Measurement, vol. 6, no. 4, pp. 473–492, 1982.

[7] R. J. Owen, "A bayesian approach to tailored testing," ETS Research Bulletin Series, vol. 1969, no. 2, pp. i-24, 1969.

[8] V. M. G. Jatobá, J. S. Farias, V. Freire, A. S. Ruela, and K. V. Delgado, "Alicat: A customized approach to item selection process in computerized adaptive testing," Journal of the Brazilian Computer Society, vol. 26, no. 1, 2020.

[9] A. Zenisky, R. K. Hambleton, and R. M. Luecht, "Multistage testing: Issues, designs, and research," Elements of Adaptive Testing, pp. 355–372, 2009.

[10] O. Rotou, L. Patsula, M. Steffen, and S. Rizavi, "Comparison of multistage tests with computerized adaptive and paper-and-pencil tests," ETS Research Report Series, vol. 2007, no. 1, pp. i-27, 2007.

[11] T. J. H. M. Eggen, Overexposure and underexposure of items in computerized adaptive testing. Arnhem Netherlands: Citogroep, 2001.

[12] D. Magis, D. Yan, and A.A. von Davier, Computerized Adaptive and Multistage Testing with R: Using Packages catR and mstR. Springer, 2017.

[13] C. Wendler and B. Bridgeman, The Research Foundation for the GRE Revised General Test: A Compendium of Studies. Education Testing Service, 2014.

[14] "NAPLAN – general," National Assessment Program. [Online]. Available: https://www.nap.edu.au/naplan/faqs/naplan--general. [Accessed: 13-Apr-2022].

[15] "Tailored test design study 2013: Summary research report," National Assessment Program. [Online]. Available: https://nap.edu.au/docs/default-source/default-documentlibrary/tailored_test_design_study_2013_summary_research_report.pdf. [Accessed: 13-Apr-2022].

[16] "TIMSS 2011 Assessment Frameworks," Trends In International Mathematics And Science Study. [Online]. Available: https://timssandpirls.bc.edu/timss2011/downloads/TIMSS2011_Frameworks.pdf. [Accessed: 10-Oct-2022].

[17] M. A. McDaniel, R. P. Fisher, "Tests and test feedback as learning sources," Contemporary Educational Psychology, vol. 16, no. 2, pp. 192–201, 1991.

[18] Z. Parker, S. Poe, and S. V. Vrbsky, "Comparing NoSQL MongoDB to an SQL DB," Proceedings of the 51st ACM Southeast Conference on - ACMSE '13, 2013.

[19] J. Kim, J. Kim, and J. Y. Moon, "Does Age Matter in Mobile User Experience? Impact of Age on Relative Importance of Antecedents of Mobile User Experience" PACIS 2013 Proceedings, 2013.