November 2023

I've (Urn)ed This: An Application and Criterion-based Evaluation of the Urnings Algorithm

Ted Daisher
University of Massachusetts Amherst

I've (Urn)ed This: An Application and Criterion-based Evaluation of the Urnings Algorithm

A Dissertation Presented

by

TED E. DAISHER

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2023

College of Education

Research, Educational Measurement & Psychometrics

I've (Urn)ed This: An Application and Criterion-based Evaluation of the Urnings Algorithm

A Dissertation Presented

by

TED E. DAISHER

Approved as to style and content by:

_____
Lisa A. Keller, Chair

_____
Craig S. Wells, Member

_____
Logan Rome, Member

_____
Shane Hammond
Associate Dean for Student Success
College of Education

ABSTRACT

I'VE (URN)ED THIS: AN APPLICATION AND CRITERION-BASED EVALUATION

OF THE URNINGS ALGORITHM

SEPTEMBER 2023

TED DAISHER, B.A., DEPAUL UNIVERSITY

M.A., KENT STATE UNIVERSITY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Lisa A. Keller

There is increased interest in personalized learning and making e-learning environments more adaptable. Some e-learning systems may use an Item Response Theory (IRT)-based assessment system. An important distinction between assessment and learning contexts is that learner proficiency is expected to remain constant across an assessment, while it is expected to change over time in a learning context. Constant learner proficiency during an assessment enables conventional approaches to estimating person and item parameters using IRT. These IRT-based systems could be abandoned for alternative approaches to modeling learners and system learning content, but assessments may provide more functions than adapting learning material to students. Thus, there is the question, how can e-learning systems with IRT-based assessment components more dynamically adapt their learning content? Is there a solution that leverages IRT for adapting the learning content of the system?

A promising solution is the Urnings algorithm. Like other candidate algorithms, it is computationally light, but this algorithm has mechanisms for preventing variance inflation and is suitable for e-learning contexts. It also provides a measure of uncertainty around estimates. It has been studied both through simulations and applications to e-learning systems. Results are promising; however, there has not been an application of the Urnings algorithm to an e-learning context where there are conventionally estimated person parameters to compare the algorithm estimates to. This study addresses this gap by applying the Urnings algorithm to a K–8 reading and mathematics learning platform. In data from this platform, we have person parameter estimates across academic

years from an in-system diagnostic assessment. Results from this study will help industry researchers understand the feasibility of the Urnings algorithm for large e-learning systems with IRT-based assessment components.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

There is increasing interest in personalized learning because of the benefits it can have for students**.** Personalized learning is when elements of learning are varied to align with a learner's needs and interests. Learning objectives, instructional approaches, instructional content, or the sequence of content can all be varied (U.S. Department of Education, 2017). Personalized learning can be supported through e-learning systems designed to meet the needs, goals, talents, and interest of learners (Klašnja-Milićević et al., 2015). Wolf (2010) stated that technology can enhance personalized learning. Alamri et al. (2021) went further in stating it can be challenging to personalize learning without the support of information technology platforms. Sturgis et al. (2021) similarly argued technology is critical in personalizing learning. As a result of an increasing focus on customizing students' instruction, and the role technology plays in enabling that, there has been a drive in developing digital platforms that differentiate instruction (Johnson et al, 2015).

There is also increasing interest in making e-learning environments more adaptable. Some e-learning systems may use an Item Response Theory (IRT)-based assessment system such as IXL Learning (IXL Learning, 2020) and the Reading Plus program from DreamBox Learning (Reading Plus, 2021). An important distinction between assessment and learning contexts is that learner proficiency is expected to remain constant across an assessment, while it is expected to change over time in a learning context (Galvez et al., 2016). Constant learner proficiency during an assessment enables conventional approaches to estimating person and item parameters using IRT (Galvez et al., 2016). These IRT-based systems could be abandoned for alternative approaches to modeling learners and system learning content, but assessments may provide more functions than adapting learning material to students. For example, assessment results could be used for reporting purposes or to help instructors plan interventions with students. Thus, there is the question, how can e-learning systems

with IRT-based assessment components more dynamically adapt their learning content? Is there a solution that leverages IRT for adapting the learning content of the system?

Two solutions that have been attempted are progress testing and growth modeling. Progress testing is testing with frequent, relatively short formative assessments (e.g., posttests when learners finish a unit of learning material). Growth modeling involves extensions of IRT that explicitly model for changes in a leaner's proficiency over time. The disadvantage of progress testing is that it is generally considered intrusive, and e-learning designers are moving away from this kind of direct form of measuring students (Abyaa et al., 2019; Normadhi et al., 2019). The disadvantage of growth modeling is that estimating the parameters of these models is usually a computationally intensive process (Papoušek et al., 2014; Pelánek, 2016); thus, it would be difficult to scale this approach up to large systems making many on-the-fly estimations.

A third possible solution is algorithms that work with IRT. Unlike progress testing, these algorithms measure learner characteristics non-intrusively as learners complete learning tasks. Unlike growth modeling, these algorithms are computationally light weight, making it easy to scale them up to large systems (Brinkhuis et al., 2018). Some candidate algorithms are the Elo Rating System (Elo, 1978; ERS), Glicko (Glickman, 1999), and TrueSkill (Herbrich et al., 2006); however, the ERS and Glicko suffer from variance inflation and deflation, undermining the ability to measure growth over time (Bolsinova et al., 2022; Hofman et al., 2020). Being able to measure growth is important in an e-learning context. The ERS also does not provide a measure of uncertainty around learner and item estimates (Bolsinova et al., 2022; Glickman, 1999, Hofman et al., 2020). There is not direct evidence that TrueSkill suffers from variance inflation and deflation, and it provides a measure of uncertainty around estimates; however, it was built for team-based multiplayer online gaming (Herbrich et al., 2006); thus, it has features that make it unsuitable for e-learning contexts. For example, along with evaluating individual performance, the algorithm also considers the performance of the team each

player is on (e.g., Did the player do well, but their team was defeated in the match?). This likely does not apply to many individual-focused e-learning contexts.

A promising alternative algorithm is the Urnings algorithm. Like the other algorithms, it is computationally light, but this algorithm has mechanisms for preventing variance inflation and is suitable for e-learning contexts. It also provides a measure of uncertainty around estimates (Bolsinova et al., 2022; Hofman et al., 2020). It has been studied both through simulations and applications to e-learning systems. Results are promising; however, there has not been an application of the Urnings algorithm to an e-learning context where there are conventionally estimated person parameters to compare the algorithm estimates to. This study addresses this gap by applying the Urnings algorithm to a K-8 reading and mathematics learning platform. In data from this platform, we have person parameter estimates across academic years from an in–system diagnostic assessment. Results from this study will help industry researchers understand the feasibility of the Urnings algorithm for large e-learning systems with IRT-based assessment components.

CHAPTER 2

LITERATURE REVIEW

The following is a review of research relevant to the problem of dynamically adapting content in a learning system with an IRT-based assessment component, with the key pre-requisite to doing so being accurately following proficiency as it changes in the system. First, contextualizing information is given on adaptive learning systems, the parts of these systems, the different types of these systems, and the techniques that have been used to model proficiency within these systems. This section is meant to show the broader field of work on adaptive learning systems this IRT-focused study is situated within. Second, how IRT has been used in adaptive learning systems is described—with a focus on the approaches of progress testing, modeling change, and tracking. The organization of this section follows how dealing with changing proficiency was, over time, first tackled with progress testing, then with attempting to model change directly, and most recently with algorithms that track proficiency. The third approach marks a shift from having learning and assessment separate to an approach that integrates them, continually measuring proficiency while a learner is learning. The tracking section also weighs the advantages and disadvantages of different potential algorithms, building the argument for the Urnings algorithm as a promising method for following changing proficiency.

2.1 What is an adaptive learning system?

Many e-learning environments are static, meaning each learner gets the same information, through the same structure, using the same interface. Some e-learning systems are adaptive, meaning they adapt in some way to meet the needs or preferences of learners (Brusilovsky, 1999). These systems can be called adaptive learning systems (ALS). These adaptive learning environments are part of a new generation of e-learning systems (Normadhi et al., 2019). Adaptation in these systems happens with the goal of providing an efficient and enjoyable learning experience (Shute & Towle, 2003; Shute & Zapata-Rivera, 2007; Vagale & Niedrite, 2012).

2.1.1 What are the typical parts of an adaptive learning system?

In most cases, ALSs are based on four models (Chrysafiadi & Virvou, 2015; Shute & Towle, 2003; Vagale & Niedrite 2012; Vandewaetere et al. 2011). The first is the learner model (also called the student model), which contains characteristics of the learner. The second is the domain model (also called the content model), which contains characteristics of the material in the system. The third is the media model (also called the instructional or presentation model), which monitors learners' interaction with content and adjusts the presentation of material to support learning (e.g., providing hints to a student when they struggle to solve a problem). The fourth is the adaptation model (also called the adaptive engine) which guides the matching of learners to system material (Chrysafiadi & Virvou, 2015; Shute & Towle, 2003; Vagale & Niedrite 2012; Vandewaetere et al. 2011).

Another part of ALSs is the learner profile. The learner profile is a part of the learner model. The learner profile contains static information about the learner that is typically not used by the adaptation model (Abyaa et al., 2019). This includes information such as age and name (Abyaa et al., 2019). The other part of the learner model is what is measured by the ALS and is typically used for adaptation (Abyaa et al., 2019; Sani, 2016). This can include characteristics like a learner's proficiency in the domain subject.

To illustrate the models of an ALS, a system for teaching algebra may represent a learner through their proficiency as measured by Item Response Theory (the learner model). Course material or items in the system may be represented by their difficulty, also measured through Item Response Theory (the domain model). As a learner works through algebra problems, the system may provide hints when the learner gets a step in a problem wrong (the media model). As the learner's proficiency grows, the system will match them to content appropriate for the learner based on their proficiency estimate (the adaptation model).

2.1.2 What are the parts and methods for adaptation in an adaptive learning system?

Adaptation in these systems, determined by the adaptation model, can happen in different ways. Wauters et al. (2010) described three ways: form representation, content representation, and curriculum sequencing. Form representation refers to the way material is presented to learners. This includes, for example, whether the content contains pictures, video, or text only. Adaptive content representation refers to the system giving the learner help as needed in, for example, the steps of a problem-solving task. The system helps the learner based on the learner's knowledge gaps that the system has identified. Adaptive curriculum sequencing refers to the system selecting material for a learner that is optimal, given some known characteristic of the learner (e.g., their proficiency; Wauters et al., 2010).

In addition to the ways a system can adapt, Wauters et al. (2010) also described the elements that can be used to guide adaptation. They classified them into three categories: course/item features, person features, and combinations. Course/item features (which would fall within the domain model) refers to characteristics of material in the system, (e.g., the difficulty of items). Person features (which would fall within the learner model) refer to characteristics of learners (e.g., proficiency, motivation, interests). Combinations refers to using both course/item features and person features together (e.g., matching material to learners by the difficulty of material and proficiency of learners; Wauters et al., 2010).

2.1.3 What are the ways that adaptive learning systems model learners?

To model learner characteristics, ALSs use explicit modeling, implicit modeling, or a combination of both (Normadhi et al., 2019). In explicit modeling (also called collaborative modeling) information is collected directly from the learner using methods such as questionnaires (Abyaa et al., 2019; Normadhi et al., 2019). The learner's response to the survey reveals their degree of the target trait. In implicit modeling, information is collected in an indirect and non-intrusive way

(Abyaa et al., 2019). Implicit modeling typically uses some algorithm or program to automatically measure the target trait (Normadhi et al., 2019).

Some of the most common sources of data for explicit modeling are tests, questionnaires, group interactions, peer assessments, and monitoring tools such as face monitoring and eye tracking (Abyaa et al., 2019). Some common sources of data for implicit modeling are the learner's behavior in the system (e.g., log data), search terms they used in the system, natural language inputs, and direct sensors (e.g., keyboard and mouse; Abyaa et al., 2019).

Abyaa et al. (2019) claims that explicit modeling tends to be more reliable. Others consider implicit modeling more accurate (Albadvi & Shahbazi, 2009), as the target trait, depending on what it is, can sometimes change quickly. Implicit modeling is better equipped to detect these changes (Botsios et al., 2008; Graf et al., 2010). Implicit modeling generally seems to be preferred over explicit modeling. Explicit modeling is considered intrusive, which is consequential as it can distract and demotivate learners (Abyaa et al., 2019). This is reflected in research trends. In their review of 107 articles from 2013–2017, Abyaa et al. (2019) found that 76% of studies used implicit modeling. The only exception to this was when a system only used overlay modeling, in which case explicit modeling was more popular. In their review of 78 articles from 2010–2017, Normadhi et al. (2019) found 57 studies used implicit modeling, while only 12 studies used explicit modeling (in the form of questionnaires). However, Normadhi et al. (2019) noted that several studies used a combination of explicit and implicit modeling (about 45% of the reviewed articles).

2.1.4 What learner characteristics tend to be used in the learner model?

Within the four typical models of an ALS, the learner model is considered a central pillar of the system (Abyaa et al., 2019). Many leaner characteristics have been used in the learner model over time. In their review, Abyaa et al. (2019) identified six categories. The first is the learner profile, containing static information about the learner such as name and age. The second is knowledge-

related information, such as the learner's proficiency in the domain, the skills they have mastered, and their misconceptions. The third is cognitive characteristics, such as learner's working memory capacity. The fourth category is social characteristics such as the learner's tendency to be collaborative. The fifth category is personality traits. The last category is characteristics related to motivation, such as the learner's interests and level of engagement. Normadhi et al. (2019) grouped learner characteristics into three main categories based on Bloom's taxonomy. The first, cognition, includes traits like working memory capacity, prior knowledge, and thinking process. The second, affective, includes interests, emotions, and attitudes. The third, behavior or psychomotor, includes physical movements and coordination.

Early work on learner modeling tended to focus on knowledge and related traits (Conati et al., 1997; Jackson et al. 2003; Tennyson, 1975; Tennyson, 1993; Tennyson & Rothen, 1977). This aligns with what Chrysafiadi and Virvou (2013) found in their literature review from 2002 to 2013. They found cognitive traits such as knowledge, problem-solving ability, and critical thinking to be the most used characteristics in student models. For 2002 to 2008, they found knowledge level to be one of most focused on characteristics for student modeling. Nakic et al. (2015), in their review of 98 articles published from 2001 to 2013, found that learning styles was the most popular trait used for learner modeling (appearing in 27.6% of articles). However, background knowledge was the second most frequent (appearing in 16.3% of reviewed articles). Normadhi et al.'s (2019) more recent literature review, covering 78 articles from 2010 to 2017 found that learning styles was the most focused on trait for learner modeling (appearing in 44.87% of articles). This shows a strong historical focus on modeling learner proficiency with a recent shift towards interest in learning styles.

2.1.5 What techniques are used to model learners' traits?

2.1.5.1 Bayesian Knowledge Tracing.

Many techniques have been used to model learners' traits. One common technique is Bayesian Knowledge Tracing (Abyaa et al., 2019). Bayesian Knowledge Tracing was introduced by Corbett and Anderson (1995). The model is a Hidden Markov Model in which student knowledge is a hidden variable and student performance is an observed variable (Sani et al., 2016). Each piece of knowledge or skill is represented with a model and can be represented in two states: learned or not yet learned (Abyaa et al., 2019). The model has four parameters:

1. The probability the learner already knows a skill (prior knowledge)
2. The probability the learner will learn the skill after the item (or after each learning opportunity), even if they have no prior knowledge (learning rate)
3. The probability the learner will answer the item correctly even though they have not learned the skill (guess)
4. The probability the learner will answer the item incorrectly even though they have learned the skill (slip; Abyaa et al., 2019; Sani et al., 2016)

Some limitations of Bayesian Knowledge Tracing are that it cannot account for forgetting (Abyaa et al., 2019). The model assumes that a skill can only go from not yet learned to learned. The parameters for each skill in Bayesian Knowledge Tracing are also constant for a given skill, meaning they do not vary by student (Abyaa et al., 2019). Thus, this model cannot account for individual differences in this way (Sani et al. 2016). A third limitation is that each item is assumed to be associated with one piece of knowledge or skill. If an item represents multiple skills, for Bayesian Knowledge Tracing, it must be decided what one skill the item will represent (Sani et al., 2016).

2.1.5.2 Machine Learning

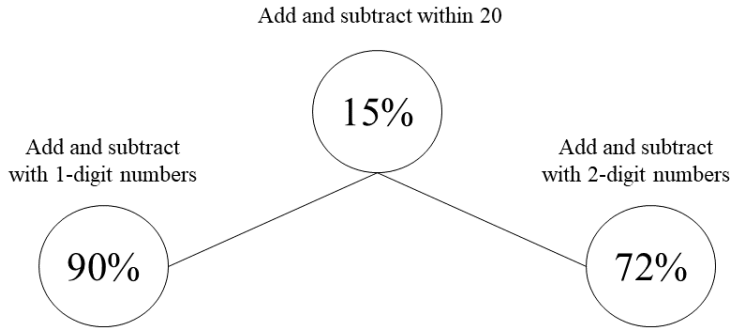
Machine learning is another technique for learner modeling. Observations of a learner's behavior in a system can be used as training data for a model. That model can then predict future actions of the learner, which can guide the system (Webb et al., 2001). One example technique from machine learning that can be used for learner modeling is deep learning, specifically, neural networks (Abyaa et al., 2019). Deep learning is a form of machine learning where machines learn from data to understand the world as hierarchical concepts (Goodfellow et al., 2016). Neural networks are multilayer structures made up of nodes and the connections between nodes. These nodes are divided into layers through which input is passed (Schmidhuber, 2015). A disadvantage of machine learning to student modeling is that it can be difficult to know how the algorithm arrived at a particular outcome based on the predictors (Sani et al., 2016). It is important in the context of learner modeling to have clear support for a particular decision (Sani et al., 2016).

2.1.5.3 Overlay Model

Another very common approach to learner modeling is overlay modeling (Abyaa et al., 2019). This approach was introduced by Stansfield et al. (1976). In the overlay model, knowledge is represented in the same way in both the domain and learner model (Sani et al., 2016). Both are represented as sets of elements—individual topics and concepts (Chrysafiadi & Virvou, 2013). The collection of elements in the domain represents expert-level knowledge of the domain subject (Brusilovsky & Millán, 2007; Liu & Wang, 2007). The elements in the learner model are a subset of the elements of the domain model (Martins et al., 2008; Vélez et al., 2008). The overlay model represents the degree of mastery the learner has (Nguyen & Do, 2008). The difference between the learner and domain model represents the learner's lack of skill (Sani et al., 2016). Figure 1 shows an example of a small overlay model.

Figure 1

Example Overlay Model



In this example, three mathematics skills are represented: adding and subtracting with 1-digit numbers, adding and subtracting with 2-digit numbers and adding and subtracting within 20. The first two skills are prerequisites for the third. This is represented in the placement of the third skill above the others and the lines connecting each of the first two skills to the third. The values in each node of the network represent the probability the learner has mastered the represented skill.

A benefit of the overlay model is that mastery over the elements in the model can be represented in different ways. They can be represented as a binary (i.e., mastered or not mastered), qualitative labels such as poor-average-good, or a probability that the learner has mastered the skill (Brusilovsky & Millán, 2007). Also, like Bayesian Knowledge Tracing, knowledge of each skill can be represented independently (Chrysafiadi & Virvou, 2013). A limitation of the base form of the overlay model is it can only represent a lack of knowledge. In its base form, it cannot, for example, model misconceptions (Sani et al. 2016).

An extension of the overlay model that can represent misconceptions is the perturbation model (Mayo & Mitroic, 2001). The perturbation model is built on the overlay model by including possible mistakes the learner might make (Martins et al., 2008). Problems are generated based on

these possible mistakes. The generated problem a learner gets wrong indicates what misconceptions they may have about the concept in the problem (Martins et al., 2008).

Another approach that falls within the overlay model category is constraint-based modeling, which was introduced by Ohlsson (1994). In this approach, both the learner and domain are represented with constraints. A constraint is composed of a relevance clause, indicating when in a task the constraint is relevant (e.g., if the task calls for a certain concept or skill), and a satisfaction clause, which details what the learner needs to do in their solution to satisfy the constraint (Martin, 1999). The learner model is the set of constraints the learner does or does not know, and the domain model is the full set of constraints that compose the domain subject (Guerra et al. 2015; Vištica et al. 2016). A benefit of constraint-based modeling is it is computationally simple (Mitrovic et al., 2001). A disadvantage of constraint-based modeling is it does not capture the learner's problem-solving strategy for a problem. It cannot, for example, recognize and accept a novel solution if it violates a constraint (Sani et al., 2016).

2.1.5.4 Stereotyping

Stereotyping is another common approach to learner modeling. This approach was introduced by Rich (1979). Stereotyping involves clustering learners into groups based on some shared characteristic (Chrysafiadi & Virvou, 2013). This approach is often a solution when new learners enter a system, and there is little information for the system to model them (Tsirigi & Viryou, 2002). A benefit of stereotyping is that information about an individual user can be inferred from their stereotype membership (Zhang & Han, 2005). A disadvantage of stereotyping is that learners in the system have to be divisible into meaningful groups, which may not be possible, and these groups have to be defined manually by a system designer (Kass, 1991).

2.1.5.5 Ontologies

Ontologies are a formal naming and defining of categories and concepts in a domain as well as the relationships between those categories and concepts (Abyaa et al., 2019). They provide a common vocabulary to share information about a domain that machines can interpret (Noy & Mcguinness, 2001). The advantage of ontologies is their ability to be extended, their reusability, and their simplicity regarding implementation (Abyaa et al., 2019).

2.1.5.6 Bayesian Networks

There are some techniques in learner modeling used to model uncertainty about traits. Bayesian networks are one common approach to this (Abyaa et al., 2019). A Bayesian network is a directed acyclic graph. The nodes of the graph represent variables, and the arcs between nodes represent the probabilistic dependencies among the variables (Pearl, 1988). In student modeling, the nodes of Bayesian networks can represent different dimensions of the learner being modeled, such as knowledge, misconceptions, and emotions (Chrysafiadi & Virvou, 2013). Bayesian networks are attractive because of their intuitive graphical representation and their strong probability computations of unobserved variables from observed variables (Desmarais & Baker, 2012).

2.1.5.7 Fuzzy Logic

Like Bayesian networks, fuzzy logic is a way of handling uncertainty around a learner characteristic. Fuzzy logic is essentially representing traits in degrees rather than binaries (e.g., a percentage ranging from 0 to 1 rather than just a 0 or 1). Fuzzy logic involves a membership function that, based on input, provides as output the probability of membership in a group (Drigas et al., 2009). Fuzzy logic can be used with data that is imprecise or incomplete, as well as with subjective human judgment (Drigas et al., 2009). A weakness of fuzzy systems is that the rules of the membership function are decided using human logic. Fuzzy systems cannot be trained on data like other

approaches (Sani et al., 2016); thus, when systems become more complex, and more input is considered, building the membership function becomes more difficult (Goel et al., 2012).

2.2 What techniques are most often used for learner modeling?

In their review of articles from 2002 to 2013, Chrysafiadi and Virvou (2013) found that from 2002 to 2007, the overlay model was the most used approach to learner modeling. They also noted that from 2002 to 2007, fuzzy logic was the typical approach to incorporating uncertainty into learner modeling. From 2007 to 2013, they saw an increase in probabilistic approaches to incorporating uncertainty. Sani et al. (2016) found in their review of articles from 2010 to 2015 that stereotyping, fuzzy logic, and Bayesian approaches were the most common. Normadhi et al. (2019) found in their review of articles from 2010 to 2017 that machine learning was the most common approach. Abyaa et al. (2019) in their review of articles from 2013 to 2017 found that in these more recent years, machine learning was the most used learner modeling approach. They note that many researchers used hybrid approaches, combining multiple techniques for learner modeling. There is evidence indicating Bayesian networks for learner modeling may be more effective than other types of learner modeling (Ma et al., 2014).

2.3 What are common types of adaptive learning systems?

The most common types of ALSs are intelligent tutoring systems (ITS) and adaptive hypermedia systems, sometimes also called adaptive educational hypermedia systems (AHS; Abyaa et al., 2019; Brusilovsky, 1999). ITSs typically provide a limited amount of material and are focused on supporting learners in problem solving tasks (Wauters et al., 2010). As Pelánek (2017) explained, these systems tend to focus more on learning complex skills than facts. AHSs typically provide a lot of material in different formats (text, graphics, animation, audio etc.) connected though a linking system (Papadimitriou & Gyftodimos, 2017).

2.3.1.1 Intelligent Tutoring Systems.

ITSs are typically intended to imitate a one-on-one teaching experience, similar to being instructed by a teacher or tutor (du Boulay, 2016). For each learner, an ITS typically 1) presents information to be learned, 2) assigns the learner tasks, 3) provides feedback or hints for tasks based on how the learner is doing, and 4) offers prompts to promote learning and growth (Ma et al., 2014).

ITSs can be divided into two types: task-based and item-based (Wauters et al., 2010). A task-based ITS presents substantial problems which are typically broken down into learnable chunks. The learner tries to form solutions to the tasks and the system provides scaffolding and support (Wauters et al., 2010). An item-based ITS is composed of simple questions. The system usually gives hints and feedback to learners as they respond to questions (Wauters et al., 2010). Task-based ITSs can be further divided into step-based systems and sub-step based systems (VanLehn, 2011). Both types provide help at steps in procedural problem solving. The distinction between the two is that a sub-step system provides feedback at more granular steps than is conventional. The distinction between the two is subjective.

Alabdulhadi and Faisal (2021), in their review of 47 articles published from 2010 to 2018, found that for STEM-related ITSs, most ITSs focused on computer science and engineering, as well as basic computing skills. They also found that these ITSs usually targeted undergraduate students (Alabdulhadi & Faisal, 2021).

The benefits of ITSs are that they offer convenient and low-cost studying support that is not restricted by class time or location (Ding & Cao, 2017). They can also reduce the workload of tutors by helping with grading and decisions around student performance (Paravatiet et al., 2017). A weakness of ITSs is that they cannot fully emulate the complexity and sophistication of one-to-one tutoring with a person (Alabdulhadi & Faisal, 2021).

There is evidence ITSs are effective at helping students learn. VanLehn (2011) reviewed experiments comparing the effectiveness of human tutoring, computer tutoring, and no tutoring. Their meta-analysis with 95 comparisons indicated that step-based ITSs are around as effective as human tutors (d = 0.76 compared to d = 0.79). Ma et al. (2014) also did a meta-analysis with 107 effect sizes and 14,321 extracted participants. They compared the effectiveness of ITSs for different outcomes (e.g., declarative and procedural knowledge) and different non-ITS learning environments (e.g., traditional classroom instruction, individual computer instruction). The studies reviewed ITSs used in different ways. Typically, they were used to provide students feedback on their work, but they were also sometimes used as the primary means of instruction. The authors found using an ITS lead to greater achievement than large-group teacher-led instruction (g = 0.42), non-ITS computer-based instruction (g = 0.57) and workbooks (g = 0.35). They found there was not a significant difference between ITS use and one-to-one tutoring (g = .11) or small-group instruction (0.42). They also found that regardless of the way an ITS was used (e.g., principal means of instruction, an aid to instruction), there was a positive average effect size over not using an ITS (Ma et al., 2014).

2.3.1.2 Adaptive Hypermedia Systems.

AHSs combine hypermedia and ITSs to produce systems whose content, links, and other features dynamically adapt to learners based on learner characteristics (Papadimitriou & Gyftodimos, 2017). Wauters et al. (2010) broadly described how ALSs can adapt through how content is presented, providing the learner hints and support as needed, and changing the sequence of material. AHSs can adapt in these ways but can also adapt through navigation and meta-adaptive navigation (Papadimitriou & Gyftodimos, 2017). Navigation refers to adapting link structures available to the learner to guide them towards relevant material (Papadimitriou & Gyftodimos, 2017). An example of this would be hiding links to material from the learner based on their performance in the system. Meta-adaptive navigation refers to adapting the navigation technique for the learner based on what likely suits them best given the context. Examples of navigation techniques are the mentioned

approach of hiding links, as well as sorting links the learner can see or offering direct suggestions (Papadimitriou & Gyftodimos, 2017).

There have been three generations of AHSs over time: first, second, and third. The distinction between generations is based on when they were developed and how the system is deployed (Carver, et al., 1999). First generation AHSs were not distributed, meaning a system existed entirely on a single computer (not across a series of networked computers). These systems had limited adaptability based on stereotype models of learners (Böcker, et al., 1990; Boyle & Teh, 1993; Brusilovsky, 1992). The second generation was deployed through the Internet. Adaptation in these systems was more sophisticated (e.g., more forms of media available) and learners were modeled with more characteristics (Brusilovsky & Eklund, 1998; De Bra & Calvi, 1998). The third generation further increased the number of characteristics learners could be modeled with and provided a finer grain of multimedia adaptation (Colace et al., 2014).

In their review of 40 articles, Papadimitriou and Gyftodimos (2017), found that, for AHSs, the most used learner characteristic in learner modeling is knowledge level, closely followed by learning or cognitive style. Knowledge level was used in 27 out of the 40 articles reviewed. Cognitive or learning style was used in 24 out of the 40 articles. They also found for AHSs that the most common form of adaption is adaptive navigation. Out of the 40 articles reviewed, 33 systems used this form of adaptation. They note that many systems used a combination of different forms of adaptation (e.g., adaptive navigation with adaptive curriculum sequencing).

2.4 How has Item Response Theory been used with adaptive learning systems?

2.4.1 What is item response theory?

Item Response Theory (IRT) is a statistical framework in which examinees can be characterized by one or more traits (e.g., their proficiency in a domain). Scores for these traits are estimated, through mathematical models, using examinees' observed performance on test items.

Similarly, one or more traits of items are estimated through a group of examinees' performance on the items. The person and central item trait (the item difficulty) are located on the same mathematical scale (de Ayala, 2009). The simplest IRT model is the Rasch model (Rasch, 1960), which estimates the probability of an examinee getting an item correct (i.e., endorsing the item) based on the distance between the person's score on the trait measured by the item and the item difficulty on their shared scale. The following equation shows the Rasch model. The equation shows the probability of examinee *i* endorsing item *j* given the proficiency of the examinee and the difficulty of the item.

$$p(x_j = 1|\theta_i, b_j) = \frac{e^{(\theta_i - b_j)}}{1 + e^{(\theta_i - b_j)}} \quad (1)$$

Where:

$p(x_j = 1|\theta_i, b_j)$ is the probability of student *i* endorsing item *j* given the proficiency of the student (expressed as a logit value) and the difficulty of the item (also expressed as a logit value).

θ_i is the proficiency of student *i*.

b_j is the difficulty of item *j*.

IRT has been used with adaptive learning systems over the last two decades. Unlike assessments, in a learning environment, learner proficiency is expected to change over tasks; thus, applications of IRT with ALS are typically towards measuring changes in learners over time. Early applications typically used progress testing for this (i.e., frequent, relatively short tests). More complex modeling that directly models growth is another approach researchers have considered. In the last decade, more attention has been paid to combining IRT with algorithms to track learner proficiency.

2.4.2 Progress Testing

Progress testing is frequently administering tests to quickly detect changes in proficiency and possibly intervene with learners (Wauters et al., 2010). Many applications of IRT with ALSs involve

progress testing. Chen et al. (2004) developed a courseware recommendation system that used a combination of IRT and fuzzy logic to recommend courseware to learners. After completing a unit of learning material, a learner reported the percent of material they understood and rated the difficulty of the unit. Their reported degree of understanding was used in a modified proficiency estimation approach using the Rasch model to update their proficiency estimate. A learner was then recommended the unit in the system with a difficulty parameter closest to the learner's updated proficiency.

Chen and Hsieh (2005) developed a similar system to Chen et al. (2004). Rather than report a percentage of understanding, at the end of a unit of material, learners reported a "yes" or "no" for if they understood the material. They also rated the difficulty of course material on a Likert-type scale from "very hard" to "very easy." In a similar way to Chen et al. (2004), the self-reporting in a distinct testing section at the end of course material lends this to be classified as progress testing. Chen et al. (2006) developed a learning system that considered courseware difficulty, learner proficiency, and the continuity of the concept being studied in recommending course material. IRT was used similarly to Chen et al. (2004) and Chen and Hsieh (2005) in a self-reporting section at the end of units of course material.

Leung and Li (2007) created an adaptive system for a data management course offered in the e-school of a University in Japan. In their system, material was divided into compulsory and optional material. Pretests and posttests with compulsory material were used to decide if a student should receive optional material.

Chen and Hsu (2008) applied fuzzy item response theory in a mobile system to recommend English news articles to learners based on their vocabulary ability. Like other work by Chen, after a learner finished a news article, they were asked to self-report their understanding (from 0 to 100%) and how difficult they thought the article was to read (easy, moderate, or hard). Like Chen et al.

(2004), a learner's reported degree of understanding was used with the Rasch model to update their proficiency estimate. The system then recommended new English news articles to the learner based on the learner's updated proficiency and the difficulty estimates of the news articles. The recommended articles were those with difficulty values closest to the learner's updated proficiency estimate. Chen and Chung (2008) similarly applied IRT in a system for learning English vocabulary; however, their application did not involve self-reporting. Their system was also more focused on direct vocabulary review rather than reading news articles. Learners took 10-item quizzes at the end of a period of learning new words. A learner's proficiency estimate was updated using their responses to quiz items and the Rasch model. The learner was then recommended new vocabulary words to learn based on their updated proficiency estimate and the difficulty values of the words. The words recommended to the learner were those with difficulty values closest to the updated proficiency estimate of the learner.

Baylari and Montazer (2009) applied IRT with an artificial neural network in their proposed system. Their system involved 10-item pretests, posttests, and review tests interspersed among units of learning material. Items were modeled using the 3-parameter model. The artificial neural network took as input item characteristics and responses. It used them to diagnose learner problems and then recommended appropriate learning material. Yarandi et. al (2012) presented an ontology-based system that helped learners select appropriate learning paths within the system. Similar to other applications, they used IRT in discrete tests within the system. After learners completed units of learning material, they took a test, and the IRT-based results of that test were used by the system to guide learners' choices of what material to select next. Huang and Shiu (2012) proposed a user-centric adaptive learning system that used sequential pattern mining to build learning paths for learners. Their system used IRT in a pretest, posttest set up. Learners took pre-tests that identified concepts they were unfamiliar with. The system built a learning path for each learner based on their tests results. Once the learner finished their learning path, they were tested again.

Hosseini et al. (2013) implemented an ontology-based adaptive learning system that used several learner characteristics to model the learner (proficiency estimates from IRT, learning styles, preferences, and prior knowledge). After a learner finished a topic, they took a test, and their proficiency was estimated using the 3-parameter model. This estimate was used to move the student forward or to suggest an alternative learning path, such as reviewing pre-requisite material. The other characteristics of the learner, such as learning style, were used to adapt the presentation and navigation support with material.

An issue with progress testing is that it is generally considered intrusive; this is important as it may distract or demotivate learners (Abyaa et al., 2019). The broad trend in ALSs is towards less direct forms of updating learner characteristics. In their review of 107 articles from 2013–2017, Abyaa et al. (2019) found 76% of studies used implicit measurement over explicit forms. Progress testing is an explicit form of measurement; thus, this is not an ideal long-term solution for making IRT-based learning systems more adaptive.

2.4.3 Modeling Change

Along with progress testing, modeling growth directly has been another approach to using IRT in learning contexts. In the context of student modeling in ITSs, Pelánek and Jarušek (2015) proposed a model estimating the time it takes to successfully complete a problem based on learner proficiency and item-related parameters. They also incorporated learning into the model through modeling growth with a learning curve. In their proposed model, an increase to a student's proficiency is calculated by multiplying the individual student's learning rate by the logarithm of the serial position of an item or problem (e.g., if the at-hand item is the 50^{th} the student has seen, then log(50)). The following is their model.

$$t_s p = b_p + a_p\left(\theta_s + \delta_s \cdot \log\left(k_{sp}\right)\right) + \epsilon$$

(2)

Where:

$t_s p$ is the time it takes student *s* to successfully complete problem *p.*

b_p is the difficulty parameter for item *p.*

a_p is the discrimination parameter for item *p.*

θ_s is the learner's proficiency at the start of their session.

δ_s is the learning rate for student *s* (i.e., how much they learn per item).

k_{sp} is the position of problem *p* in the sequence of items student *s* has seen.

ϵ is error.

Abbakumov et al. (2019) proposed two dynamic extensions of the Rasch model for massive open online courses. They apply their model to data from three Coursera courses on economics, neuroeconomics, and game theory. These are video-lecture based courses where learners are allowed multiple attempts at items with hints and feedback between attempts. While these types of courses are not adaptive learning systems, this is still an application of IRT to an e-learning context that addresses changes to learners' proficiency.

The following is their most granular model. Their other model is the same but does not allow for variation of the effects of video and attempt across individual students. This model is based on a reformulation of the Rasch model by Van den Noortgate et al. (2003) and follows the approach to modeling individual proficiency dynamics presented by Verguts and De Boeck (2000), De Boek et al. (2011) and Kadengye et al. (2014,2015).

$$Logit\left(\pi_{ij}\right) = b_0 + (b_{10} + b_{1i}) * video_{ij} + (b_{20} + b_{2i}) * attempt_{ij} + u_{1i} + u_{2j}$$

(3)

Where:

π_{ij} is the probability of learner *i* endorsing item *j.*

b_0 is the logit probability of a correct response by an average student on an average item.

$video_{ij}$ is the progressive sum of video lectures that student *i* watched before responding to item *j*.

b_{10} is the overall effect of the progressive sum of video lectures.

b_{1i} is the individual deviation of student *i* from the overall effect of the progressive sum of video lectures.

$attempt_{ij}$ takes on values of 0, 1, 2, 3 or 4–representing the first (0) and higher (1 and up) attempts student *i* took to endorse item *j*. More than 4 attempts is represented as 4.

b_{20} represents the overall effect of an extra attempt.

b_{2i} represents the deviation of learner *i* from the overall effect of attempt.

u_{1i} is the deviation from the intercept for individual student *i*.

u_{2j} is the deviation from the intercept for individual item *j*.

While not directly modeling growth, Galvez et al. (2016) addressed changing learner proficiency without progress testing. Galvez et al. (2016) applied IRT to a constraint-based modeling system. Instead of modeling a problem as a single item, they modeled each constraint as an item (in constraint-based modeling, each problem is made up of a set of relevant constraints that must be satisfied by the learner's solution). Each constraint had its own equivalent to the item characteristic curve, which they referred to as the constraint characteristic curve. An assumption of calibrating item parameters is that an examinee's proficiency remains constant over the entire assessment. Galvez et al. (2016) proposed three methods for addressing this in their context: constant knowledge session, first time relevant, and problem grouping. All these approaches work by manipulating learner response data.

In the constant knowledge session approach, we assume that a learner's knowledge is constant for some amount of time. If consecutive sessions are close enough in time, they are grouped. The same student, represented across groups, where in each group their knowledge is assumed to be different, is, instead, considered a set of separate virtual students with different proficiencies.

The first-time relevant approach is when the response of a learner to a constraint is only considered if it is the first time the learner has encountered the constraint. A given constraint can be relevant across multiple problems. The second time a learner responds to a problem calling for the given constraint, it is possible that some relevant learning happened from their first encounter with the constraint. The problem grouping approach involves counting all attempts at a single problem as one session, meaning each problem is a single session. This approach assumes learning happens between problems (i.e., at the successful completion of a problem).

There are some disadvantages to using growth modeling approaches in ALSs. One issue is that these approaches often require large samples for parameter calibration and involve computationally intensive estimation techniques (Papoušek et al., 2014; Pelánek, 2016). This issue applies to other, non-growth IRT-based approaches such as Bayesian Knowledge Tracing. Another issue is that, by applying a specific growth model, an assumption is made about how learners will grow over time. There may be large individual differences between learners in how they develop over time (Bolsinova et al., 2022). There may also be complex feedback loops based on performance in a system, such as teacher intervention, that may change the trajectory of proficiency for learners (Bolsinova et al., 2022). These limitations make growth modeling a non-ideal choice for large e-learning systems where many estimations need to be made quickly.

2.4.4 Tracking

A third approach to measuring changes in learner proficiency with IRT is tracking. Tracking is using algorithms to trace parameters as they develop over time, rather than model them (Brinkhuis & Maris, 2010). With tracking, parameter estimates happen on-the-fly as learners do tasks in the learning environment.

2.4.4.1 Elo Rating System

A popular tracking system that has been applied in e-learning contexts is the Elo rating system (Elo, 1978; ERS). This system was developed by Arpad Elo in 1960 and is intended for dynamic ratings in contexts where there are large amounts of paired comparison data. Paired comparison data are data representing matches between two players or teams (Brinkhuis et al., 2018). There are several different forms of the ERS. The following is the Current Rating Formula for Continuous Measurement (Elo, 1978, p. 25)

$$R_n = R_o + K(W - W_e) \tag{4}$$

Where:

R_n is the new rating after the event.

R_o is the pre-event rating.

K is the rating point value of a single game score.

W is the actual game score, each win counting 1, each draw 1/2.

W_e is the expected game score based on R_o.

In a match between two players, each player has a rating, R_o , before the match. This rating is updated to R_n based on the weighted difference between the observed outcome of the match W and the expected outcome of the match W_e. The expected match outcome is based on the players' ratings before the match begins. W can be 1, ½, or 0, representing a win, a tie, or a loss. The difference between the actual and expected match outcome is weighted by K. K is, thus, a control for how much ratings can change between matches.

The expected match outcome is calculated using the Bradley Terry Luce (BTL) model, which is closely related to the Rasch model in IRT. In the BTL, two people oppose each other, while in the Rasch model, a person opposes an item (Rasch, 1960; Bradley & Terry, 1952). Klinkenberg et al.

(2011) introduced the ERS to e-learning. They adapted the ERS such that the update for the second person in the match is replaced with an update for the item. The following is their adaptation.

$$\widehat{\theta_i} = \theta_i + K\big(S_i - E(S_i)\big)$$

$$\widehat{\theta_j} = \theta_j + K\left(S_j - E\big(S_j\big)\right) \tag{5}$$

Where:

$\widehat{\theta_i}$ is the estimated rating for learner *i* after the match.

θ_i is the estimated rating for learner *i* before the match.

K is a weight controlling how much the difference between the observed and expected match outcome can update estimated ratings.

S_i is the observed outcome of the match for learner *i* (1 if the learner endorsed the item and 0 if they did not).

$E(S_i)$ is the expected outcome of the match for learner *i* based on the ratings of the learner and item at the start of the match. This is expressed as the probability of the learner endorsing the item and comes from the Rasch model.

$\widehat{\theta_j}$ is the estimated rating for item *j* after the match.

θ_j is the estimated rating for item *j* before the match.

S_j is the observed outcome of the match for item *j* (1 if the learner did not endorse the item and 0 if they did).

$E\big(S_j\big)$ is the expected outcome of the match for item *j* based on the ratings of the learner and item at the start of the match. This is expressed as the probability of the learner not endorsing the item and comes from the Rasch model.

There are several desirable properties to the ERS. First, it is self-correcting (Brinkhuis et al., 2018). For example, if the actual outcome of the match was the learner endorsed the item, and it was estimated that they only had a 20% probability of endorsing the item, then they will have a relatively large increase to their proficiency estimate (compared to if they had a 70% estimated probability of endorsing the item). If this was a moderate overestimate, then in a later match the learner may not endorse an item when they had a 60% estimated probability of endorsing the item. Their proficiency

estimate would then be lowered by the system closer to its true value. Another benefit of the ERS is that it is computationally light (Brinkhuis et al., 2018). The ERS can easily update parameter estimates in real-time on very large data. A similar scaling up of more computationally intense methods for real-time updates would be more challenging (Brinkhuis et al., 2018; Pelánek et al., 2017; Reddick, 2019).

A problem with the ERS is that it suffers from rating inflation and deflation (Glickman, 1999). This can also be called scale drift (Klinkenberg et al, 2011). One source of drift is from when learners enter and leave a system. When a learner leaves a system, they tend to have higher ratings than when they entered the system. This causes a downward drift in item ratings, which in turn lowers person ratings (Klinkenberg et al, 2011; Reddick, 2019). Another source of drift, inflation specifically, is adaptive matchmaking. This is when a system matches learners to items based on learner and item ratings (Hofman et al., 2020). A consequence of this is that ratings at different time points cannot be compared, meaning we cannot measure growth (Klinkenberg et al., 2011).

There have been several applications and evaluations of the ERS in e-learning contexts. Klinkenberg et al. (2011), when they introduced ERS to e-learning, applied it to Math Garden. Math Garden is a web-based environment where learners can practice addition, subtraction, multiplication, and division. They used an extension of the Rasch model with the ERS that considered response time. They also used a function to adjust the weight, *K*, in the ERS based on recency and frequency. If a learner had not been in many matches, and there was a large lapse in time between the last match involving the learner and the current match involving the learner, then there was more uncertainty around the rating of the learner. The function controlling *K* accounted for instances of this by increasing the weight. This allowed for more changes in learner rating estimates between matches. If a learner had been in many matches and was in a match recently, then there was less uncertainty around their rating estimate. The function controlling for *K* accounted for this by decreasing the

weight. This restricted the amount the learner rating could change between matches. The same applied to items.

The authors evaluated how well the ERS estimated learner proficiency by seeing how estimates correlated with learner scores from the pupil monitoring system of the National Institute for Educational Measurement (CITO). The correlations between the ERS and CITO ranged from 0.78 to 0.84 for the four domains practiced in Math Gardens. As context, the correlation of mid-year and end-of-year CITO scores from the 2007 to 2008 academic year was 0.95. They also evaluated item difficulty estimates by assessing their reliability. They correlated item difficulties from week 44 of the study, which they considered established, to all item ratings in subsequent weeks. Across 32 weeks, the correlation stayed above 0.95.

Wauters et al. (2011) compared six item parameter estimation techniques to conventional IRT-based calibration: learner feedback, proportion correct, expert rating, paired comparison among learners, paired comparison with an expert, and the ERS. Using learner feedback to estimate item parameters involves asking learners two questions: “Do you understand the content of the course material?” and “How do you think about the difficulty of the course material?” Learners respond to the second question using a Likert-type scale, and their ratings are averaged. The difficulty level of course material is the weighted linear combination of the difficulty from the averaged learner ratings and expert ratings. Proportion correct is the number of learners who responded to the item correctly over the total number of learners who responded to the item. Paired comparison refers to taking an item with an unknown difficulty and locating it in a series of 11 items that are ordered by their difficulty. The item is located using human judgment.

The data for the study were items on French verb conjugation. Participants were students and French teachers from the Flemish region of Belgium. These methods were evaluated by the correlation of difficulty estimates with the true difficulties of items. The ERS achieved a correlation

of 0.85. For context, conventional IRT-calibration, proportion correct, and learner feedback achieved correlations of 0.90, 0.90, and 0.88 respectively. The correlations for the other methods were lower than the ERS.

Papoušek et al. (2014) applied the ERS to a system for learning geography facts, specifically the names of countries (slepemapy.cz). In the system, learners responded to a series of questions ("Where is country X?", "What is the name of country X?"). They responded to questions using an interactive map, and learners received feedback on their responses. The authors used both the Rasch model and the ERS to estimate item difficulties. They found that the results from joint maximum likelihood estimation, a conventional approach for estimating item parameters with IRT, and the estimates from the ERS were nearly identical (a correlation of 0.97).

Nižnan et al. (2015) applied the ERS to the same online system for learning geography facts (slepemapy.cz) to model learner prior knowledge. Their use of the system covered the names and locations of countries, but also included such geographic structures as cities and mountains. They compared several extensions of the Rasch model (e.g., Bayesian, hierarchical, networked) including the ERS. The ERS used the Rasch model to get expected match outcomes. Models were evaluated by the discrepancy between their predicted probability of a learner endorsing an item and the observed response of the learner. Across metrics—root-mean-square error, log-likelihood, and area under the ROC curve—the ERS was very close in performance to the other models, which were themselves very close in performance.

Pelánek (2016) compared conventional IRT item parameter estimation, the ERS, and proportion correct using simulated data. The simulated data had learners responding to items across degrees of adaptive matching between items and learners. At one extreme, items were matched to learners completely at random. At the other extreme, items were matched to learners based on what item had a difficulty closest to the learner's proficiency. Learners' proficiencies were made constant

across the items they responded to. The author found that when items were selected randomly, joint maximum likelihood estimation, the ERS, and proportion correct performed similarly, achieving correlations between the estimates and true item parameters around 0.90. As matching became less random, proportion correct performed worse, while the estimates from the ERS and joint maximum likelihood estimation were very similar. They also found that once the ERS had stable item difficulty estimates, it could estimate learner proficiency quickly. After 10 items, the ERS was able to achieve a 0.80 correlation of proficiency estimates with their true values.

Pelánek et al. (2017) built on the work of Papoušek et al. (2014) in their application of the ERS to a system for learning geography facts. Pelánek et al. (2017) applied the ERS to the system outlinemaps.org. This site let learners choose specific maps and types of places (e.g., a map of Africa, focusing on rivers). Learners were asked two types of questions: the first about the location of places (e.g., "Where is France?") and the second about the name of places ("What is the name of the highlighted country?"). Learners responded to questions through working with an interactive map or through selecting from multiple response options. The authors compared several extensions of the Rasch model (e.g., Bayesian, hierarchical, networked) and the ERS model to see how well they estimated learner prior knowledge. They found that all approaches, including the ERS performed similarly across metrics (root mean square error, log-likelihood, and area under the ROC). They also found that joint maximum likelihood estimation of item parameters and ERS estimates were very similar (a correlation of 0.97).

Park et al. (2019) proposed a multidimensional extension of the ERS in which multiple learner proficiencies are updated after each item. The authors conducted a simulation study using two item banks, one in which items loaded on a single latent dimension and another in which items were allowed to load on multiple dimensions. They had 250 simulated learners each with three latent traits

being measured. They also had conditions in which the latent traits were not correlated, weakly correlated, and moderately correlated. This resulted in 6 sets of data.

They evaluated how well their multidimensional ERS estimated the latent traits using the mean square error of estimated and true learner proficiency values at different numbers of item responded to. They also compared ERS learner proficiency estimates to expected a posteriori estimates from a fitted compensatory IRT model. In the first evaluation, they found that mean square error decreased as the number of items increased across all 6 data sets, though the type of item bank used moderated this effect. More revealing, they found that the learner proficiency estimates from the ERS, after 200 item responses, were highly correlated with expected a posteriori estimates across all data sets. The correlations ranged from 0.97 to 0.99.

2.4.4.2 Glicko.

Glicko is a Bayesian generalization of the ERS developed by Glickman (1999). Like the original purpose of the ERS, Glicko was developed for working with large paired-comparison data, such as in chess player rankings (Glickman, 1999). Glicko builds on the ERS by incorporating the variability of parameter estimates (Glickman, 1999). The ERS provides point estimates of parameters but does not give a measure of the uncertainty around those estimates (Goldowsky, 2006). The Glicko system does provide a measure of estimate uncertainty.

As explained by Glickman (1999, 2022), the algorithm works by conceptualizing player ratings as probability distributions rather than only point estimates. For a player, we initialize them with a prior distribution for their rating. The distribution is normal with a known mean and variance (decided on rather than estimated from data). The player then enters a rating period of a certain number of matches. Glickman recommends 5–10 matches. The player's rating/proficiency is assumed to be constant across the rating period. At the end of a rating period, the player's prior distribution is updated with the results from the matches in the rating period. This updated distribution then becomes

the prior for the next rating period (Glickman, 1999). The time elapsed within rating periods is taken into consideration. As the rating periods get longer, it is assumed that accuracy of the rating estimate becomes less certain, so the variance of the distribution modeling a player's rating increases. Uncertainty around an estimate can be represented as a 95% confidence interval, formed by adding and subtracting two standard deviations of the player rating distribution from its mean.

In 2001, Glickman (2001), developed the Glicko-2 system, which incorporates stochastic volatility into the system. This took the form of a rating volatility parameter, which indicates the expected fluctuation in a player's performance. This parameter has a high value when the player has unexpected performances based on their past performance (e.g., unexpectedly winning against several other players with higher ratings) and a low value when they are playing consistently (Glickman, 2022). A greater rating volatility is considered in calculating the variance of a player's rating distribution to reflect the greater uncertainty around the player's true rating (Goldowsky, 2006). In the Glicko-2 system, a parameter is set using human judgment to constrain how much the rating volatility parameter can change over time.

There have been two applications of the Glicko system to an e-learning context. Reddick (2019) applied the Glicko system to data from the online learning platform Coursera to estimate learner skill proficiency and item difficulty. They counted exams with multiple-choice or text answers, programming assignments, and peer review assignments as items. Their data came from courses on business, computer science, and data science. Learners could retake many items. The authors only used data from learners' first attempts at items or first and later attempts when the later attempt had a different outcome from the first attempt (e.g., they did not get the item correct on the first attempt but did get the item correct on the second attempt). The authors adapted the Glicko system to have a rating period of one "match" long, meaning learner estimates were updated after each item.

The results were evaluated based on if results made sense with the design of courses. The authors found that difficulty estimates for programming assignments tended to be more difficult than exams, which is what they expected. The authors also found that within courses, later items in the course tended to have higher difficulty estimates than items earlier in the course. Specifically, across skills, they found a 0.25 correlation between item order within a course and item difficulty estimates. The authors also found that courses had median item difficulties that made sense. For example, courses considered fundamental tended to have lower median item difficulties than more advanced courses located towards the end of series of courses on a particular topic.

Park (2021) also applied the Glicko system, specifically the Glicko-2 system, to an e-learning context. Their data was from K–12 math learning software in which students were given problems to practice based on their grade and content areas they struggled with. Their data was stratified, meaning learners only saw items within their grade strata. Learners were allowed multiple attempts at items. For their analysis, the author only considered the first attempt at the item. They drew data from January 2016 to December 2019 from learners in the United States.

Like Reddick (2019), Park (2021) adapted the Glicko-2 system to remove the need for rating periods. Their reformulation allowed the system to consider the time elapsed between the current match involving a learner or item and the last match involving that learner or item. As the elapsed time increased, the variability around the estimate for the learner or item increased.

To evaluate their application, they looked at the root-mean-square error between the system estimated outcome of matches and the observed outcomes. They found that as the system calibrated learner proficiency estimates, the root-mean-square error across the entire system decreased. They also looked at the distribution of item difficulties within curricular units. They showed that with graph-based rating initialization the distributions of item difficulties moved higher across sequential, increasingly difficult curricular units.

2.4.4.3 TrueSkill.

Like Glicko, TrueSkill is also a Bayesian generalization of the ERS. This system was developed by Herbrich et al. (2006) for multiplayer online game environments. There are no instances of TrueSkill being applied in an e-learning context. The authors developed TrueSkill to tackle two main issues in multiplayer online gaming environments: 1) game outcomes are often for teams of players, but skill ratings for individual players will be needed for future matchmaking. 2) Sometimes more than two teams compete, such that the game outcome is not a simple designation of a winner and a loser.

The algorithm involves four variables: the skill estimates of all players, the performances of all players, the performances of all teams, and the differences in performances across teams. Similar to Glicko, each player has a prior distribution representing their skill estimate. After a game, these skill distributions are updated using the performance of the individual players, the performances of their respective teams, and the difference in those team performances. These updated individual player skill distributions then become the priors for those players in subsequent games.

Minka et al. (2018) developed a second version of TrueSkill, TrueSkill 2. This iteration of the algorithm improved on the first in several ways. For example, TrueSkill 2 uses more information to estimate a player's skill, such as their number of kills, death count, tendency to quit, and skill rating in other modes of the game. Another improvement was the random walk representing an individual player's skill rating was biased towards the skill increasing. In the original TrueSkill, it was equally probable that a player's skill rating increased or decreased.

Glicko suffers from the same scale drift issue as the Elo Rating System (Goldowsky, 2006; Hofman et al., 2020; Redick, 2019). While scale drift is not explicitly stated as an issue with TrueSkill, its focus on team-based gameplay does not map well to more individual-focused e-learning contexts. While Glicko and TrueSkill improve on the ERS by providing a measure of uncertainty

around skill estimates, they achieve this by making distributional assumptions of normality of skills (Bolsinova et al, 2022). This assumption may not be accurate to reality

2.4.4.4 Urnings.

Hofman et al. (2020) proposed a tracking algorithm that addresses the scale drift issue that concerns the ERS and extensions of the ERS (i.e., Glicko and possibly TrueSkill). Their proposed algorithm is called Urnings. Like the ERS, it is self-correcting, easily scalable to large paired-comparison data, capable of tracking dynamically changing parameters, and does not require a specified model for growth (Hofman et al., 2020). Like Glicko and TrueSkill, Urnings provides standard errors for parameter estimates, given that the parameter is stable for long enough (Hofman et al., 2020).

Urnings reconceptualizes a match between a learner and item as a game of chance involving urns. The urns are each filled with green and red marbles. The proportion of green marbles in the learner's urn represents their proficiency. The proportion of green marbles in the item urn represents its difficulty. In a match between a learner and item, a marble is drawn from the learner urn, and a marble is drawn from the item urn. If the marbles are the same color (e.g., both marbles are green), the marbles are returned to their respective urns, and another marble is drawn from each urn. This process goes until the two marbles are different colors. When the marble from the learner's urn is green and the marble from the item urn is red, the learner wins the match (i.e., the learner endorsed the item). When the marble from the learner's urn is red, and the marble from the item urn is green, the item won the match (i.e., the learner did not endorse the item).

The probability of the learner winning in this game of chance can be represented with a re-parameterization of the Bradley-Luce/Rasch model.

$$p(X_{ij} = 1|\pi_i, \pi_j) = \frac{\pi_i(1-\pi_j)}{\pi_i(1-\pi_j) + (1-\pi_i)\pi_j}$$

(6)

Where:

$$\pi_i = \frac{exp(\theta_i)}{1 + exp(\theta_i)}$$

(7)

π_i is the proportion of green marbles in the urn of learner *i* (representing their proficiency/rating estimate).

θ_i is the logit value for learner *i* (representing their proficiency/rating estimate on the logit scale).

$$\pi_j = \frac{exp(b_j)}{1 + exp(b_j)}$$

(8)

π_j is the proportion of green marbles in the urn of item *j* (representing its difficulty/rating estimate).

b_j is the item difficulty of item *j* (expressed on the logit scale)

For a learner, the proportion of green marbles in their urn will change over their time in the system as they learn. The proportion of green marbles in the urn of an item will change as the proportion converges on the true difficulty of the item or if the difficulty of the item drifts for some reason.

Updating the proportion of marbles in urns first involves setting an urn size for both learners and items. The urn size can differ for learners and items. The size of the urn controls how much learner or item proportions can change between matches. A larger urn size will lessen the change in proportion that adding or removing a single green marble causes, meaning the learner or item rating (represented by the proportion) will change less between matches. A smaller urn size will increase the change in proportion that adding or removing a single green marble causes, meaning the learner or item rating can change more between matches (Hofman et al., 2020).

Proposed updates to urns come from comparing observed match outcomes to expected match outcomes based on learner and item ratings. In the following, observed match outcomes are represented on the left and expected match outcomes are represented on the right.

Modeled reality:

repeat

$Y_i \sim Bernoulli(\pi_i)$

$Y_j \sim Bernoulli(\pi_j)$

until $Y_i \neq Y_j$

return (Y_i, Y_j)

Rating system:

repeat

$Y_i^* \sim Bernoulli(r_i/n_i)$

$Y_j^* \sim Bernoulli(r_j/n_j)$

until $Y_i^* \neq Y_j^*$

return (Y_i^*, Y_j^*)

Where:

r_i is the current number of green balls in the urn of learner *i.*

r_j is the current number of green balls in the urn of item *j*.

n_i is the urn size for leaner *i* (this is the same across all learners).

n_j is the urn size for item *j* (this is the same across all items and can vary from the urn size for learners).

Y_i is the observed outcome of the match for learner *i* (1 if they endorsed the item, 0 if they did not).

Y_j is the observed outcome of the match for item *j* (0 if the learner endorsed the item, 1 if they did not).

Y_i^* is the expected outcome of the match for learner *i* (expressed as 1 or 0 based on the probability of the learner endorsing the item given the current learner and item rating estimates).

Y_j^* is the expected outcome of the match for item *j* (expressed as a 1 or 0 based on the probability of the learner endorsing the item given the current learner and item rating estimates).

The following is the formula for the proposed update to the number of green balls in the learner urn and item urn.

$$\tilde{r}_i = r_i + Y_i - Y_i^*$$

$$\tilde{r}_j = r_j + Y_j - Y_j^*$$

(9)

Where:

$\tilde{r}_i$ is the proposed new number of green balls in the urn of learner *i* based on the difference between the observed and expected outcome of the match.

$\tilde{r}_j$ is the proposed new number of green balls in the urn of item *j* based on the difference between the observed and expected outcome of the match.

This update is only a proposed update. There is a final Metropolis-Hastings step that determines if the proposed change is accepted.

$$min\left(1, \frac{r_i(n - r_j) + (n - r_i)r_j}{\tilde{r}_i(n - \tilde{r}_j) + (n - \tilde{r}_i)\tilde{r}_j} \times \frac{p_\times(i, j|\tilde{\mathbf{r}})}{p_\times(i, j|\mathbf{r})}\right)$$

(10)

Where:

$\tilde{\mathbf{r}}$ is the vector of the proposed new number of green balls for the learner and the item urns.

$\mathbf{r}$ is the vector of the current number of green balls in the learner and item urns.

$p_\times(i, j|\tilde{\mathbf{r}})$ is the probability of learner *i* and item *j* matching based on the matching mechanism in the system and the proposed update to their rating estimates.

$p_\times(i, j|\mathbf{r})$ is the probability of learner *i* and item *j* matching based on the matching mechanism in the system and their current rating estimates.

If the proposed update makes future matches between player *i* and item *j* less likely, it is less likely to be accepted (Bolsinova et al., 2022). This component is only needed when the system is adaptively matching learners to items based on learner and item ratings. This step is a part of what prevents scale drift with the Urnings algorithm when there is adaptive matchmaking (Bolsinova et al., 2020). It also prevents a distortion of the invariant distribution for parameters we get when they are stable for long enough (Bolsinova et al., 2022).

Another part of anchoring the scale is choosing a core subset of items. Whenever one of these items loses or gains a green ball, a green ball is taken or given to another randomly chosen item in the

system. This keeps the number of green balls for this subset constant. The ratings of learners and items can then always be interpreted in relation to this core subset of items (Hofman et al., 2020). The following is the entire algorithm

Select players *i* and *j* according to $p_{\times}(i, j|\mathbf{r})$

Modeled reality:

repeat

$Y_i \sim Bernoulli(\pi_i)$

$Y_j \sim Bernoulli(\pi_j)$

until $Y_i \neq Y_j$

return (Y_i, Y_j)

Rating system:

repeat

$Y_i^* \sim Bernoulli(r_i/n_i)$

$Y_j^* \sim Bernoulli(r_j/n_j)$

until $Y_i^* \neq Y_j^*$

return (Y_i^*, Y_j^*)

Proposed update:

$$\tilde{r_i} = r_i + Y_i - Y_i^*$$

$$\tilde{r_j} = r_j + Y_j - Y_j^*$$

Metropolis-Hastings: accept $\tilde{\mathbf{r}}$ with probability:

$$min\left(1, \frac{r_i(n - r_j) + (n - r_i)r_j}{\tilde{r_i}(n - \tilde{r_j}) + (n - \tilde{r_i})\tilde{r_j}} \times \frac{p_{\times}(i, j|\tilde{\mathbf{r}})}{p_{\times}(i, j|\mathbf{r})}\right)$$

(Bolsinova et al., 2022, p. 99)

Urnings generates a Markov chain for the rating parameters of learners and items. Given the parameters are sufficiently stable over time, the invariant distributions of the Markov chains are binomial distributions with parameters π and *n*, with *n* being the total number of marbles in the urn. These distributions can be used to form confidence intervals around rating estimates as a measure of uncertainty. The following formula can be used to form 95% confidence intervals around rating estimates (Bolsinova et al., 2022).

$$\pi_i \pm 1.96\sqrt{\pi_i(1-\pi_i)}/\sqrt{n_i} \quad (11)$$

Here, we can see that the size of the confidence interval can be controlled through the urn size. Larger urn sizes will reduce the width of the confidence interval, while smaller urn sizes will increase it.

Hofman et al. (2020) and Bolsinova et al. (2022) evaluated Urnings with simulations and applications to real e-learning data. Hofman et al. (2020) simulated 500 learners responding to 100 items. Items were adaptively selected based on a learner's rating estimate such that the learner had a 50% estimated probability of endorsing the item. The urn size was 60 for learners and 200 for items. As starting values, all urns had a 0.50 proportion of green marbles. Both learner and item true ratings were held constant across all matches in the system, except for one learner to demonstrate how the algorithm recovered. The final ratings for learners had a 0.96 correlation with true values. The final ratings for items had a 0.98 correlation with true values. For the one learner with the non-constant true rating, their rating jumped suddenly at about halfway through their time in the system. After 93 matches, the rating estimate from Urnings fell within the 95% confidence interval of the true rating.

Bolsinova et al. (2022) simulated 1,000 players playing 100,000,000 matches. Unlike Hofman et al. (2020), this simulation did not make a distinction between learners and items among simulees. Player true ratings were held constant across matches. The correlation between true values and final Urnings rating estimates was 0.98.

Hofman et al. (2020) also applied Urnings to two games from the Math Garden system, an online e-learning system where children play games to practice different mathematical and cognitive skills. The first of the two games was a logical reasoning task, and the second was a subtraction task. Data for the logical task were collected from January 1, 2015 to June 3, 2019. Data for the subtraction task were collected from January 1, 2013 to June 30, 2017. For the logical task, this data consisted of

8,616 learners with 4,556,884 responses to 725 items. For the subtraction task, data consisted of 4,310 learners with 1,784,457 responses to 508 items. The authors set the urn size for learners to 30 and the urn size for items to 80. The authors found that for binned differences between learner and item ratings, represented in logits (e.g., the learner rating logit being 1 unit higher than the item rating) the Urnings algorithm predicted probability of the learner getting the item correct was very close to the observed proportion of correct responses for matches also in that bin (e.g., matches where the learner rating was 1 logit higher than the item rating).

Bolsinova et al. (2022) also applied the Urnings algorithm to Math Garden. They used data from 100 multiplication exercises. They followed a cohort of 14,175 learners over 3 years, gathering 1,696,112 responses. They used an urn size of 200 for items and 20 for leaners. Each player (learner or item) was given a random number of green marbles in their urn to start, ranging from 0 to the size of the urn for their group (either 200 or 20). Bolsinova et al. (2022) evaluated their application in the same way as Hofman et al. (2020) and achieved a similarly good match between expected and observed match outcomes.

The Urnings algorithm is similar to the ERS in some ways, such as being computationally light and self-correcting. In other ways, Urnings is different. Urnings provides unbiased estimates with a known error variance if there is no change in the rating estimate for some time (Bolsinova et al., 2022). The ERS provides no measure of uncertainty around rating estimates. Other systems, Glicko and TrueSkill, provide measures of uncertainty around estimates; however, these systems do this by approximating rating variance with an assumption of normality. The measure of uncertainty in Urnings is not an approximation, but a known invariant distribution.

Both the ERS and Glicko also suffer from variance inflation. The Urnings algorithm accounts for this, in part, by taking the adaptive matchmaking mechanism into account when considering proposed urn updates. This allows comparisons of learner and item ratings over time. This desirable

property comes at the price of discarding some proposed rating updates, but this is not problematic in a low-stakes e-learning context in which there are a lot of match data (Bolsinova et al., 2022). The desirable properties of the Urnings algorithm make it a promising solution for ALSs, especially those that already have an IRT-based assessment component. Proficiency estimates from Urnings could, for example, be used as starting values during interim assessments. The results of those assessments could then be handed back to the algorithm as learners continue working through learning material.

A gap in the literature is that the Urnings algorithm has not been evaluated using a criterion of learner rating estimates. The algorithm has been used in simulations and has been applied to e-learning data (specifically Math Garden). While results from these studies are promising, the algorithm estimates have not been compared to person estimates from conventional IRT approaches. Such a comparison would be valuable to industry researchers for assessing the accuracy of Urnings for large, complicated learning systems with IRT-based assessment components.

CHAPTER 3

METHOD

This section describes the research questions, sample, procedure, and evaluation of results in this study. For the sample, the source of the data and filtering process are described. Descriptive summaries of relevant characteristics of the sample are also provided. For the procedure, the steps for implementing the Urnings algorithm are described. Last, for the evaluation, the metrics and visualizations for evaluating and understanding the performance of the Urnings algorithm are detailed.

3.1 Research Questions

1. How well does the Urnings algorithm track the growth trajectory of students as measured by a conventional Item Response Theory (IRT)-based assessment?
    a. Where there is error between the Urnings algorithm and the IRT-based assessment, for what students does the error occur, when does the error occur (winter or spring), and what is the direction of the error?
    b. Is error associated with the number of items a student responded to, the amount the student grew, or the distribution of estimated b-parameters?

3.2 Sample

Mathematics data from the *i-Ready Diagnostic* (hereafter referred to as the Diagnostic) and the *i-Ready* platform were used for this study. The Diagnostic is a fixed-length, vertically scaled computer-adaptive interim assessment in reading and mathematics for students in kindergarten through high school (Curriculum Associates, 2017). The Diagnostic is intended to be taken three times a year—in fall, winter, and spring. The Diagnostic provides an overall subject score as well as scores for each of the domains within a subject. For the mathematics assessment, kindergarten through Grade 8 examinees responded to items for the following domains in the following sequence:

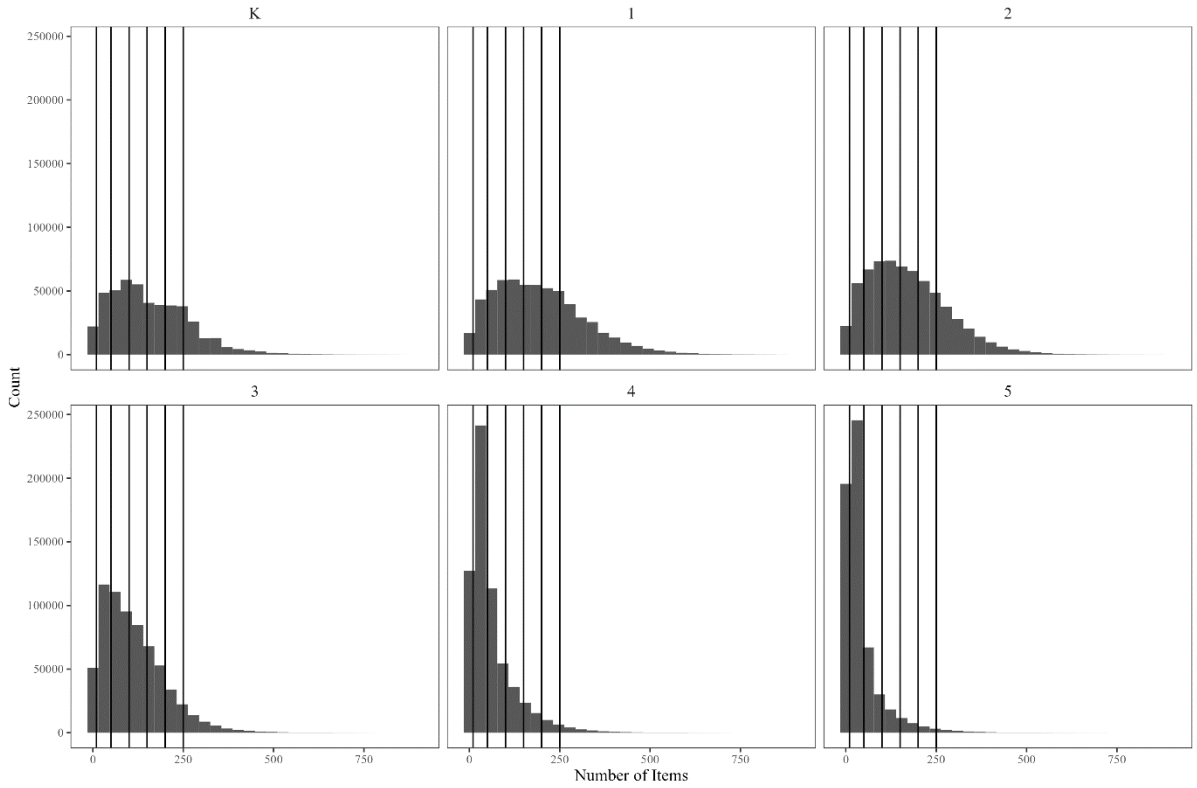
algebra and algebraic thinking, number and operations, geometry, and measurement and data. Examinees see 18 algebra and algebraic thinking items, 20 number and operations items, 14 geometry items, and 14 measurement and data items.

The domain-level performance from the Diagnostic is used to place students in the sequence of lessons within *i-Ready*. *i-Ready* is a digital learning platform for students in kindergarten through Grade 8 with a lesson sequence intended to support students' growth. Teachers may also assign specific lessons as desired. Each lesson within *i-Ready* is written for one of the domains assessed in the Diagnostic.

Students in the sample completed Diagnostics for the fall, winter, and spring in the 2021–2022 school year. Only students in kindergarten through Grade 5 were included in the sample. Items were taken from the quiz section of all system-assigned lessons a student completed. Lessons assigned to students by their teacher were not included. All quiz items in the sample were scored dichotomously. Students were divided into groups based on the number of items they responded to during the school year. These item groups were 10–49 items, 50–99 items, 100–149 items, 150–199 items, and 200–250 items. A minimum of 10 items was chosen to have enough data for the analysis method to work. These groups were chosen based on their coverage of the distributions of total items responded to by each student for the grades included in the study (See Figure 1). The lines in Figure 1 show how the item groupings cover these distributions. For Grades K–2, there is a noticeable portion of the distribution that responded to more than 250 items. This upper limit of 250 items was decided on as a compromise between adequately covering the distributions for Grades K–2 and having enough students in higher item groups for Grades 3–5 where there are fewer students who responded to more than 250 items.

Figure 2

Distribution of Total Items Responded to By Grade

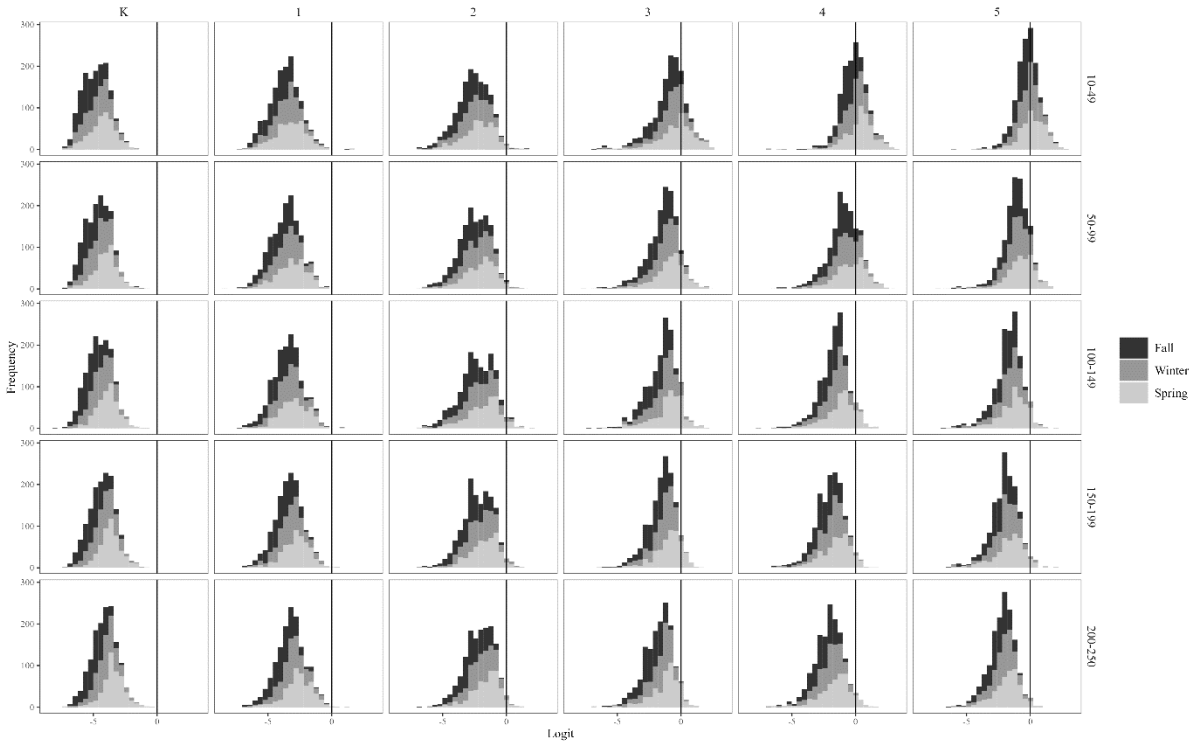


Note. The x-axis shows the number of items students responded to. The y-axis shows frequency. Each square of the plot represents a grade. The lines mark, from left to right, 10, 50, 100, 150, 200, and 250 items. Each square shows the frequency distribution of how many items students in that grade responded to.

From each grade and item grouping combination, 500 students were randomly sampled. As a result, the sample contains 2,500 students per grade and 15,000 unique students in total. Figure 2 shows the distribution of student Diagnostic scores for fall, winter, and spring by grade and item group. Table 1 in the Appendix shows means and standard deviations for student Diagnostic scores by season, grade, and item group.

Figure 3

Distribution of Diagnostic Scores for Fall, Winter, and Spring by Grade and Item Groups

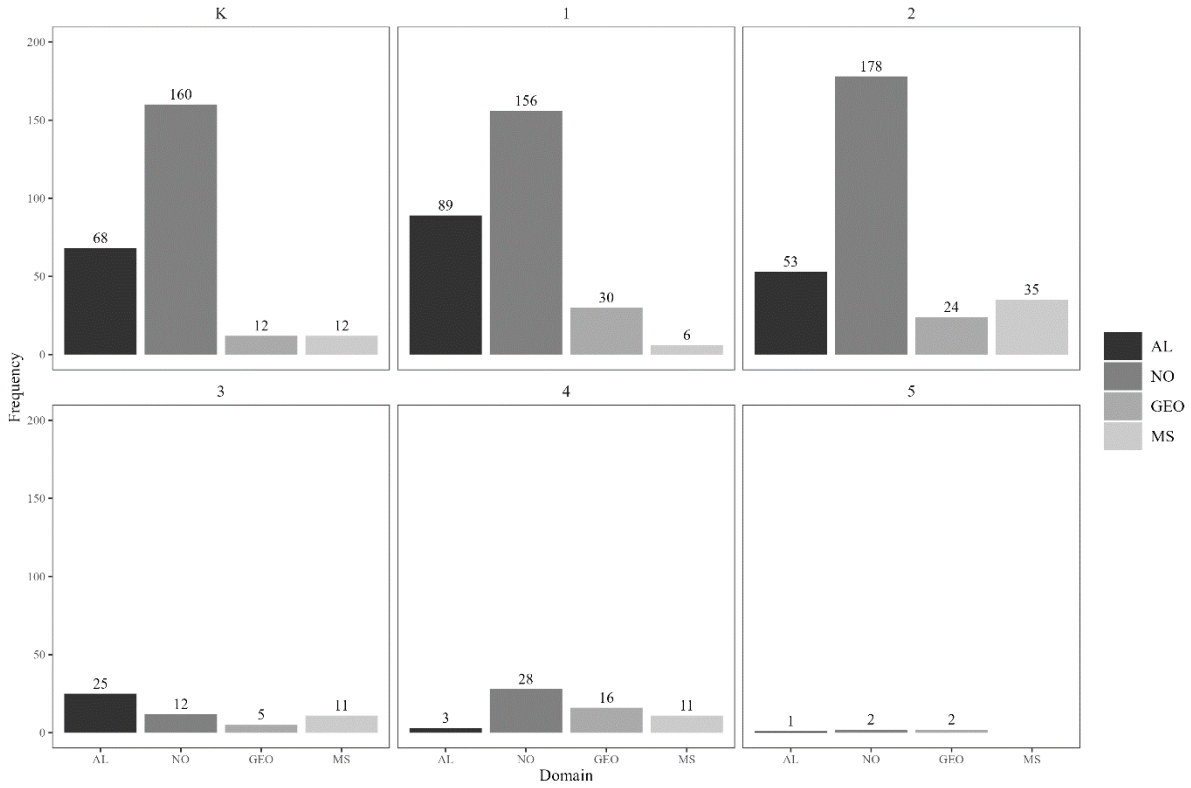


Note. The x-axis represents Diagnostic proficiency estimates in logits. The y-axis shows frequency. The columns of the plot represent student grade. The rows of the plot represent item groups (i.e., groupings by the number of items students responded to). Within a square, the darkest distribution is the frequency distribution of proficiency estimates from the fall Diagnostic. The medium gray distribution is for proficiency estimates from the winter Diagnostic. The lightest gray represents the spring Diagnostic. The line in each square marks a logit value of 0.

It can be seen in Figure 2 that, within grade, the score distributions are about the same across item groups for Grades K–2. For Grades 3–5, it appears that students who saw fewer items tended to have higher scores. This is most pronounced in Grade 5. The sample for this study contains 1,909,971 responses to items from students. Figure 3 shows the number of unique items in the sample by the domain and grade they were written for. Figure 4 shows, for each grade and item group, the distribution of responses to items across domains. Table 2 in the Appendix shows means and standard deviations for the number of items responded to by grade and item group.

Figure 4

Distribution of Unique Items across Grade and Domain Written for



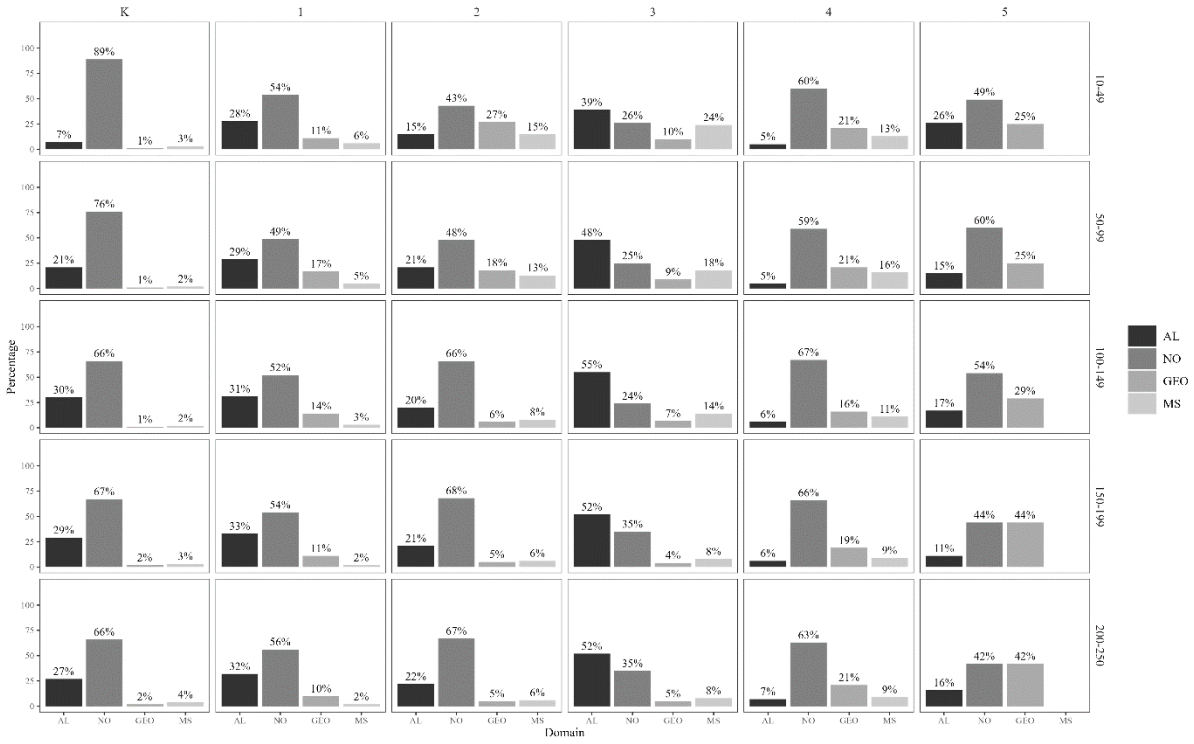
Note. The x-axis shows the domains of mathematics in *i-Ready.* The abbreviations are AL for Algebra and Algebraic Thinking, NO for Number and Operations, GEO for Geometry, and MS for Measurement and Data. The y-axis shows frequency. Each square in the plot represents a student grade. Within a square, the gray shade of a bar shows what domain it represents. The number above a bar shows the exact frequency for that grade and domain.

There were 927 unique items in the sample. It can be seen in Figure 4 that most items in the sample were written for Number and Operations and for kindergarten to Grade 2. Many items were also written for Algebra and Algebraic Thinking for these same grades. The lack of items written for Grades 3 and 4, and the almost absence of items written for Grade 5 is likely due to the placement of students. Students in higher grades can be routed to material intended for students in lower grades if they have a low score on a given domain in their Diagnostic. It is likely that many students in Grades 3, 4, and 5 placed in kindergarten, Grade 1, and Grade 2 lessons. It is also possible for students in lower grades to be routed to material written for higher grades if they score within a certain range in a

given domain; however, given the distribution of unique items in the sample, this does not seem to be prevalent.

Figure 5

Distribution of Responses to Items by Domain, Grade, and Item Group



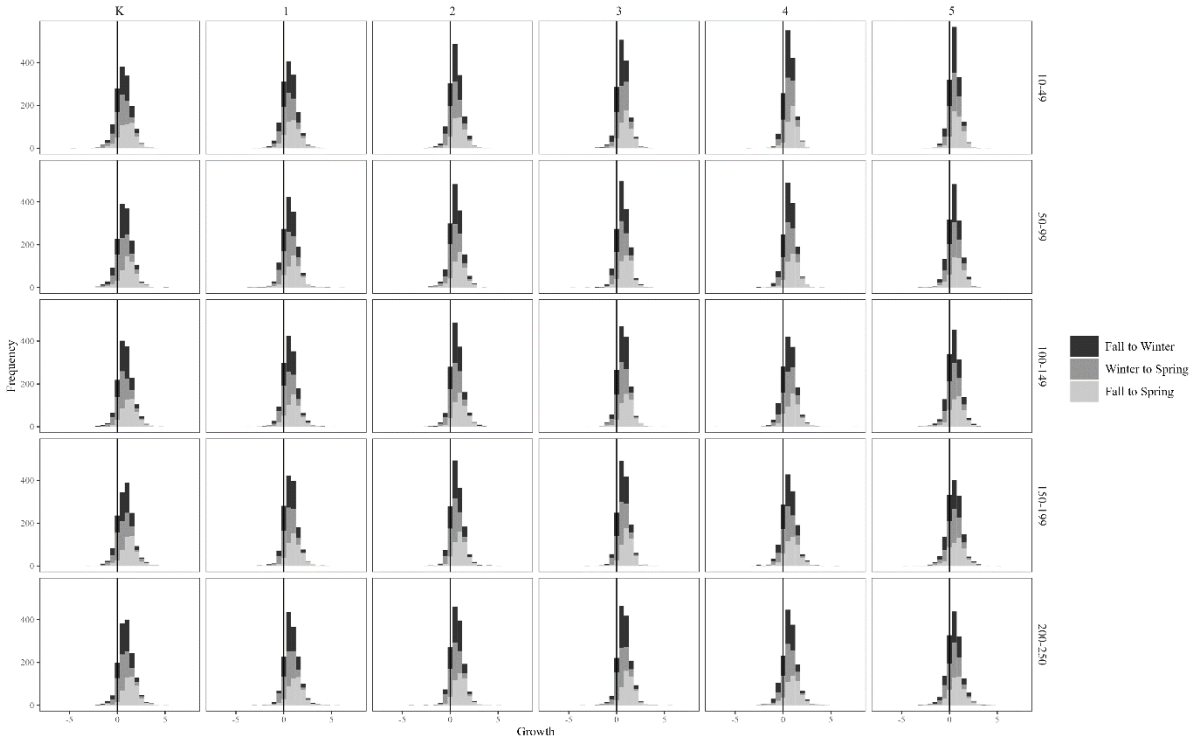
Note. The x-axis shows the domains of mathematics in *i-Ready*. The abbreviations are AL for Algebra and Algebraic Thinking, NO for Number and Operations, GEO for Geometry, and MS for Measurement and Data. The y-axis shows percentage. The columns of the plot represent student grade. The rows of the plot represent item groups (i.e., groupings by the number of items students responded to). Within a square, the shade of the bar shows the domain it represents. The value above each bar is the percentage of responses from the represented grade and item group that were for items belonging to the represented domain (e.g., for Grade K students in item group 10–49, 89% of all responses to items from this group were to Number and Operations items).

It can be seen in Figure 5 that most responses to items are for Number and Operation items, except for Grade 3, for which most responses were for Algebra and Algebraic Thinking. It also appears Grade 5 students did not respond to any Measurement and Data items. Figure 6 shows the distribution of growth from fall to winter, winter to spring, and spring to winter by grade and item group. It can be seen in the figure that both across grade and item group, the distribution of growth is

generally similar. It seems for Grades 4 and 5 that growth is less spread for smaller item groups than larger item groups. Table 3 in the Appendix shows means and standard deviations for growth by growth period, grade, and item group.

Figure 6

Distribution of Growth for Fall to Winter, Winter to Spring, and Fall to Spring by Grade and Item Groups

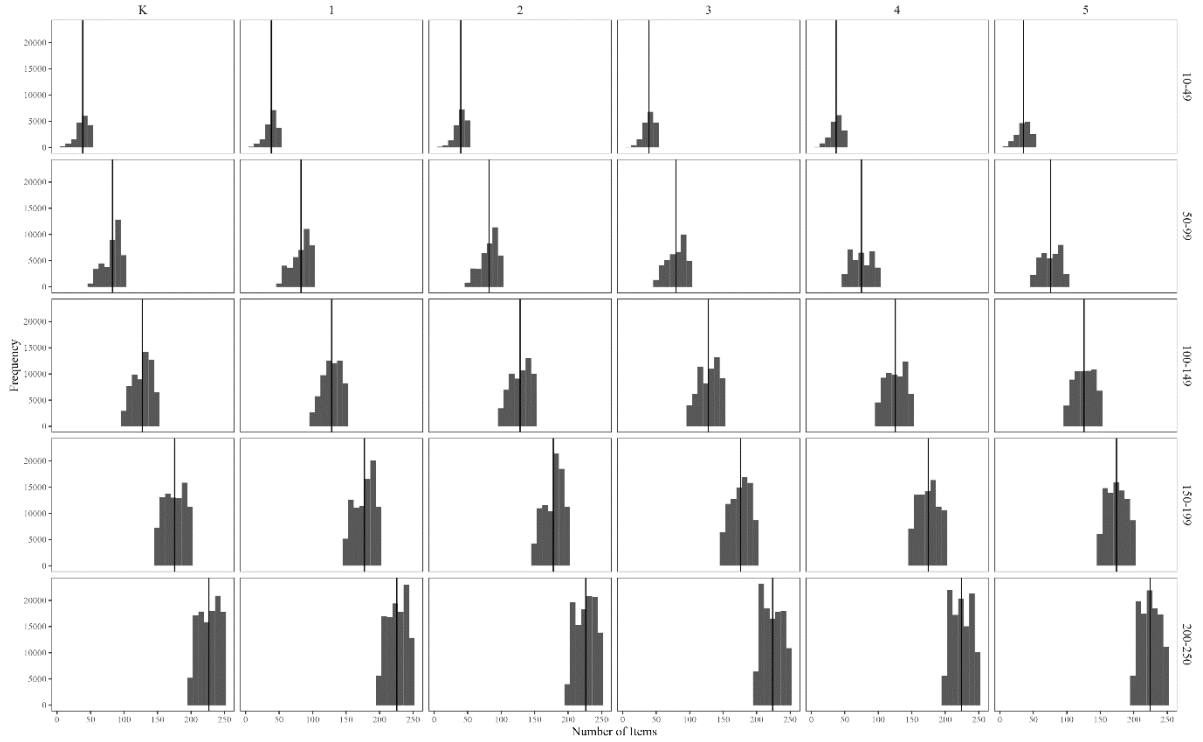


Note. The x-axis shows growth in logits. The y-axis shows frequency. The columns of the plot represent student grade. The rows of the plot represent item groups (i.e., groupings by the number of items students responded to). Within a square, the darkest distribution is the frequency distribution of growth from fall to winter (i.e., the winter Diagnostic estimate minus the fall Diagnostic estimate). The medium gray distribution is for growth from winter to spring. The lightest gray represents growth from fall to spring. The line in each square marks 0 growth.

Figure 7 shows the distribution of number of items responded to within item group. It can be seen in the figure that, across grade and item group, the distribution of items students responded to is relatively even about the group average. This indicates that there is not a skewed distribution of items responded to within any item group.

Figure 7

Distribution of Number of Items within Item Group by Grade and Item Group



Note. The x-axis shows the number of items students responded to. The y-axis shows frequency. The columns of the plot represent student grade. The rows of the plot represent item groups (i.e., groupings by the number of items students responded to). The lines mark the average number of items responded to for the represented grade and item group.

3.3 Procedure

The Urnings algorithm was applied to student response data, and proficiency estimates from the algorithm were compared to the proficiency estimates from student winter and spring Diagnostics to evaluate how well the Urnings algorithm tracked the growth of students over time. The algorithm was run under nine conditions. Each condition was a different combination of a person urn size and an item urn size with the urn sizes 50, 125, and 200 being possible for person and item urns.

The fall Diagnostic was used as the starting student proficiency estimate for the algorithm. All items were started with a number of green balls equal to half the urn size for the condition (e.g., 25 for a condition in which the item urn size is 50). For winter and spring, the algorithm estimate

closest in time to the completion of the Diagnostic was compared to the estimate from the Diagnostic. The algorithm estimate had to be within a week, either before or after, of the completion time of the Diagnostic for the comparison to be included in the analysis.

3.4 Evaluation Criteria

The performance of the Urnings algorithm was evaluated in three ways. For the first, root-mean-square error (RMSE) was calculated for each season (winter and spring), grade, and item group across the nine conditions. The following is the equation for RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(\widehat{\theta_i} - \theta_i\right)^2}{N}} \quad (12)$$

Where:

$\widehat{\theta_i}$ is the proficiency estimate from the Urnings algorithm for student i.

θ_i is the proficiency estimate from the Diagnostic for student i.

N is the total number of students.

RMSE provided a summary of how close algorithm estimates were to Diagnostic estimates across students. Comparing RMSE values across winter and spring can also possibly show how the performance of the algorithm differed with time. For the second way the algorithm was evaluated, the proportion of the sample that had their algorithm estimate within 1 and 2 standard errors of the Diagnostic estimate was calculated across season, grade, item group, and condition. This provided another way of evaluating how close the algorithm estimates were to the Diagnostic estimates, considering the error in the Diagnostic estimates.

The third way the algorithm was evaluated was by classification consistency. This was done for each season, grade, and item group across the nine conditions. Based on their Diagnostic scale scores, students were classified into grade- and within-grade-placement levels. The possible grade-level placements were kindergarten through Grade 12. The possible within-grade-placements were

early, mid, and late. There was an additional within-grade-placement level of emerging for kindergarten. Estimates from the Urnings algorithm were transformed using the linear transformation that transforms Diagnostic logit values to scale scores. Classifications were determined for the Diagnostic scale score and Urnings scale score using score ranges internal to Curriculum Associates. Classification consistency was calculated for both grade-level placements alone and for grade and within-grade placements together. Across organizations and systems, classifications are sometimes used instead of scores for operational decisions (e.g., if a learner can advance to new material). Classification consistency provides a way to assess the algorithm that allows a range of tolerable error. For example, while a point estimate comparison like RMSE may show error, if decisions about learners are made with classifications, and two methods classify the same, then that point estimate discrepancy may not matter in practice.

To better understand the direction of error detected through the RMSE, proportions, and classification consistency, the distribution of error was examined across season, grade, and condition. The spread of error around 0 provided a sense of if algorithm estimates tended to be positively or negatively biased. For example, an error distribution that largely sits above 0 indicates that the algorithm is consistently overestimating student proficiency.

To better understand possible causes of error both the correlation between growth and squared error and the correlation between number of items responded to and squared error were calculated by season, grade, and condition. The correlation between growth and squared error may help show if the algorithm is less accurate for students with more or less growth. For example, if the correlation is positive, it may indicate the algorithm is struggling to keep up with students who grew a lot over the time period in the data. How this differs across conditions could have implications for how urn sizes may influence the effectiveness of the algorithm. The correlation between the number of items responded to and squared error may help show how the algorithm performs with more items.

If the correlation is negative, it would support the expectation that as the algorithm is given more information, it performs better. To help interpret the correlation between growth and squared error and the correlation between number of items responded to and squared error, the correlation between number of items responded to and growth was also calculated.

Last, the distribution of b-parameter estimates was compared to the student proficiency distributions by season, grade the item was written for, and condition. B-parameters were calculated by taking the average estimated proportion of green balls in the item urn over student responses after dropping the first 500 responses. Items with fewer than 1,000 responses were not included in the examined distribution. The 1,000 responses criterion was applied before dropping the first 500 responses for each item. This resulted in 724 items (out of 927) being included in the distribution. The overlap of the b-parameter estimates distribution and the student proficiency distributions could help understand error. For example, if the b-parameter distribution is generally lower than the winter and spring proficiency distributions, it would indicate that students with higher proficiency may have more error in their algorithm estimates.

CHAPTER 4

RESULTS

This chapter describes the results of the analyses explained in the method section. Results are organized under the following sections: Assessing the Degree of Error, Assessing the Direction of Error, and Assessing Possible Causes of Error.

4.1 Assessing the Degree of Error

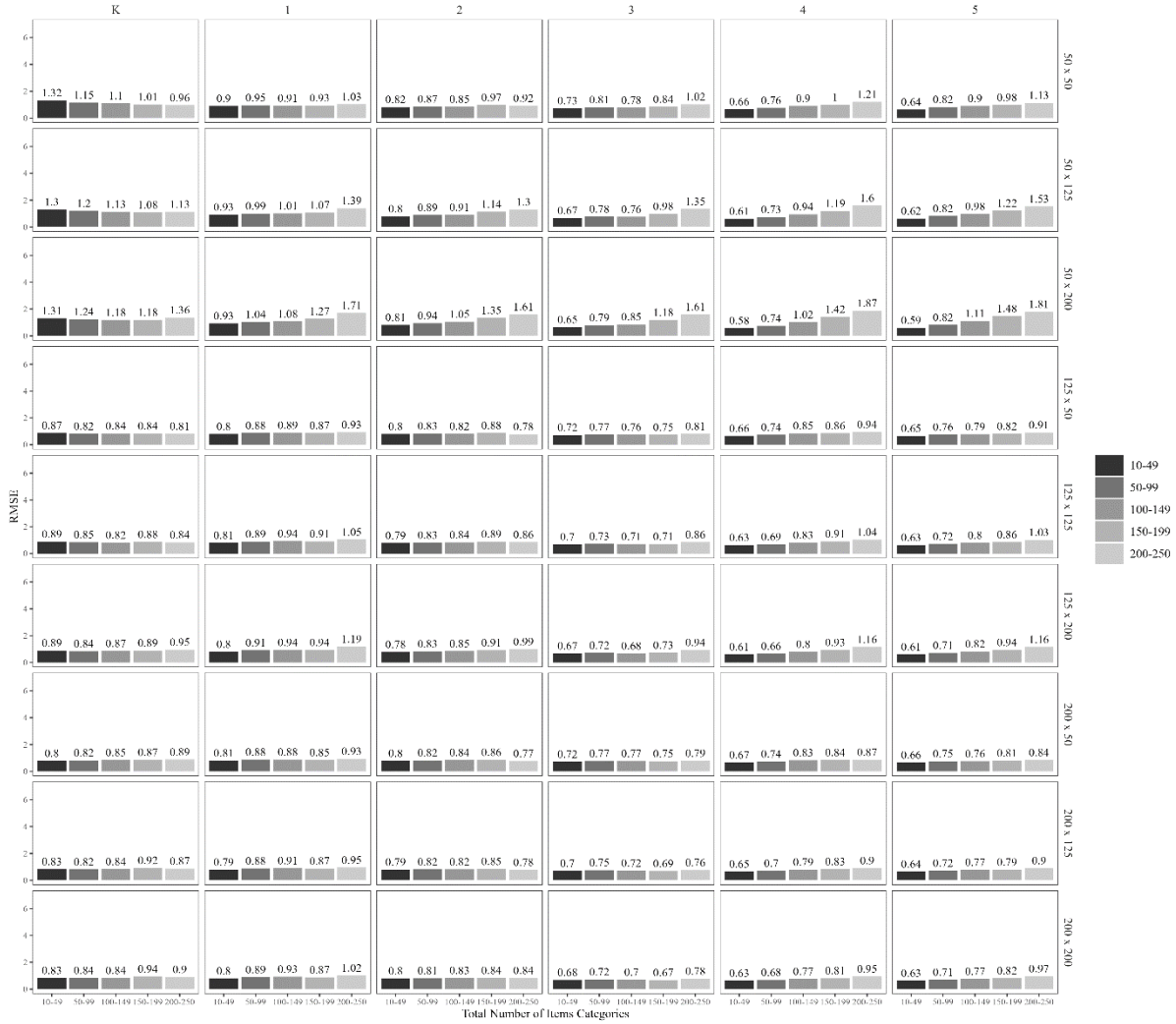
Assessing the degree of error in the algorithm estimates was done by calculating root-mean-square error (RMSE), the proportion of the sample with an algorithm estimate within 1 and 2 standard errors of the Diagnostic estimate, and classification consistency. Each of these metrics was calculated for each season (winter and spring), grade, and item group (i.e., the grouping of students by the number of items they responded to) across the nine study conditions.

4.1.1 Root-mean-square error

Figures 8 and 9 show the RMSE value calculated for each item group, grade, and condition for the winter and spring Diagnostic respectively.

Figure 8

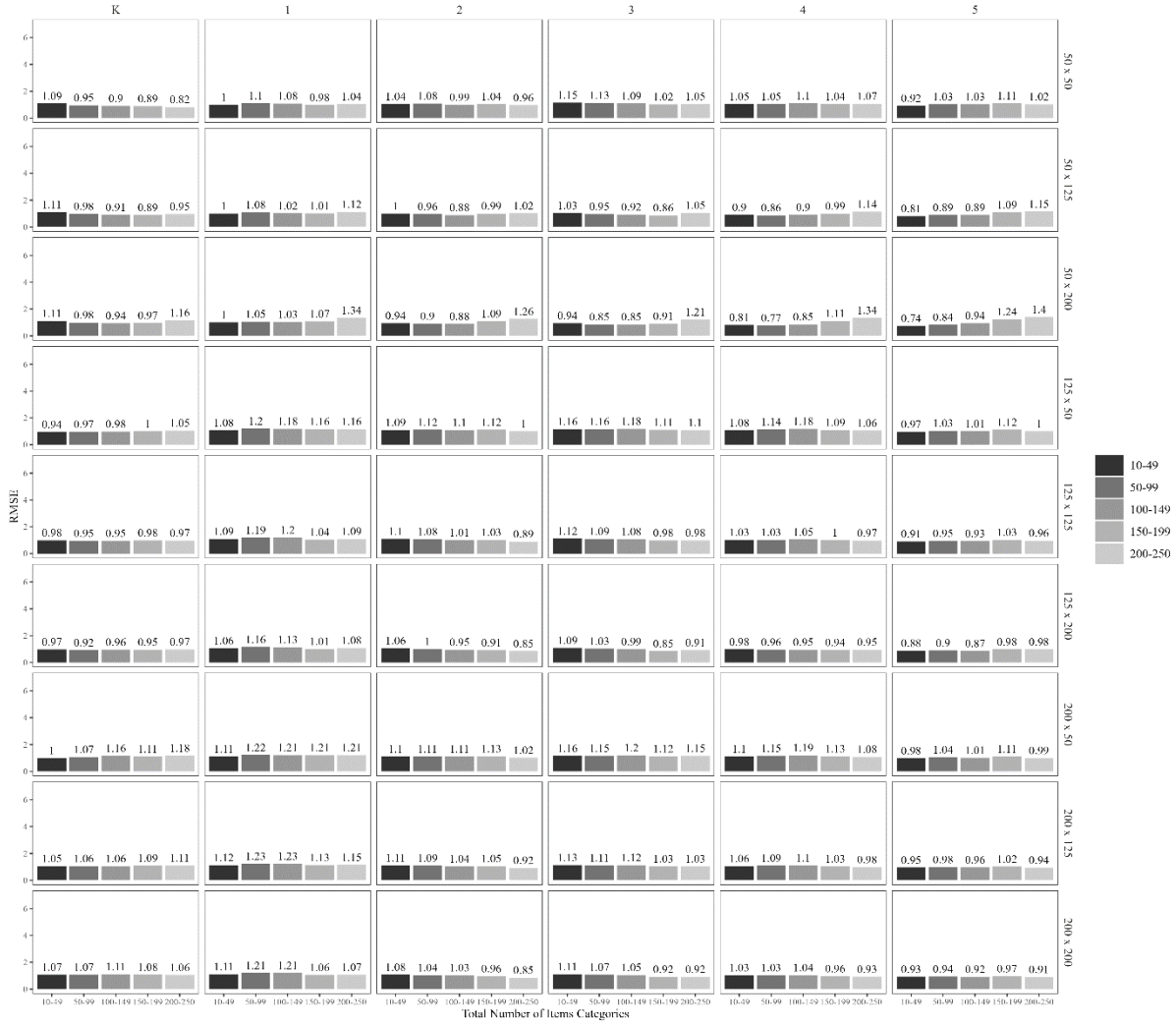
Root-Mean-Square Error for Winter by Item Group, Grade, and Condition



Note. The x-axis is item group (i.e., groupings by the total number of items students responded to). The y-axis is RMSE value. The columns of the plot are student grade. The rows of the plot are study conditions (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the RMSE value for the represented item group, grade, and condition.

Figure 9

Root-Mean-Square Error for Spring by Item Group, Grade, and Condition



Note. The x-axis is item group (i.e., groupings by the total number of items students responded to). The y-axis is RMSE value. The columns of the plot are student grade. The rows of the plot are study conditions (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the RMSE value for the represented item group, grade, and condition.

Across grades and conditions, the larger item groups tend to have higher RMSE values than smaller item groups. An exception to this is kindergarten. For conditions with smaller person urn sizes, the smaller item groups in kindergarten have higher RMSE values. This flips as person urn size gets larger across conditions. As the person urn size gets larger, RMSE values tend to decrease. As the item urn size gets larger, RMSE values tend to increase; however, this pattern appears mostly for

larger item groups (e.g., 100–149, 150–199, and 200–250). For the smaller item groups (i.e., 10–49 and 50–99), sometimes, as the item urn size gets larger, RMSE values decrease (e.g., Grade 3, item group 10–49, the conditions in which person urn size is 125, in winter and spring). Other times, the RMSE value increases (e.g., Grade 1, item group 50–99, conditions in which person urn size is 50, in winter). And, in other instances, the RMSE value stays about the same (e.g., Grade 2, item group 10–49, conditions in which person urn size is 200, in winter and spring). Overall, spring RMSE values are slightly higher than winter RMSE values; however, the range for winter RMSE values is larger than the range of spring RMSE values. The largest RMSE values in Figure 8 are around 1.8 while the smallest values are around 0.6. In Figure 9, most values are around 1.0.

4.1.2 Proportion Within 1 and 2 Standard Errors

Figures 10 and 11 show the proportion of the sample that has their algorithm estimate within plus and minus 1 standard error of the Diagnostic estimate for winter and spring respectively.

Figure 10

Proportion within 1 Standard Error of Winter Diagnostic by Item Group, Grade, and Condition

Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

Figure 11

Proportion within 1 Standard Error of Spring Diagnostic by Item Group, Grade, and Condition

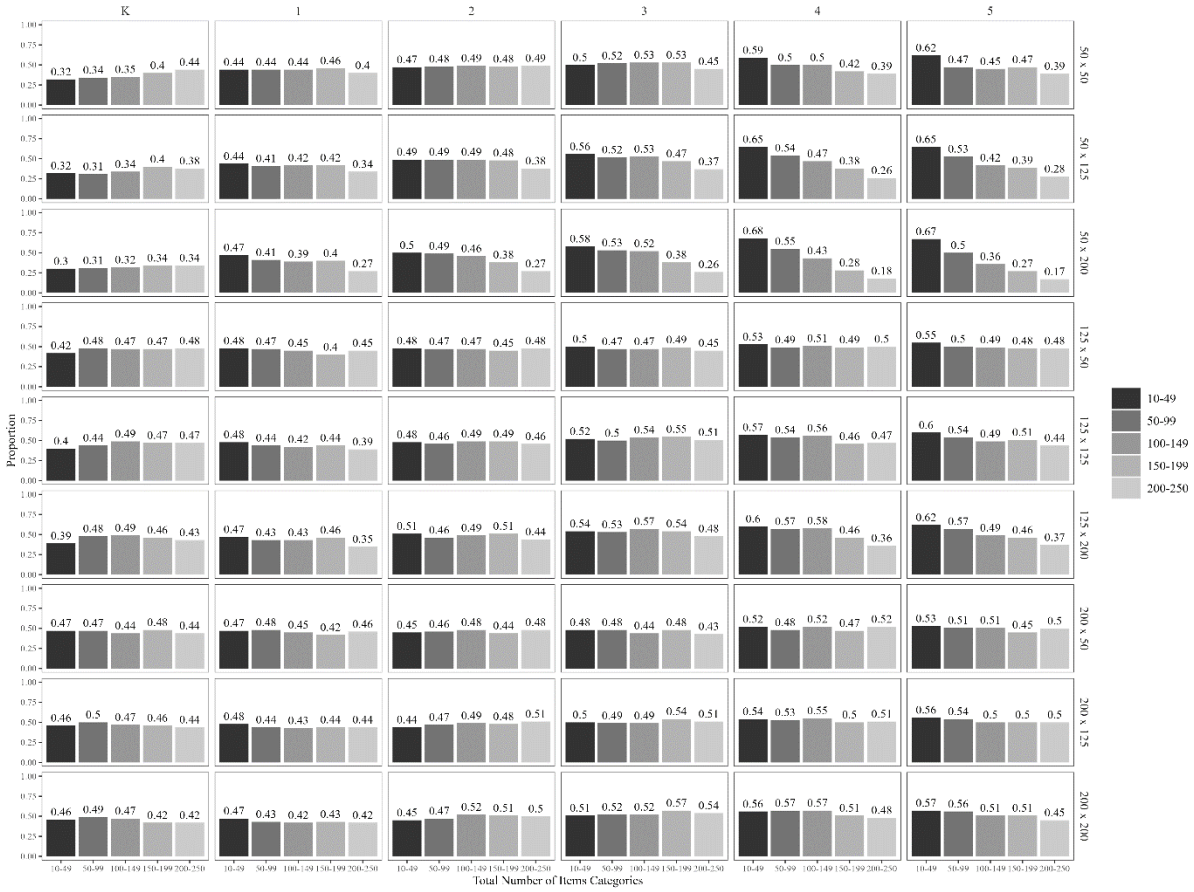
Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

Overall, a greater proportion of the sample has their algorithm estimate within 1 standard error of the Diagnostic estimate for winter compared to spring; however, the highest proportions for winter are around 0.37. In neither winter nor spring were a majority of the sample within 1 standard error of the Diagnostic estimate. For winter, Figure 10, Grades 4 and 5 have larger proportions for smaller item groups compared to larger item groups for conditions in which the person urn size is 50. This trend is reduced for conditions in which the person urn size is 125 and disappears for conditions in which the person urn size is 200. For the spring, Figure 11, proportions decrease across grades as person urn size increases. Overall, larger item groups tend to have larger proportions in the spring.

Figures 12 and 13 show the proportion of the sample with their Urnings algorithm estimate within 2 standard errors of the Diagnostic estimate for winter and spring respectively.

Figure 12

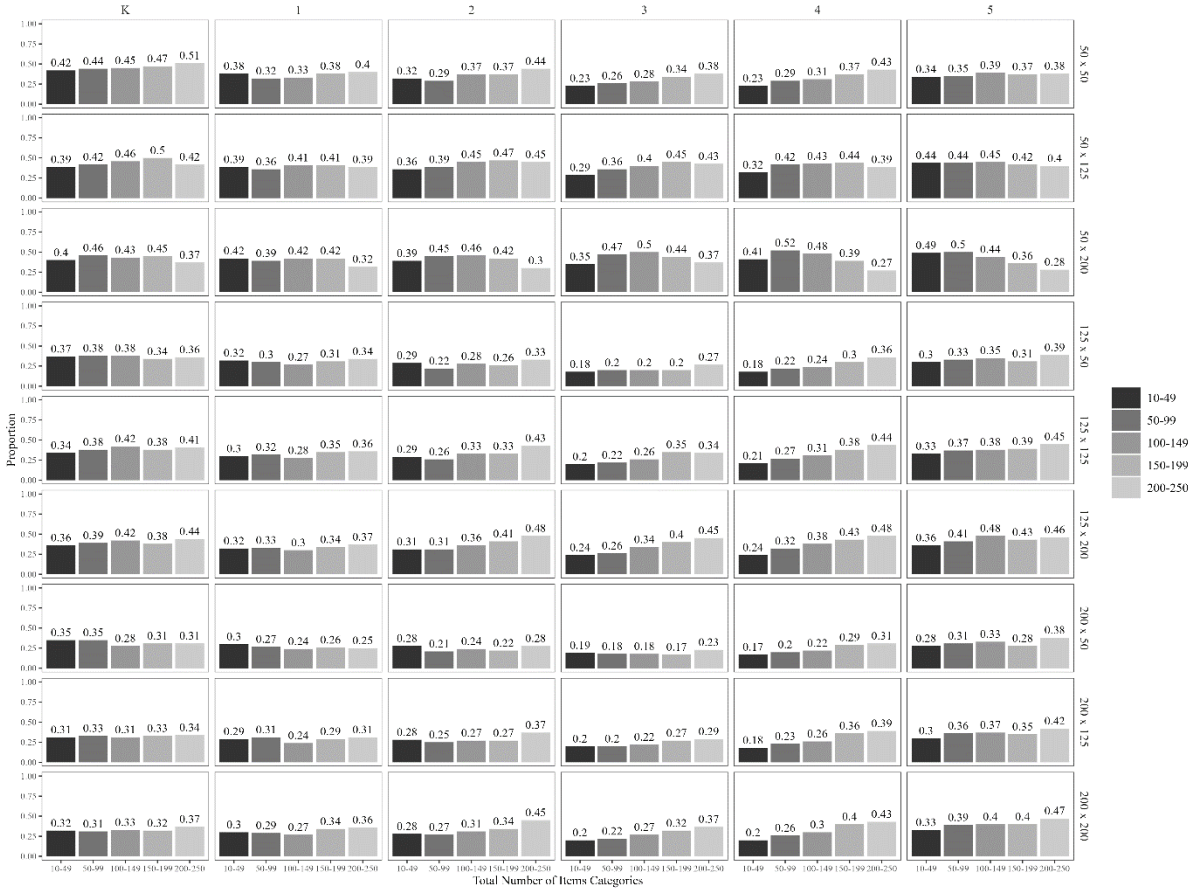
Proportion within 2 Standard Errors of Winter Diagnostic by Item Group, Grade, and Condition



Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

Figure 13

Proportion within 2 Standard Errors of Spring Diagnostic by Item Group, Grade, and Condition



Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

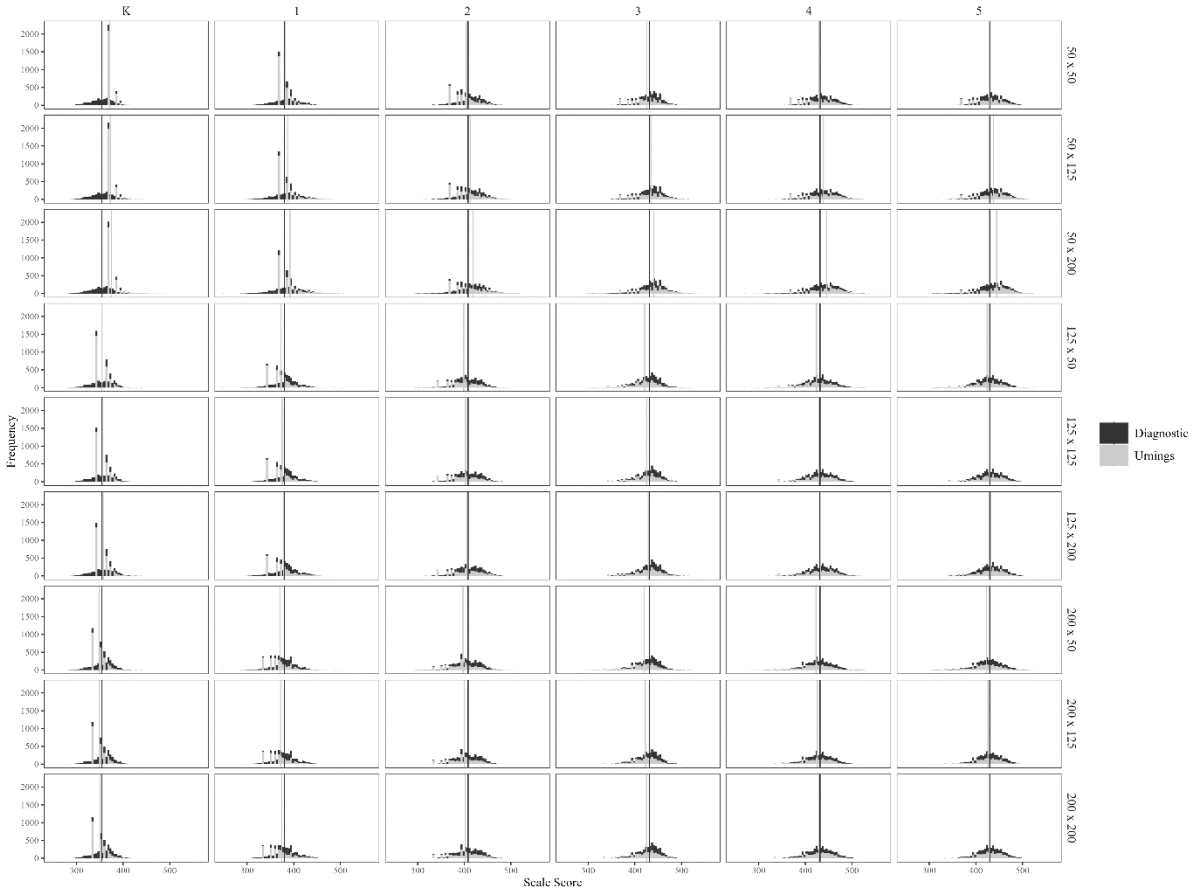
For winter, Figure 12, for Grades 4 and 5, for conditions in which the person urn size is 50, larger item groups tend to have smaller proportions than smaller item groups. This pattern disappears as person urn size increases across conditions. Kindergarten tends to have the opposite pattern but that also disappears as person urn size increases. For spring, Figure 13, proportions are lower overall. There are also the same patterns as Figure 11, showing the proportion of the sample within 1 standard error for spring. Comparing Figures 12 and 13, proportions tend to be larger for winter than spring. The largest proportions in the spring data are around 0.5. For the winter, the largest values are around 0.6.

4.1.3 Classification Consistency

Figures 14 and 15 show the scale score distributions for the Diagnostic and the Urnings algorithm by grade and study conditions for winter and spring respectively. These scale scores were used to determine classification consistency.

Figure 14

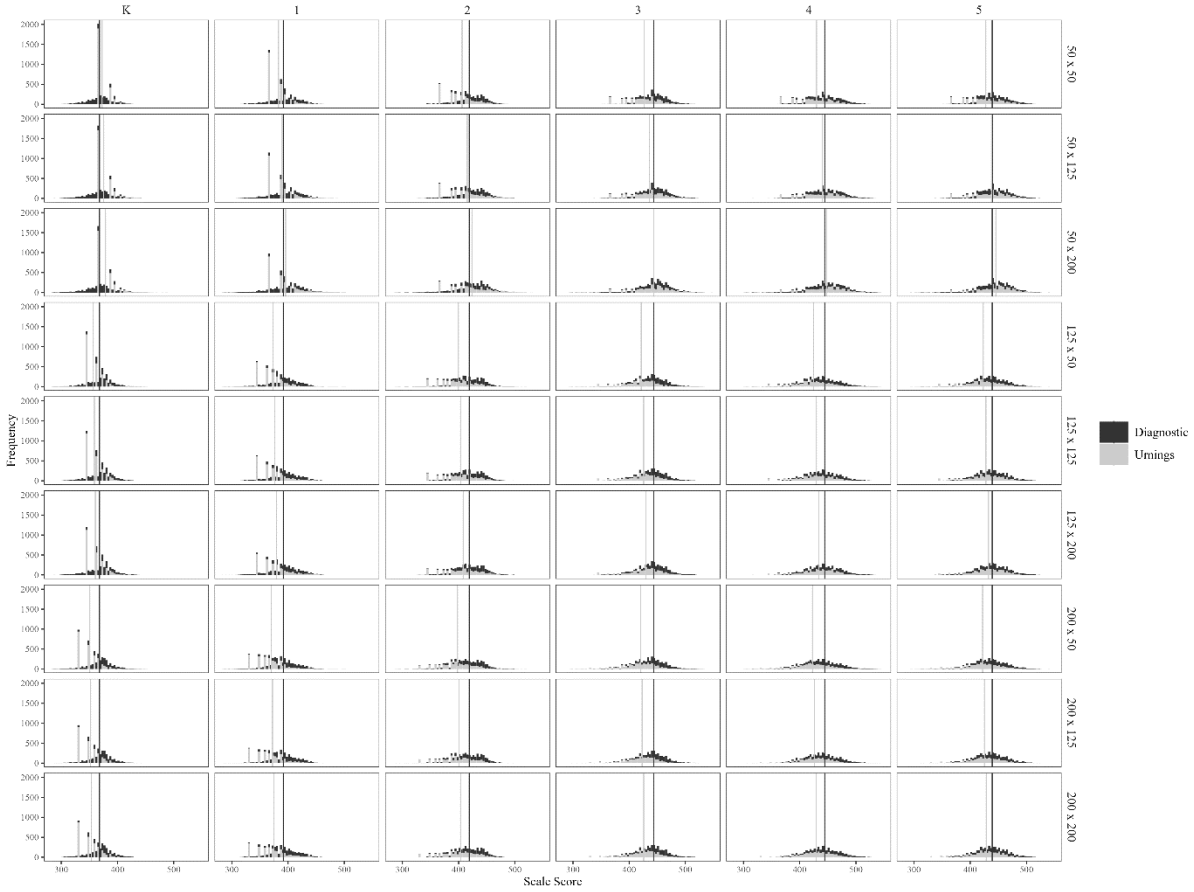
Scale Score Distribution for Winter by Grade and Item Group



Note. The x-axis is scale score. The y-axis is frequency. The columns of the plot are student grade. The rows of the plot are item group. Within a square, the darker distribution is the distribution of scale scores from the Diagnostic. The lighter distribution is the distribution of scale scores from the Urnings algorithm. The lines are the average scale score for the Diagnostic and Urnings respectively. The shading of the lines corresponds to the shading of the distributions.

Figure 15

Scale Score Distribution for Spring by Grade and Item Group



Note. The x-axis is scale score. The y-axis is frequency. The columns of the plot are student grade. The rows of the plot are item group. Within a square, the darker distribution is the distribution of scale scores from the Diagnostic. The lighter distribution is the distribution of scale scores from the Urnings algorithm. The lines are the average scale score for the Diagnostic and Urnings respectively. The shading of the lines corresponds to the shading of the distributions.

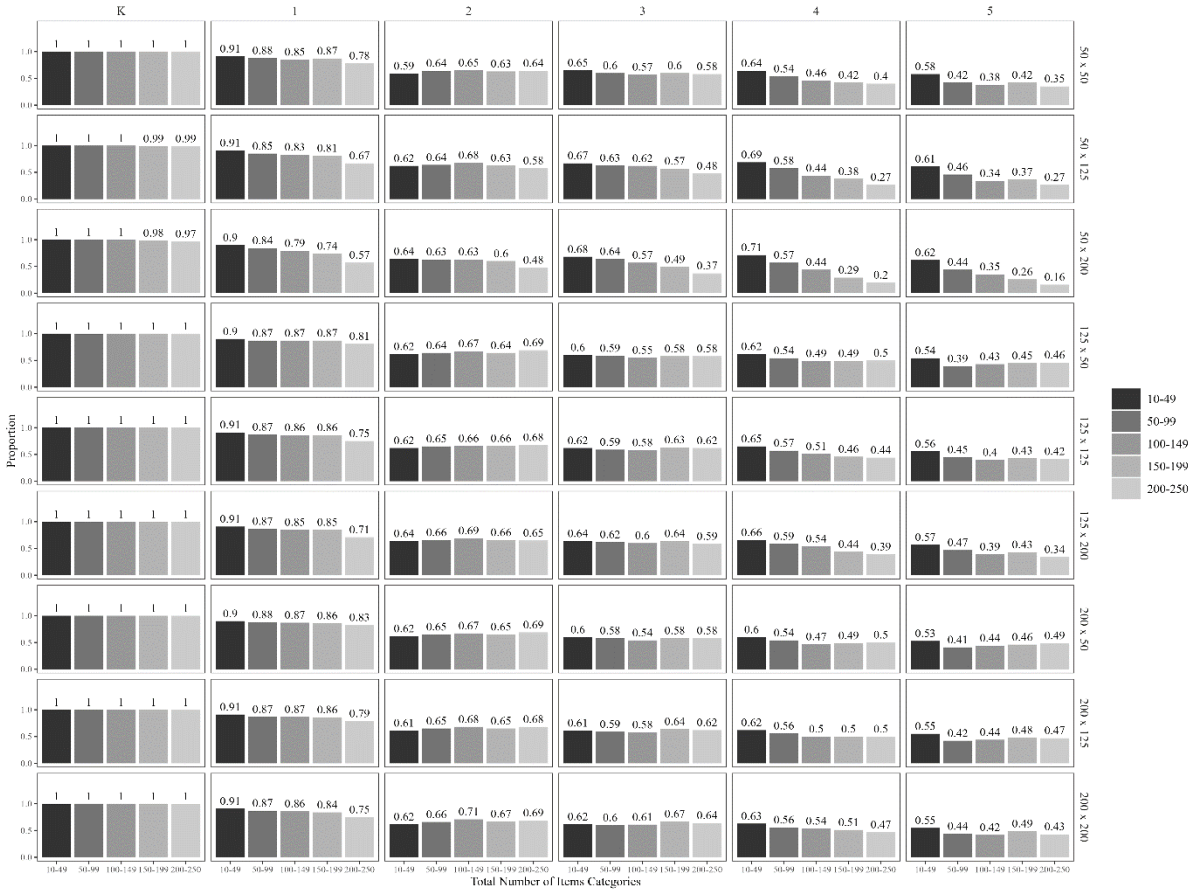
It can be seen for both winter, Figure 14, and spring, Figure 15, that conditions with smaller person urn sizes have fewer unique Urnings algorithm scale scores. This is particularly true for kindergarten and Grade 1. When the person urn size is smaller, there are fewer unique estimates possible for students. Smaller person urn sizes also set a floor and ceiling for possible estimates. Both these effects of urn size are likely contributing to the fewer unique values seen for students in lower grades. For example, under conditions in which the person urn size is 50, both kindergarten and Grade 1 students have a median logit proficiency estimate from the algorithm of -3.89, which is the

lower possible logit estimate for that urn size. Similarly, for kindergarten students under conditions in which the person urn size is 125, the median logit proficiency estimate is -4.82, which is the lowest possible logit estimate for that person urn size.

Figures 16 and 17 show classification consistency between the grade-level placement of the Urnings algorithm and the Diagnostic by item group, grade, and study condition for winter and spring respectively.

Figure 16

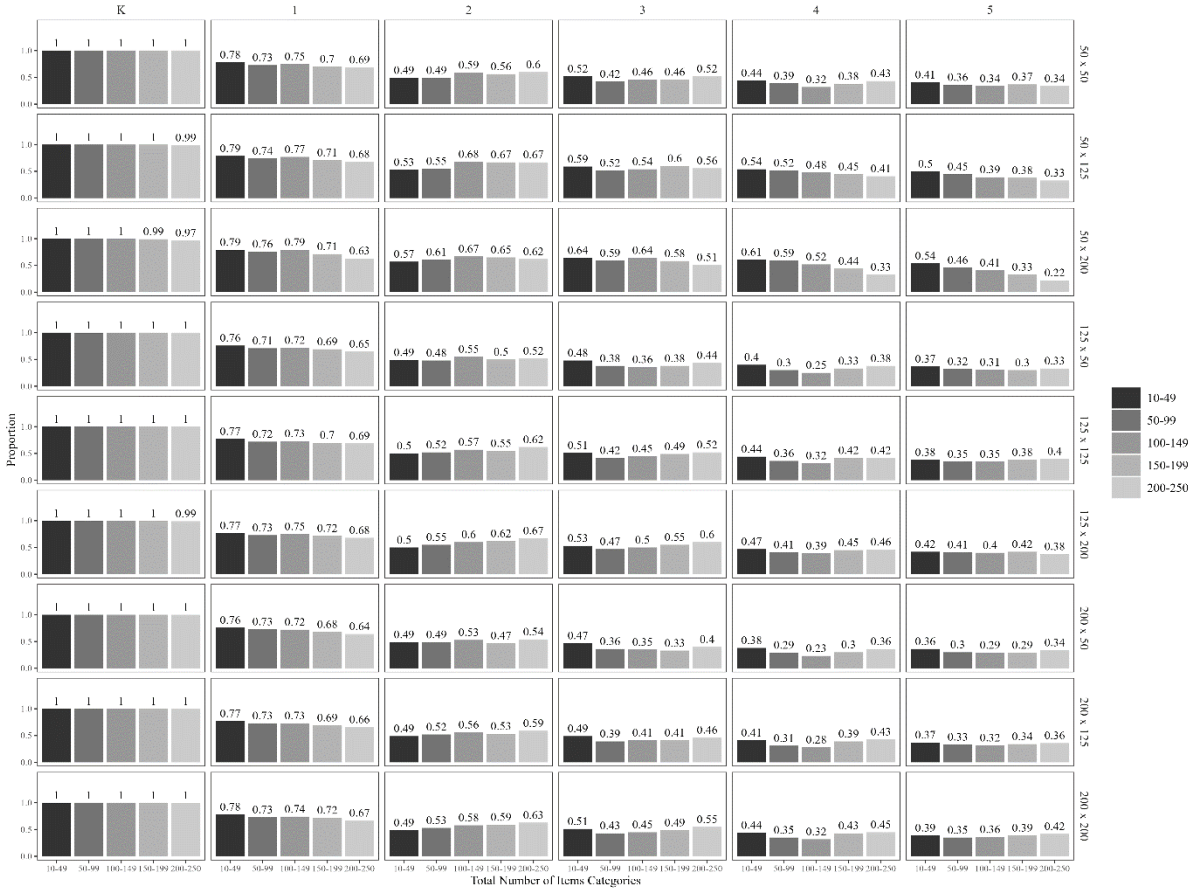
Grade-level Classification Consistency for Winter by Item Group, Grade, and Condition



Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

Figure 17

Grade-level Classification Consistency for Spring by Item Group, Grade, and Condition



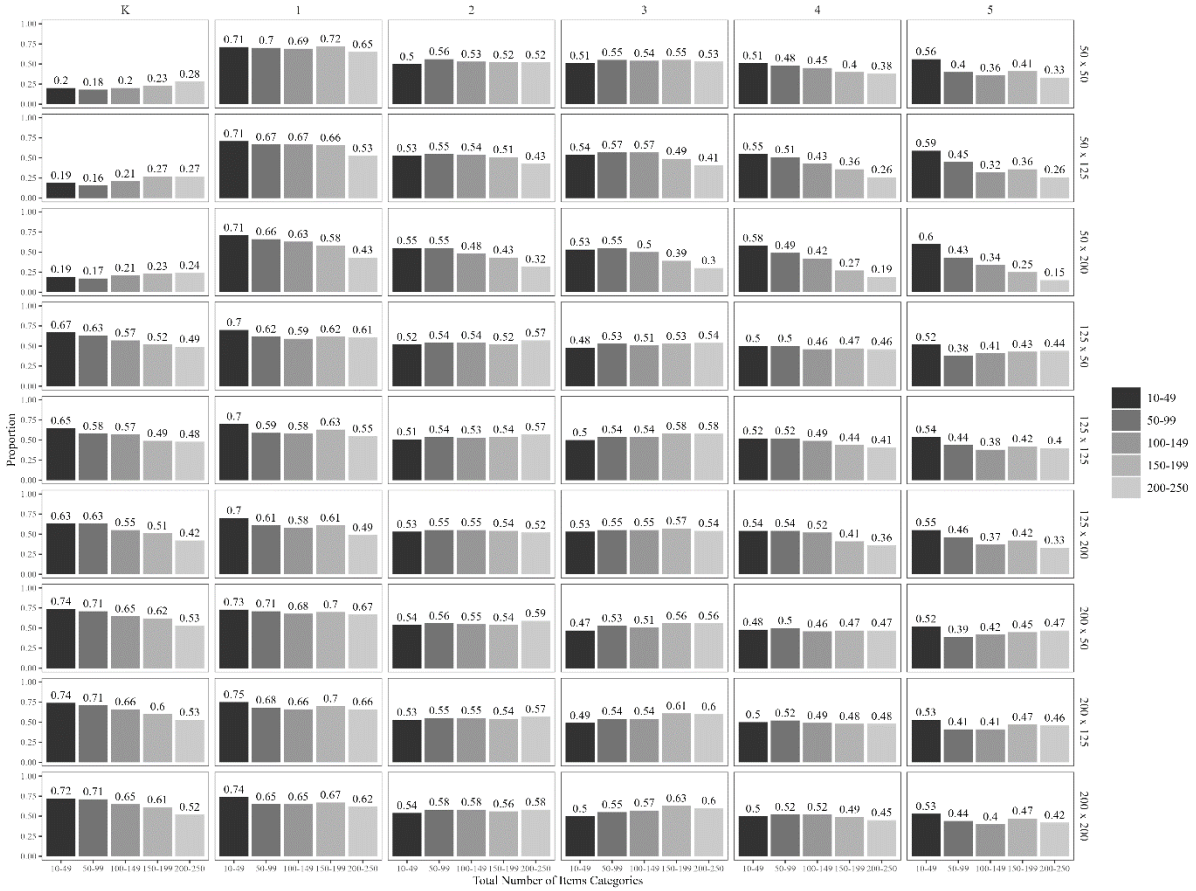
Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

The patterns in grade-level classification consistency are consistent across winter, Figure 16, and spring, Figure 17. Kindergarten students have almost perfect grade-level classification consistency for both winter and spring. Grade 1 students also have relatively high consistency. From Grade 2 to 5, as grade increases, classification consistency tends to decrease. For Grades 1 to 5, consistency is higher for winter than spring. Consistency does not seem to be substantially influenced by study condition.

Figures 18 and 19 show classification consistency between the sub-level placement (early, mid, late—and emerging for kindergarten) of the Urnings algorithm and the Diagnostic by item group, grade, and study condition for winter and spring respectively

Figure 18

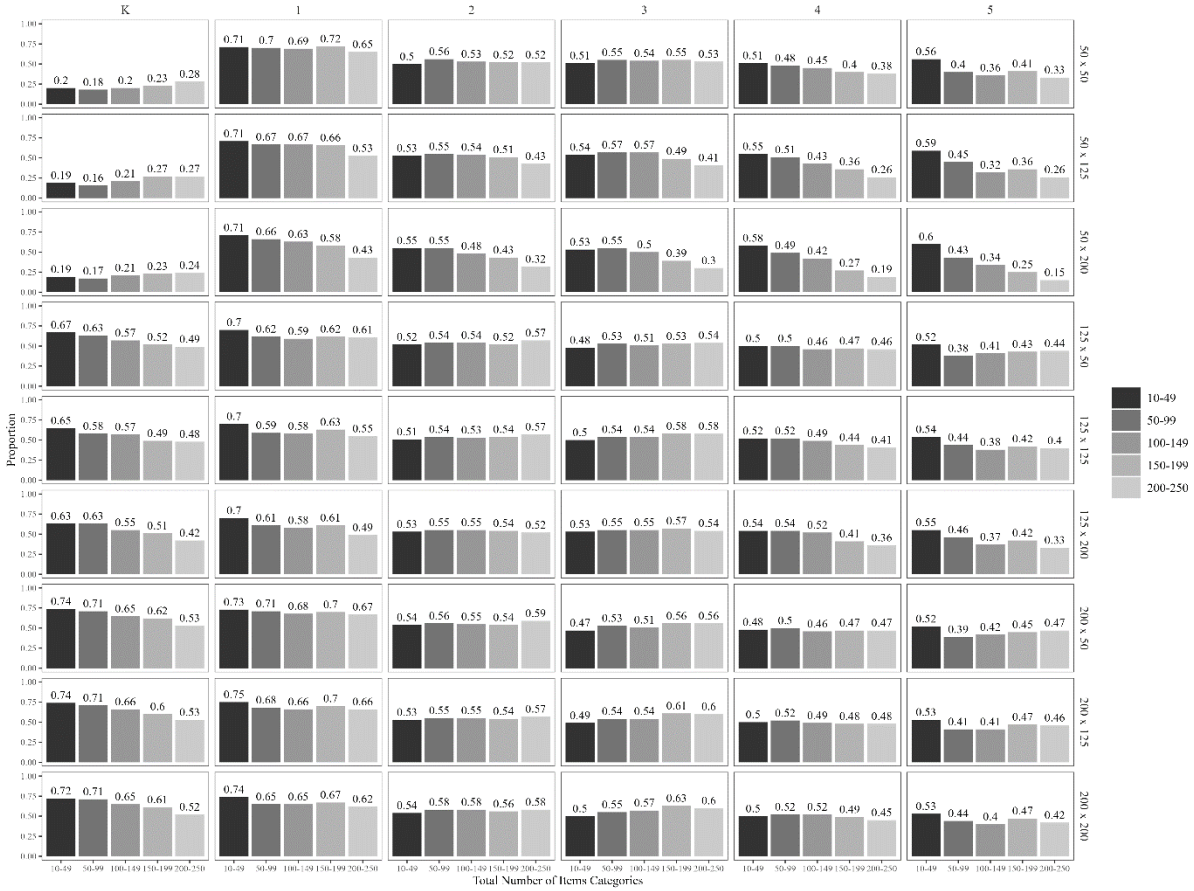
Sub-level Classification Consistency for Winter by Item Group, Grade, and Condition



Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

Figure 19

Sub-level Classification Consistency for Spring by Item Group, Grade, and Condition



Note. The x-axis is item group. The y-axis is proportion. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). Within each square, the shade of the bar shows the item group it represents. The value above each bar is the proportion for the represented item group, grade, and condition.

For both winter, Figure 18, and spring, Figure 19, for conditions in which the person urn is size 50, the proportion of the sample with consistent classification is strikingly lower in kindergarten than other grades. This is because the Diagnostic placement for many kindergarten students was the lowest possible—emerging. All scale score estimates from the Urnings algorithm were above the score range for this classification. This is because of the urn size of 50. The urn size controls the range of possible estimates. As the urn size gets larger, higher and lower estimates are possible. With an urn size of 50, the lowest and highest possible logit proficiency estimates are -3.89 and 3.89. With an urn size of 125, the lowest and highest possible logit proficiency estimates are -4.82 and 4.82.

Classifications are more consistent among Kindergarteners for conditions with person urn sizes of 125 and higher because they allow scores that fall within the emerging sub-classification level.

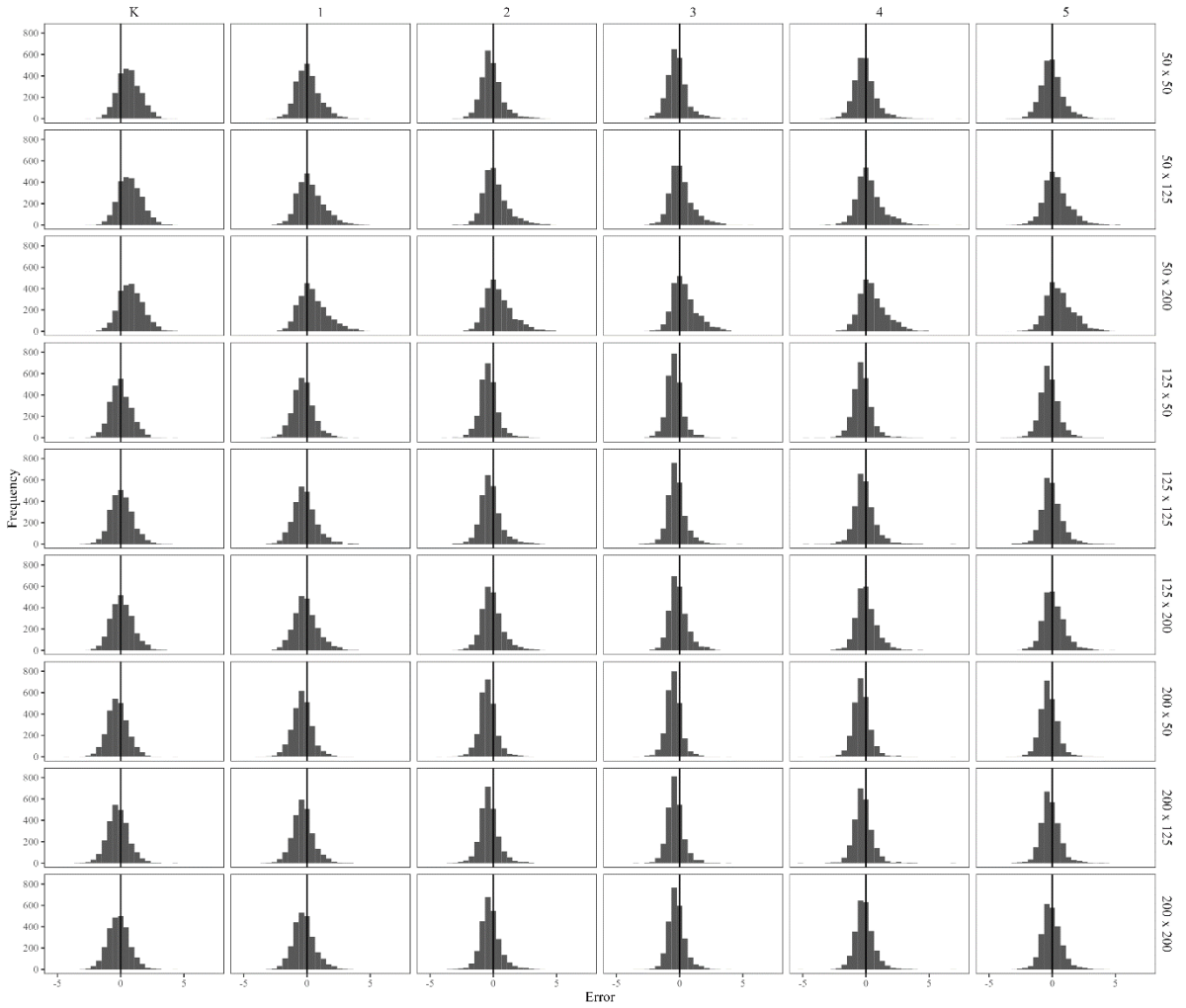
Classification consistency is more similar across item groups for winter compared to spring. Across both winter and spring, for Grade 5, at lower person urn sizes, larger item groups tend to have lower classification consistency. This pattern decreases as person urn size increases across conditions. This pattern also occurs for Grade 1 students (except for conditions in which the person urn size is 50), but it does not decrease as person urn size increases. Also across winter and spring, for Grade 3, larger item groups tend to have greater consistency than lower item groups. This pattern holds across study conditions. Overall, sub-level classification consistency is higher for winter than spring.

4.2 Assessing the Direction of Error

The direction of error was assessed by examining the distribution of error across season, grade, and condition. Figures 20 and 21 show the distributions of error across grade and study conditions across winter and spring respectively. Error was calculated by subtracting the Diagnostic estimate from the Urnings estimate.

Figure 20

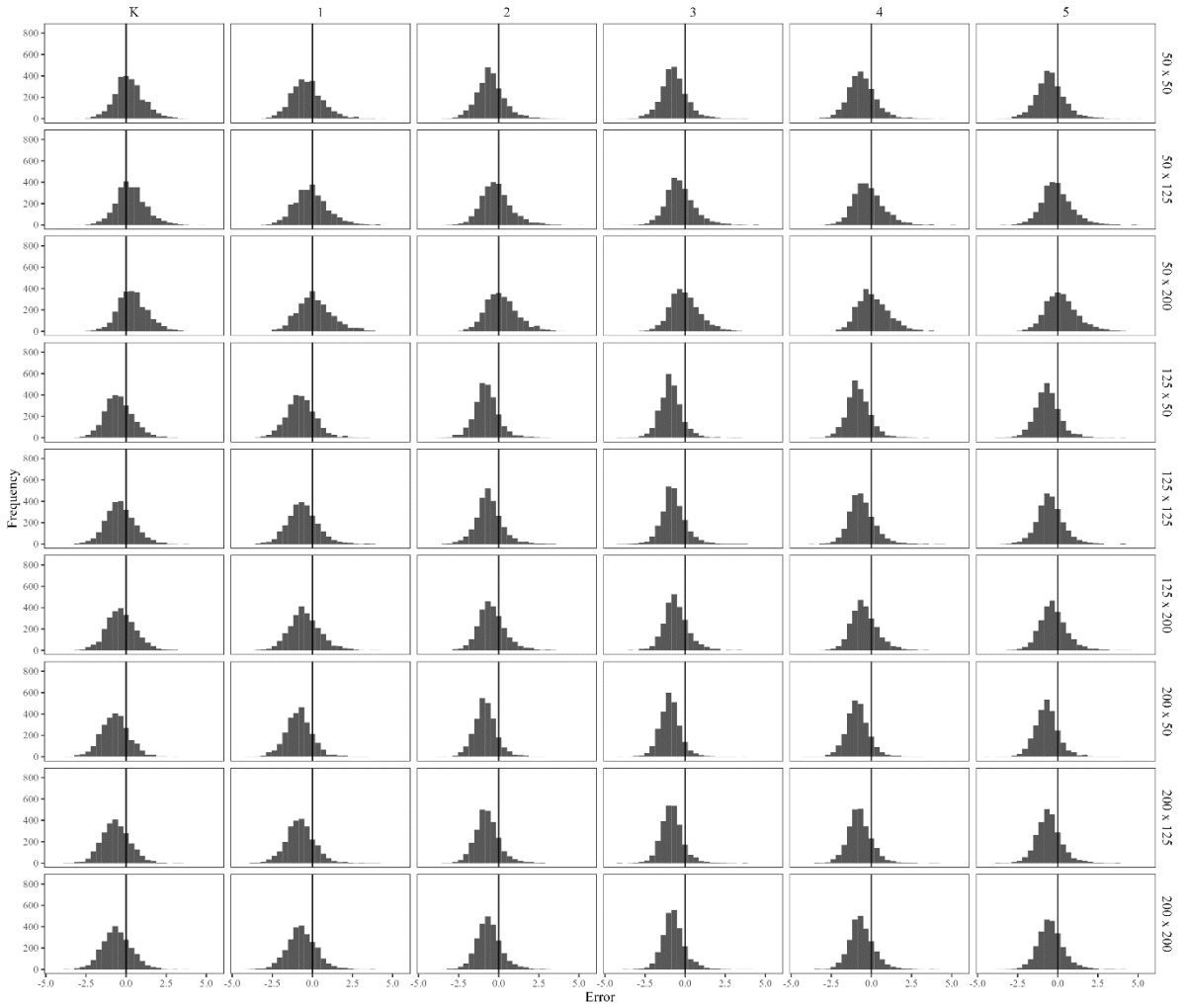
Distribution of Error for Winter by Grade and Condition



Note. The x-axis is error (the difference between the Urnings estimate and the Diagnostic estimate) in logits. The y-axis is frequency. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). The line within each square marks 0 on the x-axis.

Figure 21

Distribution of Error for Spring by Grade and Condition



Note. The x-axis is error (the difference between the Urnings estimate and the Diagnostic estimate) in logits. The y-axis is frequency. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). The line within each square marks 0 on the x-axis.

Across winter, Figure 20, and spring, Figure 21, error tends to be more negative as person urn size increases across study conditions. Within each person urn size, as item urn size increases, error tends to become more positive. Comparing the two effects, the negative influence of person urn size is greater than the positive influence of item urn size. Error also tends to be more negative for higher grades. Overall, error tends to be more negative than positive, indicating the Urnings algorithm tended to underestimate.

4.3 Assessing Possible Causes of Error

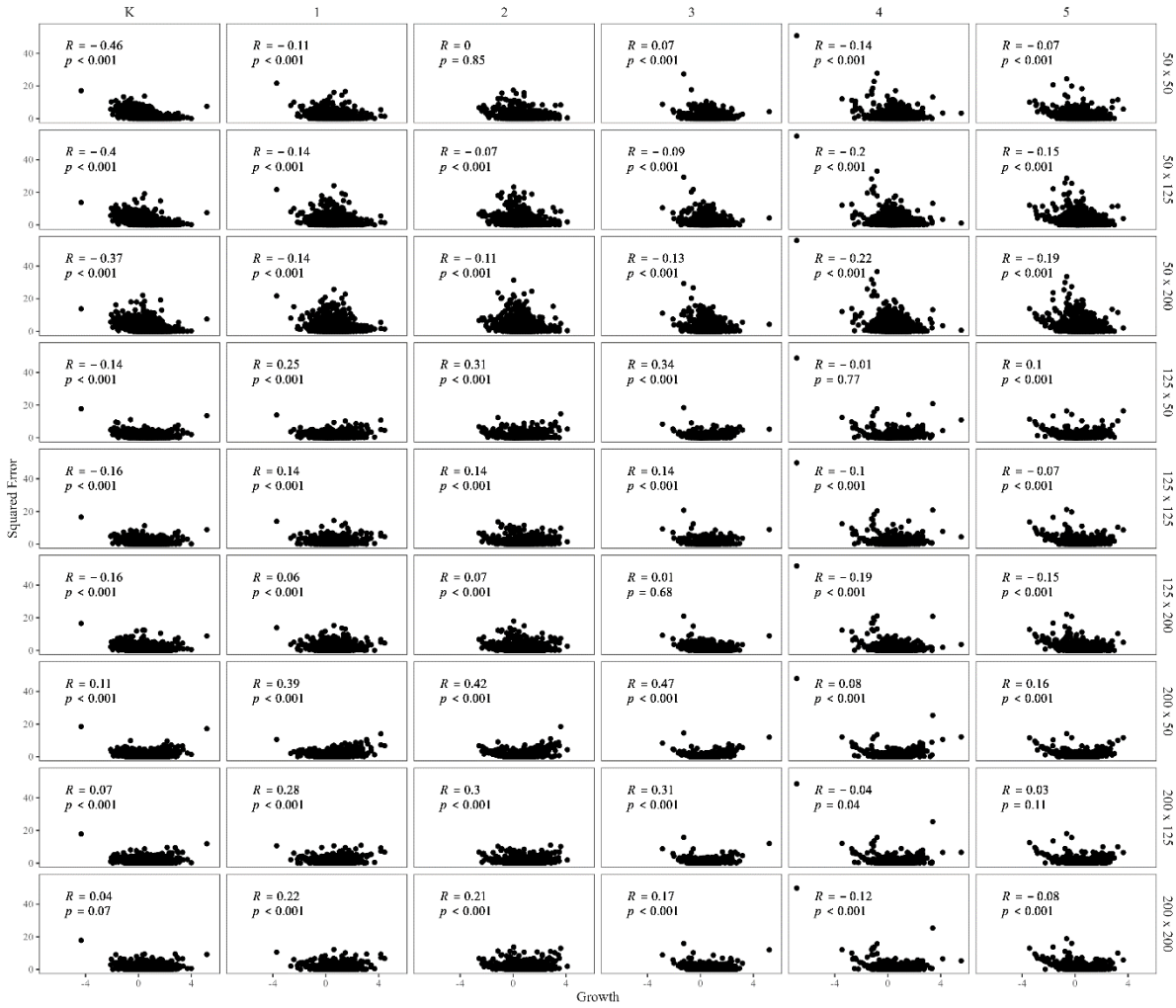
Assessing possible causes of error was done by examining the correlation between growth and squared error, the correlation between number of items responded to and squared error, the correlation between growth and the number of items responded to, and the distribution of b-parameter estimates relative to the distribution of student proficiency estimates from the Diagnostic.

4.3.1 Correlation between Growth and Squared Error

Figures 22 and 23 show scatterplots and the correlation between growth and squared error, by grade and study condition, for winter and spring respectively.

Figure 22

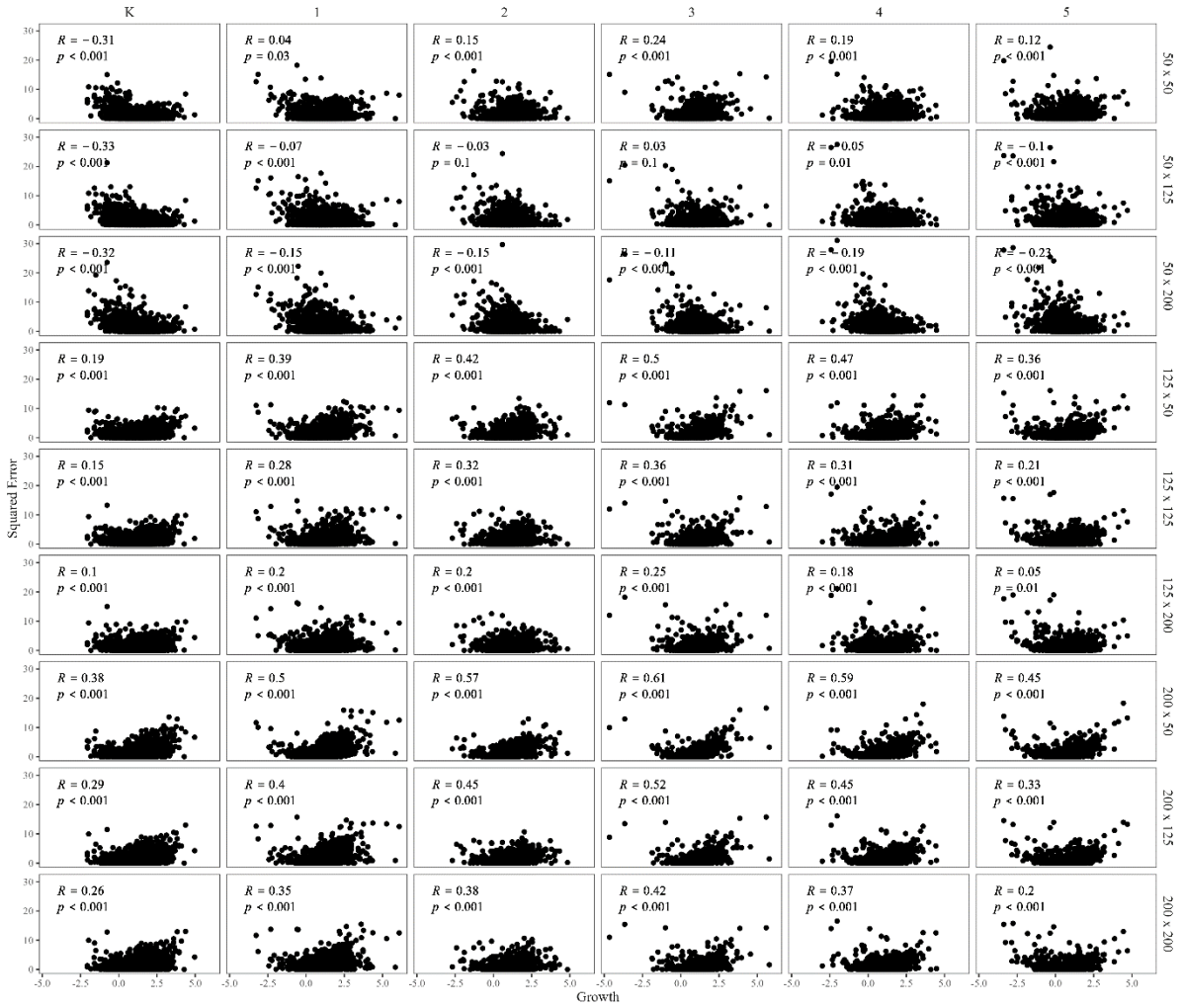
Scatterplot for Growth and Squared Error for Winter by Grade and Condition



Note. The x-axis is growth in logits. The y-axis is squared error. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). The values in each square are the correlation estimate and p-value for the represented grade and condition.

Figure 23

Scatterplot for Growth and Squared Error for Spring by Grade and Condition



Note. The x-axis is growth in logits. The y-axis is squared error. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). The values in each square are the correlation estimate and p-value for the represented grade and condition.

For winter, Figure 22, there does not appear to be a strong or consistent relationship between growth and squared error. Even for instances in which the correlation is significant, it is usually relatively small. The scatterplots also do not indicate strong positive or negative relationships. For spring, Figure 23, there does appear to be a pattern based on person urn size. For conditions in which the person urn size is 50, there tends to be smaller or negative correlations. Some of these scatterplots also indicate a possible inverse relationship (e.g., kindergarten and Grade 1, conditions 50 x 125 and

50 x 200). For conditions in which the person urn size is 125 or 200, the correlations tend to be larger and positive. The scatterplots for conditions in which the item urn size is 200 also indicate a positive relationship. It appears that, within conditions with a given person urn size, as item urn size increases, the correlation decreases. For example, For Grade 5 students, the correlations for conditions 200 x 50, 200 x 125, and 200 x 200 are 0.45, 0.33, and 0.20 respectively. The scatterplots also seem to indicate stronger, positive relationships when the item urn size is smaller.

4.3.2 Correlation between Number of Items Responded to and Squared Error

Figures 24 shows scatterplots and correlations between number of items responded to and squared error, by grade and study condition, for spring.

Figure 24

Scatterplot for Items Responded to and Squared Error for Spring by Grade and Condition

Note. The x-axis is total items responded to. The y-axis is squared error. The columns of the plot are student grade. The rows of the plot are study condition (person urn size by item urn size). The values in each square are the correlation estimate and p-value for the represented grade and condition.

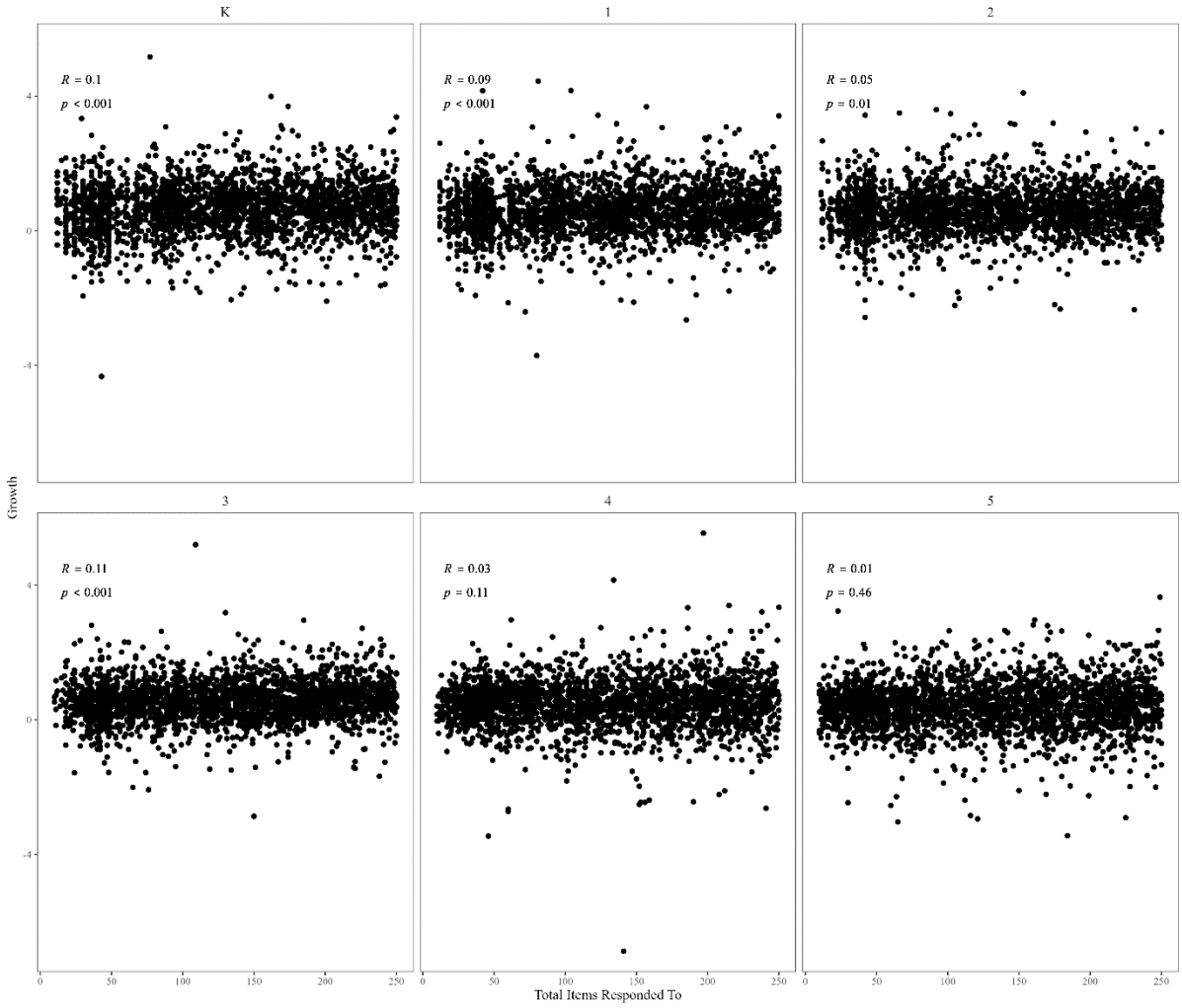
All correlations in Figure 24 are small. Many correlations are negative. The scatterplots generally do not indicate a relationship between the number of items responded to and squared error; however, there are some instances of students who responded to a large number of items having greater squared error. These students seem to be outliers compared to the general pattern in the plots.

4.3.3 Correlation between Number of Items Responded to and Growth

Figures 25 and 26 show scatterplots and correlations between number of items responded to and growth by grade, for winter and spring respectively.

Figure 25

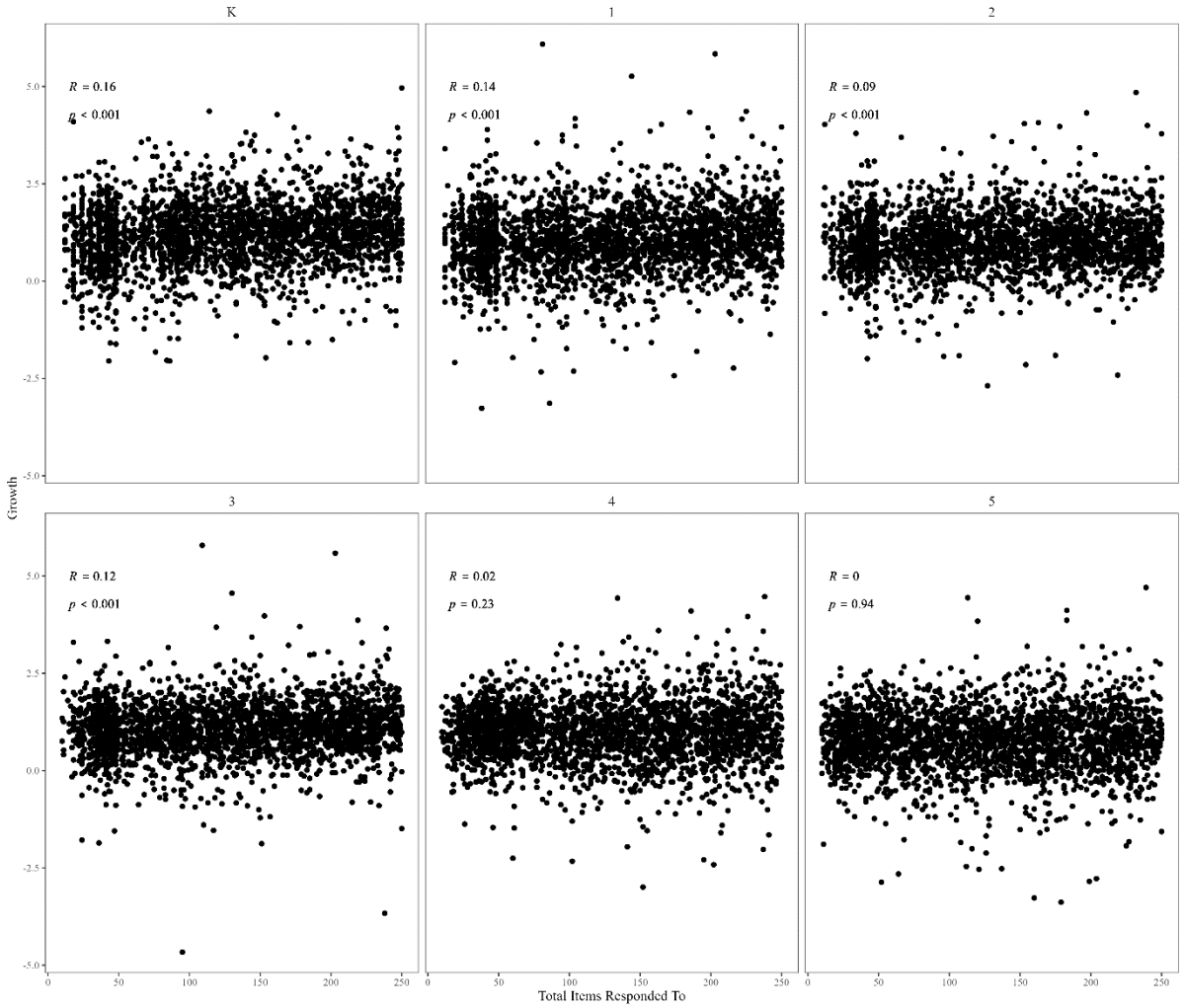
Scatterplot for Items Responded to and Growth for Winter by Grade



Note. The x-axis is total items responded to. The y-axis is growth in logits. Each square is a student grade. Within a square, the values are the correlation estimate for the represented grade and the corresponding p-value.

Figure 26

Scatterplot for Items Responded to and Growth for Spring by Grade



Note. The x-axis is total items responded to. The y-axis is growth in logits. Each square is a student grade. Within a square, the values are the correlation estimate for the represented grade and the corresponding p-value.

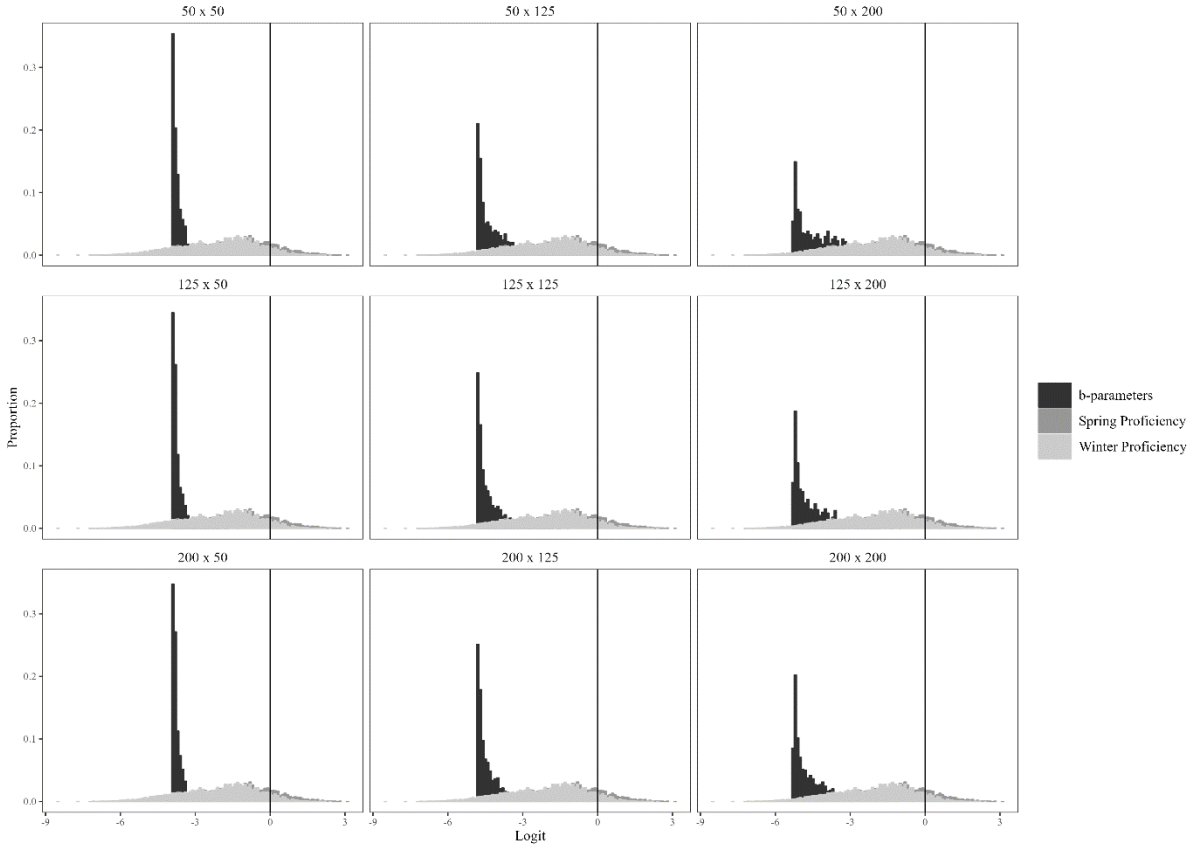
Based on Figures 25 and 26, there does not appear to be a relationship between the number of items responded to and growth. Neither the correlations nor the scatterplots indicate a strong or consistent relationship.

4.3.4 Distribution of b-parameters

Figure 27 shows the distributions of b-parameter estimates, proficiency estimates from the winter Diagnostic, and proficiency estimates from the spring Diagnostic by condition. Figure 28 shows the same distributions by condition and grade.

Figure 27

Distribution of b-parameter Estimates by Condition

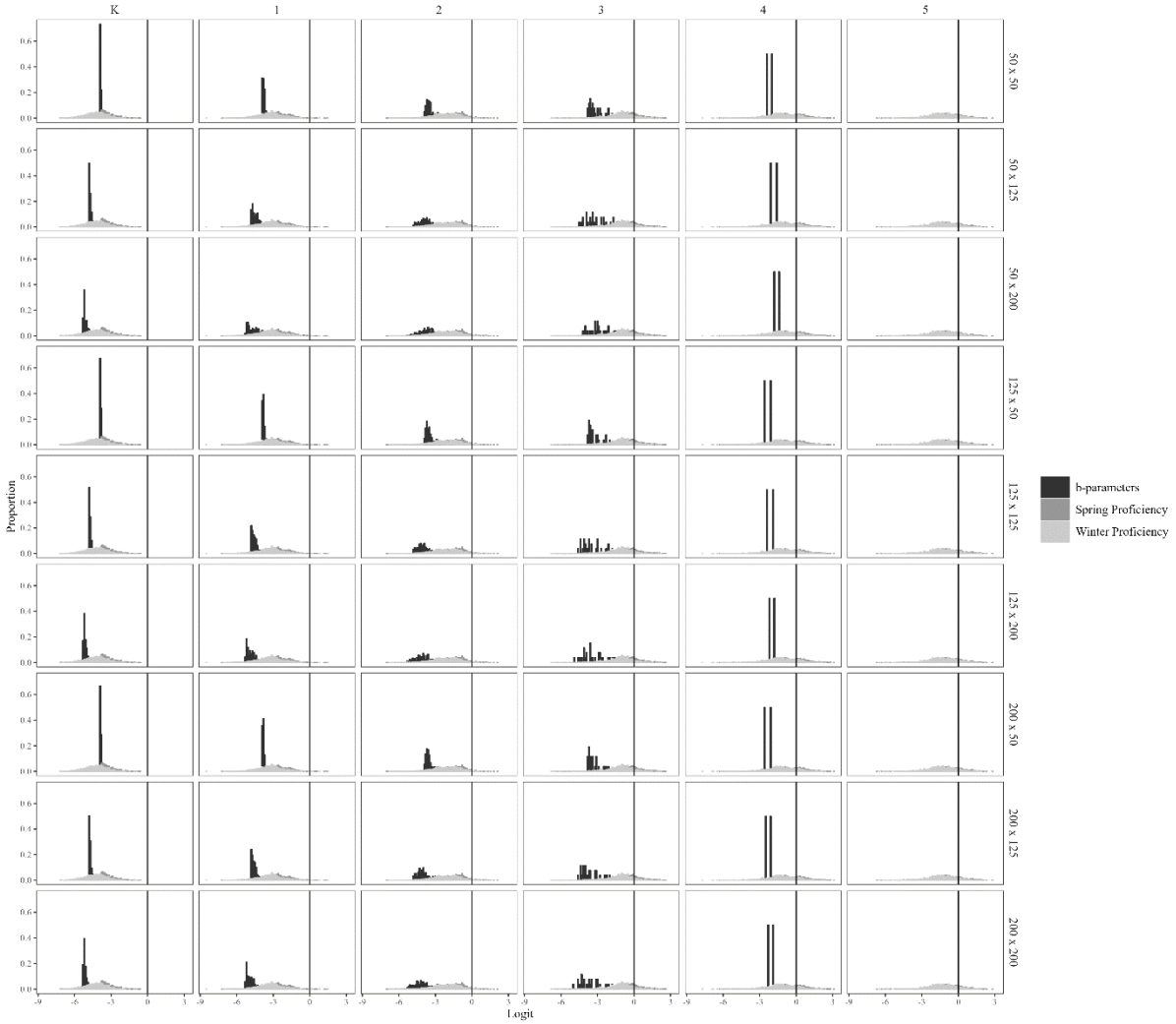


Note. The x-axis is logits. The y-axis is proportion of the sample (with b-parameter, winter proficiency, and spring proficiency estimates treated separately). Each square is a condition (person urn size by item urn size). The line in each square marks 0 on the logit scale.

Across conditions, the b-parameter distribution covers, but not entirely, the lower end of both the winter and spring proficiency distributions. Also, within each condition, there seems to be a floor effect. Many b-parameter estimates tend to be at the lower limit created by the item urn size. As item urn size increases, the distribution of b-parameter estimates becomes more spread, but largely towards more positive values.

Figure 28

Distribution of b-parameter Estimates by Grade and Condition



Note. The x-axis is logits. The y-axis is proportion of the sample (with b-parameter, winter proficiency, and spring proficiency estimates treated separately). The columns of the plot are grade for both the students and items (e.g., the students in column 4 are Grade 4 students, and items in column 4 are items written for Grade 4 material). The rows of the plot are study condition (person urn size by item urn size). The line in each square marks 0 on the logit scale.

Across conditions, as grade increases, b-parameter estimates tend to be more positive. Across grades and conditions, like Figure 27, the b-parameter estimate distribution tends to cover, but not fully, the lower end of the winter and spring proficiency estimate distributions. From kindergarten to Grade 3, as grade increases, the distribution of b-parameter estimates becomes more spread. The b-parameter distributions for Grade 4 are likely peaked in two places because there were very few items

in the data written for Grade 4. Similarly, there are likely no b-parameter estimate distributions visible in the column for Grade 5 because there were almost no items in the data written for Grade 5.

CHAPTER 5

DISCUSSION

This chapter details responses to the research questions of this study based on analyses results. The chapter is organized by the research questions of this study.

5.1 How well does the Urnings algorithm track student growth trajectories?

The response to this question is based on RMSE values, the proportion of the sample with an Urnings estimate within 1 and 2 standard errors of the Diagnostic estimate, and classification consistency. For some season, grade, and study condition combinations the algorithm performed well; however, the performance of the algorithm was inconsistent, and, overall, there was substantial error across metrics. RMSE values were generally around 1.0 in the spring. In the winter, some of the values were as high as 1.8. For the proportion of the sample with a Urnings estimate within 2 standard errors of the Diagnostic estimate, some of the highest values were only around 0.5. For classification consistency, at the grade level, the algorithm performed well for kindergarten but had values around 0.3 and 0.4 for higher grades. At the sub level, many of the values were below 0.5 in both winter and spring. Also concerning is that the algorithm tended to have more error in spring compared to winter. Ideally, the algorithm would perform consistently across time in tracing student growth.

Furthermore, while it seems that increasing person urn size for this sample decreases error in certain ways (e.g., tends to lower RMSE) the effect is likely not strong enough to substantially increase the performance of the algorithm. For example, in the spring, while person urn sizes of 200 tend to have lower RMSE values than person urn sizes of 50, all RMSE values were relatively close to 1.0.

5.2 For Whom and When does Error Occur and What is the Direction of Error?

For RMSE values, there tended to be more error in spring compared to winter. There does not seem to be a difference in error across item groups in spring, but, in winter, for conditions with lower person urn sizes, larger item groups seem to have more error. RMSE also seems to generally decrease

as person urn size increases. For the proportion of the sample within 1 and 2 standard errors of the Diagnostic estimate, there seems to be more error in spring. Also in spring, larger item groups tend to have less error. There does not seem to be a consistent pattern across grade or study condition for this metric. For classification consistency, there seems to be more error (i.e., inconsistency) in spring and in higher grades. It also appears that, for winter, smaller item groups have less error in conditions in which the person urn size is 50. Overall, students in higher grades tend to have more error, and there tends to be more error in the spring.

The pattern of more error in spring compared to winter suggests that the algorithm performs worse over time—i.e., it is diverging from students' growth trajectories as estimated from the Diagnostic. The overall negative direction of error suggests the algorithm is diverging from students' Diagnostic growth trajectories by lagging below it. This is supported by how error tends to be more negative in spring compared to winter. The pattern of students in higher grades having more error may have to do with the placement of students and the coverage of the scale by items. Higher-grade students, even if they did not place at their chronological grade, likely placed higher on the proficiency scale from the Diagnostic than lower-grade students. Many of the items students responded to were clustered towards the lower end of the scale (bounded by the floor created from the item urn size in the condition). This may have provided better coverage of the area of the scale where lower-grade students placed, thus giving them more accurate proficiency estimates than higher-grade students.

Also, for grade-level and sub-level classification consistency, lower grades have fewer or no lower-grade levels. The algorithm seems to underestimate proficiency. Higher-grade students may have worse results for this metric because there are more lower grades for them to be misplaced into. For example, kindergarten likely has perfect Grade-level classification consistency because there is

no lower grade to place the students into; thus, as the Urnings algorithm underestimates, it hits a classification floor.

5.3 What are Possible Causes of Error?

For the correlation between growth and squared error, for spring, as person urn size increases across condition, a positive relationship tends to emerge. A stronger, positive relationship emerging for larger person urn sizes may be because a larger urn size restricts how much the algorithm can adjust a proficiency estimate between items. If the algorithm tends to lag below growth as estimated from the Diagnostic, then restricting how much it can adjust a proficiency estimate may exacerbate the discrepancy. Holding person urn size constant, as item urn size increases, correlations tend to become more negative. All items start at the middle of the scale range defined by the item urn size in a condition. Most item difficulty estimates seem to decrease as learners respond to items. A larger item urn size slows down the decrease in estimated item difficulty across learner responses; thus, the algorithm records learners responding to more items that, at the time of the response, are estimated to have a higher difficulty. This likely causes the algorithm to increase student proficiency estimates bringing algorithm estimates closer to Diagnostic estimates.

For the correlation between number of items responded to and squared error, results do not indicate a strong or consistent relationship, though there are instances of students who responded to more items having higher squared error. Given the weak relationship between number of items responded to and squared error, error for students in larger item groups may be from some variable not incorporated into this study. Based on the correlation between number of items responded to and growth, it seems that number of items responded to is not a potential strong driver of growth. This indicates that any differences in error among learners in different item groups is likely not driven by differences in growth.

Concerning the distribution of b-parameter estimates, it seems that error may be coming from the limited range of item difficulties. Across conditions and grades, item difficulty estimate distributions only cover a section of the lower portion of winter and spring proficiency estimate distributions from the Diagnostic. Without more difficult items, the algorithm may not be able to increase learner proficiency estimates much beyond a limited area of the scale. In contrast, on the Diagnostic, learners have opportunities to endorse items of greater difficulty and, thus, their proficiency estimates can reach higher areas of the scale.

Another possible cause of error are the limits of the scale created by the urn size for proficiency estimates. None of the conditions in this study fully cover the scale range of the Diagnostic; thus, there are likely some learners with proficiency estimates that are beyond the range the algorithm can estimate. Increasing the urn size to better approximate the Diagnostic scale would also restrict how much the algorithm could adjust proficiency estimates. It may be that to reach a range that adequately approximates the Diagnostic scale, the urn size would need to be so large that it would paralyze proficiency estimates.

5.4 Limitations and Future Research

While this study provides an evaluation of the Urnings algorithm, it has limitations. The sample was only kindergarten to Grade 5 students. Also, students in higher grades likely placed in lower grade levels in the fall; thus, this sample had a particular proficiency profile. Additionally, *i-Ready* is a complex system with design and internal logic that may differ substantially from other digital learning platforms. All these elements limit the generalizability of these findings to students in higher grades, groups of students with different proficiency profiles, and different online learning platforms. Further, while this study used different person and item urn sizes across conditions, the urn sizes used may not have been optimal given characteristics of the sample or the material in *i-Ready*. The algorithm was also only evaluated at two time points; thus, all other estimates from the algorithm were not considered.

Considering these limitations, it would be beneficial to have additional studies evaluating the Urnings algorithm. Seeing the performance of the algorithm across students of different grades, proficiencies, and within different systems could help the field assess the viability of implementing this algorithm to make systems more adaptable. Additional work on optimal urn sizes would also be beneficial, as this element of implementing the algorithm may be important to its success. While there may be no urn size that is optimal across a wide range of student groups or systems, better understanding how the urn sizes of people and items could be adjusted based on sample or system characteristics would be beneficial. Last, an evaluation of the algorithm that considers more than two time points would help better evaluate its performance. If, for example, another system uses IRT-based progress tests, and, thus, has criterion measurements across many time points, comparing the Urnings algorithm estimates to those many criterion estimates could provide a more comprehensive evaluation of the algorithm.

CHAPTER 6

CONCLUSION

The goal of this study was to evaluate the accuracy of the Urnings algorithm in tracing student growth trajectories within a digital learning platform using proficiency estimates from an IRT-based assessment as a criterion. The core research question of this study was, how well does the Urnings algorithm track student growth trajectories? In past research, the algorithm has done well in tracking student proficiency, but it has not been evaluated against a criterion measure of proficiency. If the algorithm performed well, it would support the use of the algorithm for making digital learning systems more adaptive—particularly for systems that have an IRT-based assessment component.

Based on results of this study, the Urnings algorithm does not seem to perform consistently well enough to be used for the aforementioned application. The threshold for performing well enough will depend on individual systems and the intended uses of estimates from the algorithm. Much like building a validity argument for the use of a score from an assessment, if the use of the algorithm estimate is relatively low stakes, then a degree of inconsistency in performance may be acceptable; however, the inconsistency found in this study is to such a degree that it would be worthwhile to investigate alternatives. For example, while progress testing may be intrusive, the accuracy and consistency of performance may be worth this cost. Progress testing could also be modified to be less intrusive. For example, if instruction and practice items exist within units of learning material, they could be psychometrically modeled. Proficiency could be estimated after lessons without the need for a separate quiz component. Bayesian estimation could also be used to carry forward information about proficiency estimates from past units of material.

Other algorithms could also be considered. A core advantage of the Urnings algorithm is it has mechanisms for preventing scale drift. Scale drift is an issue for the Elo Rating System and Glicko; however, this may not be an issue if items are modeled using conventional psychometric methods and are held constant across matches. This modification could also help the Urnings algorithm perform

better. How this adjustment might affect the known properties of these algorithms is unknown but could be investigated in future research.

In summary, this dissertation evaluated the Urnings algorithm against a criterion measure of student proficiency. Results suggest the Urnings algorithm may not be the best solution for making digital learning systems more adaptable. Additional research should be done on the Urnings algorithm, given its promising results in past research and the limited work thus far with the algorithm—but alternatives should also be investigated for helping make digital learning systems with IRT-based assessment components more adaptable.

APPENDIX

DESCRIPTIVE SUMMARIES OF SAMPLE

Table 1

Means and Standard Deviation for Diagnostic Score by Grade, Season, and Item Group

Grade	Season	10-49		50-99		100-149		150-199		200-250	
		M	SD	M	SD	M	SD	M	SD	M	SD
K	Fall	-5.26	0.87	-5.20	0.85	-5.19	0.82	-4.95	0.91	-4.82	0.83
	Winter	-4.72	0.98	-4.57	0.92	-4.47	0.89	-4.24	0.89	-4.08	0.85
	Spring	-4.27	1.01	-4.05	0.93	-3.90	0.87	-3.70	0.89	-3.44	0.87
1	Fall	-4.13	1.09	-4.11	1.07	-3.98	1.08	-3.91	0.98	-3.85	1.01
	Winter	-3.59	1.16	-3.57	1.13	-3.38	1.07	-3.29	1.05	-3.15	1.04
	Spring	-3.20	1.18	-3.15	1.20	-2.97	1.15	-2.81	1.07	-2.60	1.07
2	Fall	-3.08	1.22	-2.96	1.14	-2.84	1.30	-2.78	1.12	-2.67	1.07
	Winter	-2.55	1.23	-2.41	1.20	-2.23	1.29	-2.17	1.15	-2.05	1.05
	Spring	-2.17	1.26	-2.03	1.25	-1.79	1.30	-1.73	1.13	-1.60	1.07
3	Fall	-1.35	1.25	-1.89	1.12	-2.04	1.04	-2.06	0.95	-2.41	0.93
	Winter	-0.87	1.27	-1.37	1.16	-1.45	1.04	-1.47	0.96	-1.74	1.02
	Spring	-0.38	1.41	-0.91	1.28	-0.97	1.21	-0.96	1.04	-1.21	1.11
4	Fall	-0.59	0.89	-1.43	1.04	-2.13	0.95	-2.48	0.97	-2.62	0.97
	Winter	-0.13	1.00	-0.94	1.21	-1.70	1.05	-1.98	1.06	-2.11	1.03
	Spring	0.38	1.11	-0.40	1.30	-1.15	1.17	-1.52	1.17	-1.58	1.19
5	Fall	-0.60	0.82	-1.52	1.01	-2.09	1.00	-2.41	0.98	-2.61	0.90
	Winter	-0.19	0.87	-1.15	0.99	-1.70	1.01	-2.01	1.05	-2.20	0.95
	Spring	0.21	0.99	-0.77	1.18	-1.34	1.19	-1.64	1.22	-1.83	1.05

Note. Means and standard deviations are in logits.

Table 2

Means and Standard Deviations for Number of Items Responded to by Grade and Item Group

Grade	10-49		50-99		100-149		150-199		200-250	
	M	SD	M	SD	M	SD	M	SD	M	SD
K	38	9	83	13	127	14	175	15	227	15
1	38	9	83	13	128	13	177	15	226	14
2	40	8	82	13	128	14	178	14	226	14
3	39	8	80	14	128	14	176	14	224	15
4	37	9	75	15	125	14	175	15	224	14
5	35	9	76	14	126	14	174	14	224	14

Note. Values are rounded to the nearest integer.

Table 3

Means and Standard Deviations for Growth by Grade, Growth Period, and Item Group

Grade	Growth Period	10-49		50-99		100-149		150-199		200-250	
		M	SD	M	SD	M	SD	M	SD	M	SD
K	Fall to Winter	0.53	0.77	0.63	0.75	0.72	0.72	0.71	0.76	0.75	0.71
	Winter to Spring	0.46	0.82	0.52	0.74	0.56	0.74	0.53	0.75	0.64	0.68
	Fall to Spring	0.99	0.83	1.14	0.84	1.28	0.79	1.24	0.81	1.38	0.79
1	Fall to Winter	0.53	0.70	0.54	0.72	0.60	0.73	0.62	0.66	0.70	0.66
	Winter to Spring	0.40	0.74	0.42	0.74	0.40	0.71	0.48	0.68	0.54	0.69
	Fall to Spring	0.93	0.82	0.97	0.85	1.01	0.83	1.10	0.78	1.25	0.83
2	Fall to Winter	0.53	0.67	0.56	0.67	0.60	0.69	0.61	0.66	0.62	0.60
	Winter to Spring	0.38	0.64	0.37	0.61	0.44	0.61	0.44	0.58	0.45	0.61
	Fall to Spring	0.91	0.73	0.93	0.71	1.04	0.71	1.05	0.73	1.07	0.70
3	Fall to Winter	0.48	0.55	0.51	0.61	0.59	0.60	0.60	0.58	0.67	0.59
	Winter to Spring	0.50	0.59	0.46	0.59	0.48	0.58	0.50	0.57	0.53	0.63
	Fall to Spring	0.97	0.66	0.97	0.69	1.07	0.74	1.10	0.64	1.20	0.71
4	Fall to Winter	0.47	0.51	0.49	0.62	0.43	0.75	0.50	0.77	0.51	0.75
	Winter to Spring	0.51	0.52	0.53	0.59	0.54	0.74	0.46	0.72	0.53	0.67
	Fall to Spring	0.97	0.57	1.02	0.67	0.98	0.78	0.95	0.82	1.04	0.82
5	Fall to Winter	0.41	0.54	0.38	0.68	0.39	0.73	0.40	0.77	0.42	0.76
	Winter to Spring	0.40	0.60	0.38	0.66	0.37	0.68	0.37	0.81	0.37	0.73
	Fall to Spring	0.81	0.62	0.76	0.75	0.75	0.80	0.77	0.87	0.79	0.81

Note. Means and standard deviations are in logits.

REFERENCES

Abyaa, A., Idrissi, M. K., & Bennani, S. (10 2019). Learner modelling: systematic review of the literature from the last 5 years. *Educational Technology Research & Development*, *67*, 1105–1143.

Abbakumov, D., Desmet, P., & Van den Noortgate, W. (2 2019). Measuring growth in students' proficiency in MOOCs: Two component dynamic extensions for the Rasch model. *Behavior Research Methods*, *51*, 332–341. doi:10.3758/s13428-018-1129-1

Alabdulhadi, A., & Faisal, M. (3 2021). Systematic literature review of STEM self-study related ITSs. *Education and Information Technologies*, *26*, 1549. doi:10.1007/s10639-020-10315-z

Alamri, H. A., Watson, S., & Watson, W. (1 2021). Learning Technology Models that Support Personalization within Blended Learning Environments in Higher Education. *TechTrends: For Leaders in Education & Training*, *65*, 62. doi:10.1007/s11528-020-00530-3

Albadvi, A., & Shahbazi, M. (2009). A hybrid recommendation technique based on product category attributes. *Expert Systems with Applications*, *36*, 11480–11488.

De Ayala, R. J. (2013). *The theory and practice of item response theory*. Guilford Publications.

De Boeck, P., Bakker, M., Zwitser, R., Nivard, M., Hofman, A., Tuerlinckx, F., & Partchev, I. (2011). The estimation of item response models with the lmer function from the lme4 package in R. *Journal of Statistical Software*, 39, 1-28.

De Bra, P., & Calvi, L. (1998, June). AHA: a generic adaptive hypermedia system. In *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia*, 5-12

du Boulay, B. (2016). Artificial Intelligence as an Effective Classroom Assistant. *IEEE Intelligent Systems, Intelligent Systems, IEEE, IEEE Intell. Syst*, *31*(6), 76–81. https://doi.org/10.1109/MIS.2016.93

Baylari, A., & Montazer, G. A. (5 2009). Design a personalized e-learning system based on item response theory and artificial neural network approach. *Expert Systems with Applications*, *36*, 8013–8021. doi:10.1016/j.eswa.2008.10.080

Böcker, H. D., Hohl, H., & Schwab, T. (1990, August). Upsilon-pi-ADAPT-epsilon-rho: Individualizing hypertext. In Proceedings of the *IFIP TC13 Third Interational Conference on Human-Computer Interaction* (pp. 931-936).

Bolsinova, M., Maris, G., Hofman, A. D., van der Maas, H. L. J., & Brinkhuis, M. J. S. (1 2022). Urnings: A new method for tracking dynamically changing parameters in paired comparison systems. *Journal of the Royal Statistical Society. Series C: Applied Statistics*, *71*, 91–118. doi:10.1111/rssc.12523

Botsios, S., Georgiou, D., & Safouris, N. (2008). Contributions to adaptive educational hypermedia systems via on-line learning style estimation. *Journal of Educational Technology & Society*, 11(2), 322-339.

Boyle, C., & Teh, S. H. (1993, November). Multimedia intelligent documentation: Metadoc V. In *Proceedings of the 11th annual international conference on Systems documentation* (pp. 21-27).

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4), 324-345.

Brinkhuis, M. J., & Maris, G. (2010). Adaptive estimation: How to hit a moving target. *Measurement and Research Department Reports (Report No. 2010-1)*. Arhnem: Cito.

Brinkhuis, M. J. S., Savi, A. O., Hofman, A. D., Coomans, F., Van der Maas, H. L. J., & Maris, G. (8 2018). Learning As It Happens: A Decade of Analyzing and Shaping a Large-Scale Online Learning System. *Journal of Learning Analytics*, *5*. doi:10.18608/jla.2018.52.3

Brusilovsky, P. L. (1992). Intelligent tutor, environment and manual for introductory programming. *Educational & Training Technology International*, *29*(1), 26-34.

Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education. *technology*, *2*, 1.

Brusilovsky, P., & Eklund, J. (1998). A study of user model based link annotation in educational hypermedia. *J. Univers. Comput. Sci.*, *4*, 429–448.

Carver Jr, C. A., Hill, J. M., & Pooch, U. W. (1999). Third Generation Adaptive Hypermedia Systems. *WebNet*, 177-182.

Chen, C.-M., & Chung, C. J. (2008). Personalized mobile English vocabulary learning system based on item response theory and learning memory cycle. *Computers & Education*, *51*(2), 624-645.

Chen, C.-M., & Hsieh, Y. L. (2005, July). Mining learner profile utilizing association rule for common learning misconception diagnosis. In *Fifth IEEE International Conference on Advanced Learning Technologies*, 588-592

Chen, C.-M., & Hsu, S.-H. (2008). Personalized intelligent mobile learning system for supporting effective English learning. *Educational Technology & Society*, *11*(3), 153-180.

Chen, C.-M., Liu, C.-Y., & Chang, M.-H. (2006). Personalized curriculum sequencing utilizing modified item response theory for web-based instruction. *Expert Systems with Applications*, *30(2)*, 378–396.

Chrysafiadi, K., & Virvou, M. (2013). Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, *40*(11), 4715-4729.

Conati, C., Gertner, A. S., VanLehn, K., & Druzdzel, M. J. (1997). On-line student modeling for coached problem solving using Bayesian networks. *User modeling* 231–242.

Corbett, A. T., & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, *4*, 253–278.

Curriculum Associates. (2017). *i-Ready assessments technical manual.*

Desmarais, M. C., & d Baker, R. S. J. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, *22*, 9–38.

Qing Ding, & Sitan Cao. (2017). RECT: A Cloud-Based Learning Tool for Graduate Software Engineering Practice Courses With Remote Tutor Support. *IEEE Access*, 5, 2262–2271. https://doi.org/10.1109/ACCESS.2017.2664070

Drigas, A. S., Argyri, K., & Vrettaros, J. (2009, September). Decade review (1999-2009): artificial intelligence techniques in student modeling. In *World Summit on Knowledge Society* 552-564. Springer, Berlin, Heidelberg.

Elo, A. E. (1978). *The rating of chessplayers, past and present*. Arco Pub.

Galvez, J., Guzman, E., Conejo, R., Mitrovic, A., & Mathews, M. (4 2016). Data calibration for statistical-based assessment in constraint-based tutors. *Knowledge-Based Systems*, *97*, 11-23. doi:10.1016/j.knosys.2016.01.024

Glickman, M. E. (1999). Parameter Estimation in Large Dynamic Paired Comparison Experiments. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 48(3), 377–394.

Glickman, M. E. (2001). Dynamic paired comparison models with stochastic variances. *Journal of Applied Statistics*, *28*, 673–689. doi:10.1080/02664760120059219

Glickman, M. E. (2022). *Example of the Glicko-2 system*. http://www.glicko.net/glicko/glicko2.pdf

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goel, G., Lallé, S., & Luengo, V. (2012). Fuzzy logic representation for student modelling. *International Conference on Intelligent Tutoring Systems*, 428–433.

Goldowsky, H. (2006). A Conversation with Mark Glickman. *uchess.org*. http://www.uschess.org/index.php/October/A-Conversation-with-Mark-Glickman.html

Graf, S., Ting-Wen Chang, Kersebaum, A., Rath, T., & Kurcz, J. (2014). Investigating the Effectiveness of an Advanced Adaptive Mechanism for Considering Learning Styles in Learning Management Systems. *2014 IEEE 14th International Conference on Advanced Learning Technologies, Advanced Learning Technologies (ICALT)*, 112–116. https://doi.org/10.1109/ICALT.2014.41

Guerra, J., Huang, Y., Hosseini, R., & Brusilovsky, P. (2015, January). Graph analysis of student model networks. *CEUR Workshop Proceedings* (Vol. 1446). University of Pittsburgh.

Herbrich, R., Minka, T., & Graepel, T. (2006). TrueSkill™: a Bayesian skill rating system. *Advances in neural information processing systems*, *19*.

Hofman, A. D., Brinkhuis, M. J., Bolsinova, M., Klaiber, J., Maris, G., & van der Maas, H. L. (2020). Tracking with (un) certainty. *Journal of Intelligence*, *8*(1), 10.

Hosseini, S. A., Tawil, A.-R. H., Jahankhani, H., & Yarandi, M. (5 2013). Towards an Ontological Learners' Modelling Approach for Personalised E-Learning. *International Journal of Emerging Technologies in Learning*, *8*, 4–10.

Huang, S. L., & Shiu, J. H. (2012). A user-centric adaptive learning system for e-learning 2.0. *Journal of Educational Technology & Society*, *15*(3), 214-225.

IXL Learning (2020). Validation Study of the IXL Real-Time Diagnostic using MAP Growth Assessments. https://www.ixl.com/research/IXL-Real-Time-Diagnostic-Validation-Study.pdf

Jackson, T., Mathews, E., Lin, K. I., Olney, A., & Graesser, A. (2003, June). Modeling student performance to enhance the pedagogy of autotutor. *International Conference on user modeling*, 368-372. Springer, Berlin, Heidelberg.

Johnson, L., Adams Becker, S., Estrada, V., and Freeman, A. (2015). NMC Horizon Report: 2015 Higher Education Edition.

Kadengye, D. T., Ceulemans, E., & den Noortgate, W. (2014). A generalized longitudinal mixture IRT model for measuring differential growth in learning environments. *Behavior research methods*, *46*, 823–840.

Kadengye, D. T., Ceulemans, E., & Van Den Noortgate, W. (2015). Modeling growth in electronic learning environments using a longitudinal random item response model. *The Journal of Experimental Education*, *83*, 175–202.

Klašnja-Milićević, A., Ivanović, M., & Nanopoulos, A. (2015). Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. *Artificial Intelligence Review*, *44*(4), 571-604.

Klinkenberg, S., Straatemeier, M., & van der Maas, H. L. (2011). Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education, 57*(2), 1813-1824.

Leung, E. W. C., & Li, Q. (2007). An experimental study of a personalized learning environment through open-source software tools. *IEEE Transactions on Education, 50*(4), 331-337.

Liu, Z., & Wang, H. (2007, April). A modeling method based on bayesian networks in intelligent tutoring system. *2007 11th International Conference on Computer Supported Cooperative Work in Design*, 967-972.

Martin, B. (1999). Constraint-Based Student Modeling: Representing Student Knowledge. *New Zealand Computer Science Research Students' Conference*, 22-29.

Martins, A. C., Carrapatoso, E., Faria, L., & de Carvalho, C. V. (2008). User modeling in adaptive hypermedia educational systems.

Mayo, M., & Mitrovic, A. (2001). Optimising ITS behaviour with Bayesian networks and decision theory.

Minka, T., Research, M., Cleven, R., & Zaykov, Y. (2018). TrueSkill 2: An improved Bayesian skill rating system. *Technical Report*.

Mitrovic, A., Mayo, M., Suraweera, P., & Martin, B. (2001, June). Constraint-based tutors: a success story. *International conference on industrial, engineering and other applications of applied intelligent systems*, 931-940. Springer, Berlin, Heidelberg.

Nakic, J., Granic, A., & Glavinic, V. (1 2015). Anatomy of student models in adaptive learning systems: A systematic literature review of individual differences from 2001 to 2013. *Journal of Educational Computing Research, 51*, 459–489. doi:10.2190/EC.51.4.e

Normadhi, N. B. A., Shuib, L., Nasir, H. N. M., Bimba, A., Idris, N., & Balakrishnan, V. (2019). Identification of personal traits in adaptive learning environment: *Systematic literature review*. *Computers & Education, 130*, 168-190.

Nižnan, J., Pelánek, R., & Rihák, J. (2015). Student Models for Prior Knowledge Estimation. *International Educational Data Mining Society*.

Noy, N. F., & McGuinness, D. L. (2001). Ontology development 101: A guide to creating your first ontology. *User Modeling and User-Adapted Interaction*, *1*, 203–258.

Papadimitriou, A., & Gyftodimos, G. (2017). The Role of Learner Characteristics in the Adaptive Educational Hypermedia Systems: The Case of the MATHEMA. *International Journal of Modern Education & Computer Science, 9*(10).

Papousek, J., Pelánek, R., & Stanislav, V. (2014, July). Adaptive practice of facts in domains with varied prior knowledge. *Educational Data Mining 2014*.

Park, J. (2021). Online Estimation of Student Ability and Item Difficulty with Glicko-2 Rating System on Stratified Data. *International Educational Data Mining Society*.

Park, J. Y., Cornillie, F., Van der Maas, H. L., & Van Den Noortgate, W. (2019). A multidimensional IRT approach for dynamically monitoring ability growth in computerized practice environments. *Frontiers in psychology*, *10*, 620.

Paravati, G., Lamberti, F., Gatteschi, V., Demartini, C., & Montuschi, P. (2016). Point cloud-based automatic assessment of 3D computer animation courseworks. *IEEE Transactions on Learning Technologies*, *10*(4), 532-543.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann.

Pelánek, R. (2016). Applications of the Elo rating system in adaptive educational systems. *Computers & Education*, *98*, 169-179.

Pelánek, R., & Effenberger, T. (2020). Beyond binary correctness: Classification of students' answers in learning systems. *User Modeling and User-Adapted Interaction*, *30*(5), 867-893.

Pelánek, R., & Jarušek, P. (2015). Student modeling based on problem solving times. *International Journal of Artificial Intelligence in Education*, *25*(4), 493-519.

Rasch, G. (1960). Probabilistic models for some intelligence and attainment tests.

Reading Plus (2021). Theoretical Framework and Foundational Research. https://go.dreambox.com/rs/715-ORW-647/images/theoretical-framework-foundational-research.pdf

Reddick, R. (2019). Using a Glicko-Based Algorithm to Measure In-Course Learning. *International Educational Data Mining Society*.

Rich, E. (1979). User modeling via stereotypes. *Cognitive science*, *3*, 329–354.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, *61*, 85-117.

Shute, Valeris, & Towle, B. (2003). Adaptive e-learning. *Educational psychologist*, *38*, 105–114.

Shute, V. J., & Zapata-Rivera, D. (2007). Adaptive technologies. *ETS Research Report Series*, 2007(1), i-34.

Stansfield, J. L., Carr, B. P., & Goldstein, I. P. (1976). Wumpus advisor 1: A first implementation program that tutors logical and probabilistic reasoning skills.

Sturgis, C., Patrick, S., & Pittenger, L. (2011). It's Not a Matter of Time: Highlights from the 2011 Competency-Based Summit. *International association for K-12 online learning*.

Tennyson, R. D. (1975). Adaptive instructional models for concept acquisition. *Educational Technology*, *15*(4), 7–15.

Tennyson, R. D. (1993). MAIS: A computer-based integrated instructional system. Behavior *Research Methods, Instruments, & Computers*, *25*(2), 93–100.

Tennyson, R. D., & Rothen, W. (1977). Pretask and on-task adaptive design strategies for selecting number of instances in concept acquisition. *Journal of Educational Psychology*, *69*, 586.

Tsiriga, V., & Virvou, M. (2003, July). Initializing student models in web-based ITSs: a generic approach. *Proceedings 3rd IEEE International Conference on Advanced Technologies*, 42-46.

U.S. Department of Education (2017). Reimagining the Role of Technology in Education: 2017 National Education Technology Plan Update. https://tech.ed.gov/files/2017/01/NETP17.pdf

Vagale, V., & Niedrite, L. (2012, July). Learner Model's Utilization in the E-Learning Environments. *DB&Local Proceedings*, 162-174.

Van den Noortgate, W., De Boeck, P., & Meulders, M. (2003). Cross-classification multilevel logistic models in psychometrics. *Journal of Educational and Behavioral Statistics*, *28(*4), 369-386.

Vandewaetere, M., Desmet, P., & Clarebout, G. (2011). The contribution of learner characteristics in the development of computer-based adaptive learning environments. *Computers in Human Behavior*, *27*, 118–130.

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational psychologist*, *46*(4), 197-221.

Vélez, J., Fabregat, R., Nassiff, S., Petro, J., & Fernandez, A. (2008, November). User integral model in adaptive virtual learning environment. *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, 3275-3284. Association for the Advancement of Computing in Education (AACE).

Verguts, T., & Boeck, P. D. (2000). A Rasch model for detecting learning while solving an intelligence test. *Applied Psychological Measurement*, *24*, 151–162.

Vištica, M., Grubišic, A., & Žitko, B. (2016). Applying graph sampling methods on student model initialization in intelligent tutoring systems. *The International Journal of Information and Learning Technology*. Emerald Group Publishing Limited.

Wauters, K., Desmet, P., & Van Den Noortgate, W. (2010). Adaptive item-based learning environments based on the item response theory: Possibilities and challenges. *Journal of Computer Assisted Learning*, *26*(6), 549-562.

Wauters, K., Desmet, P., & Van Noortgate, W. (2010, June). Monitoring learners' proficiency: weight adaptation in the elo rating system. *Educational Data Mining* 2011.

Webb, G. I., Pazzani, M. J., & Billsus, D. (2001). Machine learning for user modeling. *User modeling and user-adapted interaction*, *11*, 19–29.

Wolf, M. A. (2010). Innovate to educate: System [re] design for personalized learning (pp. 21). *Partoyan, E., Schneiderman, & Seltz, J.(Eds.)*, A Report from the 2010 Symposium. ACSD. Retrieved July (Vol. 28, p. 2014).

Zhang, X., & Han, H. (2005). An empirical testing of user stereotypes of information retrieval systems. *Information processing & management*, *41*, 651–664.