Bridging the Physical and the Digital Worlds of Learning Analytics in Educational Assessments through Human-AI Collaboration

by

Yancy Vance Paredes

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved January 2023 by the
Graduate Supervisory Committee:

I-Han Hsiao, Co-Chair
Kurt VanLehn, Co-Chair
Scotty D. Craig
Srividya Bansal
Hasan Davulcu

ARIZONA STATE UNIVERSITY

May 2023

ABSTRACT

Experience, whether personal or vicarious, plays an influential role in shaping human knowledge. Through these experiences, one develops an understanding of the world, which leads to learning. The process of gaining knowledge in higher education transcends beyond the passive transmission of knowledge from an expert to a novice. Instead, students are encouraged to actively engage in every learning opportunity to achieve mastery in their chosen field. Evaluation of such mastery typically entails using educational assessments that provide objective measures to determine whether the student has mastered what is required of them. With the proliferation of educational technology in the modern classroom, information about students is being collected at an unprecedented rate, covering demographic, performance, and behavioral data. In the absence of analytics expertise, stakeholders may miss out on valuable insights that can guide future instructional interventions, especially in helping students understand their strengths and weaknesses. This dissertation presents Web-Programming Grading Assistant (WebPGA), a homegrown educational technology designed based on various learning sciences principles, which has been used by 6,000+ students. In addition to streamlining and improving the grading process, it encourages students to reflect on their performance. WebPGA integrates learning analytics into educational assessments using students' physical and digital footprints. A series of classroom studies is presented demonstrating the use of learning analytics and assessment data to make students aware of their misconceptions. It aims to develop ways for students to learn from previous mistakes made by themselves or by others. The key findings of this dissertation include the identification of effective strategies of better-performing students, the demonstration of the importance of individualized guidance during the reviewing process, and the likely impact of validating one's understanding of

another's experiences. Moreover, the Personalized Recommender of Items to Master and Evaluate (PRIME) framework is introduced. It is a novel and intelligent approach for diagnosing one's domain mastery and providing tailored learning opportunities by allowing students to observe others' mistakes. Thus, this dissertation lays the groundwork for further improvement and inspires better use of available data to improve the quality of educational assessments that will benefit both students and teachers.

I dedicate this work to all the teachers who unceasingly endeavor to look for innovative ways to bring out the best in their students despite how teaching is often perceived as a thankless profession. Indeed, we are still at the forefront in the quest to seek out ways to make lasting impacts on students—our future generation. I also dedicate this work to all students who continually strive to be the best that they can be. To all my former, current, and future students, know that I continue to aspire to be the best that I can be to provide you with quality education.

ACKNOWLEDGMENTS

Words are not enough to express my gratitude for the different people who have helped me along the way. This work is a culmination of my doctoral journey which has spanned a third of my life as of this writing. Despite the time it took for me to finish it, I am still proud of all the things I have learned throughout.

I would like to acknowledge Dr. Sharon I-Han Hsiao who believed in me and gave me the opportunity to take the lead on one of her projects. Providing me with wide latitude in navigating through my professional development along with continuous guidance and mentorship over the years has helped me become the independent researcher that I am today. The same goes for Dr. Scotty D. Craig, for letting me join his research lab where I was introduced to cognitive science and gained insights into topics outside my field of expertise. I also want to acknowledge my committee co-chair, Dr. Kurt VanLehn, and members, Dr. Srividya Bansal and Dr. Hasan Davulcu, for sharing their invaluable time and expertise in this project. I am grateful for the challenging and stimulating questions throughout the dissertation process, which helped me develop my critical thinking skills.

I would like to recognize all the teachers who have used our system and who have paved the way to making the necessary improvements, especially Dr. Jennifer Yi-Ling Lin and Dr. Mutsumi Nakamura.

To my colleagues who have always provided me with feedback, Cesar and Michelle, know that I appreciate your patience and time in listening to my random academic ramblings whenever I doubt my ideas.

I would like to express my gratitude to all my teachers way back from kindergarten until graduate school. The patience and the various efforts you have shown in your passion have truly served as an inspiration to me. The same goes for all my students

in my capacity both as an instructor and a teaching assistant for you have motivated me to truly find ways to help you.

My family and friends have been with me since the beginning. For those who understood the various sacrifices I had to make to accomplish this degree, thank you. To Corinne and Florenz, I truly appreciate the support. To my very first teachers, my parents, I am forever grateful.

I also would like to take this opportunity to recognize the support I received from the ASU Graduate & Professional Student Association, the ASU Graduate College, and the Google Cloud Research Credits Program. Also, the various departments and research labs that believed in me, allowed me to learn across interdisciplinary fields, and helped me survive graduate school without hurting my finances: the School of Computing and Augmented Intelligence, formerly the School of Computing, Informatics, and Decision Systems Engineering; ASU Learning Enterprise Inspark; the ASU Advanced Distributed Learning Partnership Lab led by Dr. Scotty D. Craig; and the Global Security Initiative, Center for Human, Artificial Intelligence, and Robot Teaming Lab led by Dr. Nancy Cooke.

Finally, all these things would not have been possible without the strength, wisdom, and endurance given by the Heavenly Father. All honor and praise to Him!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

λ	overall student performance based on the normalized average performance of a student in the semester; value ranges from 0 to 1
ω	refers to an ideal student (i.e., obtains the maximum possible in every opportunity)
θ	estimated student latent trait
π	time point; value ranges from 0 to 1
π_e	latest time point with evidence of student mastery
π_q	time point of which the system is being queried to perform the forecasting
ϕ	ratio of the maximum points that can be obtained in a question and the number of knowledge components (KCs) associated to a question
$\delta(\pi)$	cumulative mastery level at time point π; value ranges from 0 to 1
γ	refers to a next term student
τ	refers to a single topic
$\Delta\delta_\tau$	normalized gain on mastery level between two time points π for topic τ
q	number of questions
t	number of topics
$\mathbf{L}$	the student model library
$\mathbf{X}$	design matrix used for modeling with dimension $(1 \times (t+1))$; combination of one-hot encoding of the topic and the time point π
$\beta(\mathbf{X})$	the unified student model given a design matrix $\mathbf{X}$
$\mathbf{X}'$, $\mathbf{Y}$	refers to the input and output values for the modeling
$\mathbf{R}$	growth rates for each t topics with dimension $(1 \times t)$

$\mathbf{M}$	cumulative mastery level for each t topics with dimension $(1 \times t)$
$\mathbf{P}$	performance matrix with dimension $(q \times 1)$; the normalized scores of a student s on a list of q questions
$\mathbf{Q}'$	raw Q matrix with dimension $(q \times t)$; the weight associated to each question (i.e., maximum points that can be obtained for a topic associated to a question)
$\mathbf{Q}$	normalized Q matrix where each row of $\mathbf{Q}'$ is divided by its row sum
$\mathbf{W}$	topic performance weights with dimension $(1 \times t)$; the multiplier values used to compute for the forecasted performance of a student
$\mathbf{I}$	question representation with dimension $(1 \times (t+1))$; a question or item profile containing the normalized distribution of the t topics along with ϕ (the scaled point per KC)
$\mathbf{S}_\pi$	student representation with dimension $(1 \times t)$; student mastery profile containing the estimated mastery level of the student s at time point π for all the t topics
$\mathrm{QD}\,(a, b)$	Euclidean distance metric between two questions a and b
$\mathrm{QT}\,(a, b)$	metric representing the temporal proximity of b w.r.t. a
$\mathrm{QS}\,(a, b)$	non symmetric indicator to determine the suitability of recommending question b w.r.t. a, given the topic coverage of both questions
$\mathrm{QR}\,(a, b)$	non symmetric weighted score to indicate the relevance of question b w.r.t a
$\mathrm{CP}\,(s, \tau, \pi)$	cumulative maximum total points that can be obtained up until π of student s for topic τ
$\mathrm{TP}\,(\tau)$	estimated maximum total points that can be obtained for the entire semester for topic τ by a student, similar to $\mathrm{CP}\,(\omega, \tau, 1)$

$\mathrm{CM}(s, \tau, \pi)$	cumulative mastery level for topic τ at time point π of student s
$\mathrm{PE}(\mathbf{Q}', s, \tau, \pi)$	raw points earned by student s for topic τ in a test with given raw $\mathbf{Q}'$ matrix administered during π

PREFACE

Only a fool learns from his own mistakes.
The wise man learns from the mistakes of others.
Otto von Bismarck

Learn from the mistakes of others.
You can't live long enough to make them all yourself.
Eleanor Roosevelt

I have always enjoyed reading the preface of dissertations that I have seen in the past. So I guess it is my turn to make one. My mind wandered as I wrote this page while sitting in the same chair I have had in this research lab for the last six years. I recall how when I was in elementary school, I always dreaded studying history. The problem persisted throughout my undergraduate studies. Oh, how tedious it was to memorize key dates and the names of notable historical figures. It was horrible for me. Back then, I knew that I would pursue a career in computer science[1], so why should I have to memorize these details again? For example, the long name of one of the greatest heroes of my country[2]. At some point, I am sure you, just like any fellow student at my age, complained: the past is the past! *Why should we study history?*

As it turned out, a Google search result[3] revealed a very helpful answer. Studying history allows us to gain a more in-depth understanding of how events from the past

[1]Plot twist: More of in the field of Education.

[2]The actual name is left as an exercise for the reader.

[3]Yes, how academic of me to ambiguously cite something vaguely.

influence the present. In addition, we learn why things are the way they are today. Nevertheless, what stood out most was learning from history will help us to avoid making the same mistakes in the future. Put simply, *we learn from their mistakes!*

Then it occurred to me, perhaps it is beyond mere recall of facts. Instead, it is about learning how to make sense of the world around us. This led me to think that perhaps we are the products or collections of our own and other people's experiences. Moreover, these experiences need not be limited to success. They may have been failures. As a matter of fact, it is often these failures that have the most profound influence on our lives. These experiences were either recorded in diaries, scrolls, or other forms of record keeping. In today's digital world, it is more convenient than ever to record and access anything at the click of a button. So, why do I place such a high value on learning from mistakes? Well, it reduces the number of *unknown unknowns* for a person. There are just so many mistakes that may be made that even one lifetime would not be enough.

It is my intention to present in this work my own perspective on how one can make sense of *various experiences* in the modern classroom, particularly those relating to Computer Science, through WebPGA. It is an educational technology I have developed with the guidance of my mentor and have worked on since day one of graduate school. I am proud of how it has evolved over the years.

Chapter 1

INTRODUCTION

Today's knowledge-based economy requires that one be knowledgeable in their field to gain a competitive advantage. Learning is an essential part of gaining knowledge. There is more to learning than passively receiving information from one individual to another. Learning is a process that one must undertake on their own and is the product of how one interprets their experiences (Ambrose et al., 2010). Actively constructing knowledge entails integrating newly acquired information gained from experience with previous knowledge (Piaget & Inhelder, 1969; Vygotsky, 1978). Since learning is a mental process, observable artifacts such as performance or product are used to evaluate one's learning.

Throughout this dissertation, computer programming is the primary focus. The process of learning how to program requires the acquisition of knowledge that extends beyond the programming language alone (Linn & Dalbey, 1985; Robins, 2019). Students who attend introductory programming courses typically have an extreme bimodal distribution of abilities—those who can program and those who cannot. Robins (2010) proposed that this occurs due to the learning edge momentum hypothesis, which suggests that a person's learning outcomes become self-reinforcing over time. Additionally, he asserts that learning one concept makes it easier to grasp others that are closely related. A similar argument can be made that mastery learning is relevant to computer programming since well-understood fundamental concepts are essential for success in advanced ones. For this reason, it is essential to resolve misconceptions as soon as possible.

Identifying and resolving misconceptions requires reflection on one's performance or outcome. Reflection refers to the learner's response to an experience that prompts them to relive, rethink and evaluate it (Boud et al., 1985). Additionally, this is the self-reflection phase of Zimmerman's (1998) self-regulated learning model (SRL), where one's strategic actions are informed by the outcomes. Furthermore, it is believed that one's experience has a significant impact on one's self-efficacy or confidence in accomplishing a task (Bandura, 1997). Indeed, as the popular saying states, "experience is the best teacher". Nonetheless, as alluded to in the preface, experience does not have to be one's own. Alternatively, it may come from someone else, which is known as vicarious experience (Craig et al., 2009). Observational learning occurs when people observe how others behave (Bandura, 1977). In particular, relevant experiences of what Vygotsky (1978) referred to as "more knowledgeable others" can facilitate the learning process.

Modern technology has enabled researchers to conduct evidence-based investigations using all kinds of approaches, such as learning analytics and data-driven artificial intelligence, to gain a deeper understanding of how these learning experiences influence students' learning. Unsurprisingly, this became one of the many motivations for developing educational technologies that support students. With the advancement of technology and the advancement of computing power, artificial intelligence and machine learning techniques are among the many ways in which these technologies can benefit students. Providing students with opportunities to demonstrate their understanding of a domain along with guidance will enable them to become knowledgeable. One example of these technologies is intelligent tutoring systems (ITS). This technology has allowed for the generation and capture of more heterogeneous student data than ever before. These data can range from interactions

to performance and are generated and captured at an unprecedented rate. Using these data, stakeholders may be able to gain insights into students' behaviors, enabling them to make relevant, actionable, and informed interventions, which is the ethos of learning analytics. Despite the numerous benefits these technologies can provide to students, deploying these in blended classrooms remains a significant challenge, especially in developing countries. It is for this reason that many teachers still use traditional assessments in their classrooms to evaluate student performance.

1.1 Educational Assessments

In higher education, assessments, also known as exams or tests, are vital for evaluating students' progress. Moreover, it guides the learning process of the student (Sheard et al., 2013). Assessments consist of items or questions that students answer to demonstrate their understanding of predetermined concepts within a particular domain. Tests are among the many tools that teachers use to collect data about their students' strengths and weaknesses (Hanna & Dettmer, 2004). In addition, it allows them to evaluate the effectiveness of their instructional strategies. Typically, assessments are administered in two different ways in the classroom, each of which has advantages and disadvantages. In the traditional approach, pen and paper are used, while in the computer-based approach, computers are used. In determining which approach should be used in a classroom, several factors are taken into account; some are beyond a teacher's control. Due to their flexibility and simplicity, paper-based tests remain popular among teachers. Moreover, these tests require in-person proctoring and physical presence, which helps deter academic dishonesty. However, it also has some disadvantages. Grading these papers takes a significant amount of time.

Additionally, providing meaningful, personalized feedback to students while ensuring consistency between and within graders requires considerable time and effort. In the classroom, there are two types of assessment that are most commonly used, which are formative assessments and summative assessments. The former pertains to low-stakes tests and are typically not graded (e.g., practice quizzes). It is generally referred to as "assessment for learning" because it serves primarily the purpose of providing feedback on students' performance for students to diagnose and monitor their deficiencies. In contrast, the latter refers to high-stakes tests that evaluate students' learning, which are commonly called "assessments of learning". The results of these assessments are often used to make decisions that have a profound impact on students. It is therefore necessary to place importance on reviewing one's performance. Recent years have seen an increase in interest in shifting toward the notion of "assessment as learning" in which students take responsibility for their own learning and improvement (Earl, 2013). It emphasizes the importance of reflection and self-evaluation during the learning process.

1.2 Learning from One's Experience

By reviewing their previous performance, students can gain knowledge from their experience. It is hoped that by learning from this experience, one will be able to avoid making the same mistakes in the future. As part of the assessment process, students attempt a task and are given feedback on their performance. Providing feedback to students is one of the most effective methods of enhancing their learning (Hattie & Timperley, 2007). Given the trend of gradually replacing traditional classrooms with technologically enhanced classrooms, such as smart classrooms or online classes, it is

imperative that blended classes be upgraded to meet the demands of the future. In recent years, a number of tools have been developed to provide personalized feedback to students. Nevertheless, these systems can only be beneficial when they are used to conduct interactions through digital platforms. Since paper-based assessment remains a dominant evaluation method, particularly in large blended-instruction classes, solely using electronic educational systems reveals the gap between the physical and digital worlds. Students may be able to obtain valuable feedback on these graded papers that can help improve their performance as well as point out misconceptions. However, it is difficult to obtain empirical evidence regarding whether students review these papers or what their reviewing strategies are. This motivated the design and development of a new educational technology to facilitate the digitization, grading, and distribution of paper-based assessments in blended-learning classes. This technology allows for the easy capture of a wide variety of learning analytics. In Chapter 2, the research platform Web-based Programming Grading Assistant (WebPGA) is introduced. It also presents the results of a retrospective analysis conducted using the platform to analyze the behavior of students in an Object-Oriented Programming and Data Structures class at a large public university. Using the digital footprints of students, behavioral differences and associated learning impacts were examined, specifically to answer the following research questions:

RQ A.1: In terms of monitoring and reviewing, are there any behavioral differences between high-achieving and low-achieving students?

RQ A.2: Are there any differences in the behavior of students when grouped according to performance trajectories (i.e., whether the student's score in a subsequent examination improved relative to a prior one)?

RQ A.3: What reviewing behaviors are associated with learning?

RQ A.4: How does personalized guidance affect the behavior of students when reviewing?

The results indicated that students made significant efforts to review their test results. The high-achieving students and those who improved spent more time reviewing their mistakes and began doing so as soon as they received the results. While reviewing graded tests allows students to develop their metacognitive skills, the absence of adequate guidance exacerbated by a lack of maturity prevents students from fully utilizing the benefits of learning from past mistakes. Chapter 3 presents a preliminary analysis of the clickstream data of students from an Introduction to Computing course as part of the effort to uncover students' reviewing strategies. Unlike the earlier study, the undertaken study took into account the temporal aspect of the data. To account for the sequential nature of the data, Hidden Markov models (HMMs) were used to model the reviewing behaviors of high-performing and low-performing students, specifically to answer the following research questions:

RQ B.1: Do students review questions based on their performance?

RQ B.2: What reviewing patterns can be uncovered?

The results of the study indicate that the two groups used similar strategies, but also employed strategies specific to each group. Reviewing frequently is an important study habit that helps students to become aware of their strengths and weaknesses (Mehrens & Lehmann, 1991). As a whole, Chapters 2 and 3 emphasize the diversity of the characteristics of students and the importance of providing tailored guidance to students in the reviewing process. In addition, by receiving guidance, students could identify items where they had misconceptions. By doing so, they are encouraged to learn from their own experiences.

1.3 Learning from Other's Experience

Learning from experience does not have to be confined to one's own. In fact, one can gain valuable knowledge from other people's experiences, or in this case, their mistakes. Considering the vast amount of data WebPGA has collected over the years, not only can it provide valuable insights to teachers, but it can also be used to benefit future students. In the educational data mining literature, one of the most common applications is predicting a new cohort's performance (Paredes et al., 2020). As an alternative to this view, one could consider this data a collection of the experiences of past students, which include their strategies and failures. As a result of observing these *experiences*, future students will be able to learn and imitate desirable *behaviors* based on the resulting *consequences*, a hallmark of Bandura's (1977) social learning theory. In essence, the answers of students can be viewed as *worked examples* of which have been extensively examined in the literature, including their effectiveness in improving student learning (Atkinson et al., 2000; Chi et al., 1989; Sweller & Cooper, 1985) and their important role in the initial acquisition of cognitive skills of learners (Renkl, 2002, 2014; VanLehn, 1996). Additionally, several of these examples would be erroneous because these answers encompass both correct and incorrect responses. Using erroneous examples involves intentionally including errors so that students can explain and correct the errors (Booth et al., 2013; Große & Renkl, 2007). A number of prior studies, mostly from the mathematics field, have explored the potential benefits of using this approach in blended learning environments. Because of the Covid-19 pandemic, most learning activities have been conducted online, making it even more significant to study how students use WebPGA. An exploratory study is presented in Chapter 4 in which students from a synchronous online Computer Informatics

class were asked to evaluate varying degrees of erroneous examples. An activity was designed to assist students in addressing their misconceptions and preparing for an upcoming test by offering them a learning opportunity that closely resembles the program debugging process, specifically to answer the following research questions:

RQ C.1: Do students learn from evaluating erroneous answers?

RQ C.2: Do students leverage feedback provided to them during the learning activity?

RQ C.3: What behaviors do students exhibit when evaluating erroneous answers?

RQ C.4: How do students benefit from receiving feedback during the learning activity?

Students were provided with feedback via the grading process by providing them with the actual marks and comments related to the examples. The amount of time students spent on the activity and the difference between their assigned grade and a subject expert's grade were examined. While it is unclear whether students in this study learned from exposure to erroneous examples, it was found that students who were proactive in seeking feedback had better midterm scores. This emphasizes the importance of feedback in the learning process. This type of supplementary resource can prove beneficial if it is appropriate for the students' needs, as has been widely acknowledged in the literature on adaptive educational systems (Brusilovsky, 1998, 2001). It is therefore necessary to identify a past *experience* that is relevant. Knowing what is needed and what is relevant is essential to achieving this goal. Thus, if it is possible to predict a student's performance on a forthcoming test, these relevant items may be identified. The Personalized Recommender of Items to Master and Evaluate (PRIME) Framework is introduced in Chapter 5. The purpose of this framework is to

provide a principled approach to enable WebPGA to intelligently describe to students their deficiencies through erroneous examples that can be used as proxies, specifically to answer the following research questions:

RQ D.1: Using performance data on complex multi-topic test items from a classroom setting, how can the growth of the mastery level of students be modeled?

RQ D.2: How can a student's outcomes on individual items be predicted on a test that contains items that allow partial credit?

RQ D.3: Having knowledge of the potential outcomes of a test, what innovative learning opportunities, particularly in the domain of computer programming, can be provided to students to assist them in preparing?

Performance associated with these proxies closely resembles that of the student for whom they are recommended. This builds upon the idea of knowledge-gap-based remedial recommendations with the goal of providing resources to fill these gaps (Bauman & Tuzhilin, 2018; Thaker et al., 2020). Together, Chapters 4 and 5 demonstrate a pragmatic approach to leveraging existing data, thereby allowing students to gain insight from the experiences of others. Furthermore, it enhances the functionality and utility of the system without requiring the teacher to exert additional effort.

1.4 Data-Driven Educational Assessment Ecosystem

Over the past six years, WebPGA has been used by more than 6000 students from two universities, resulting in a 40% reduction in grading turnaround times. As a result

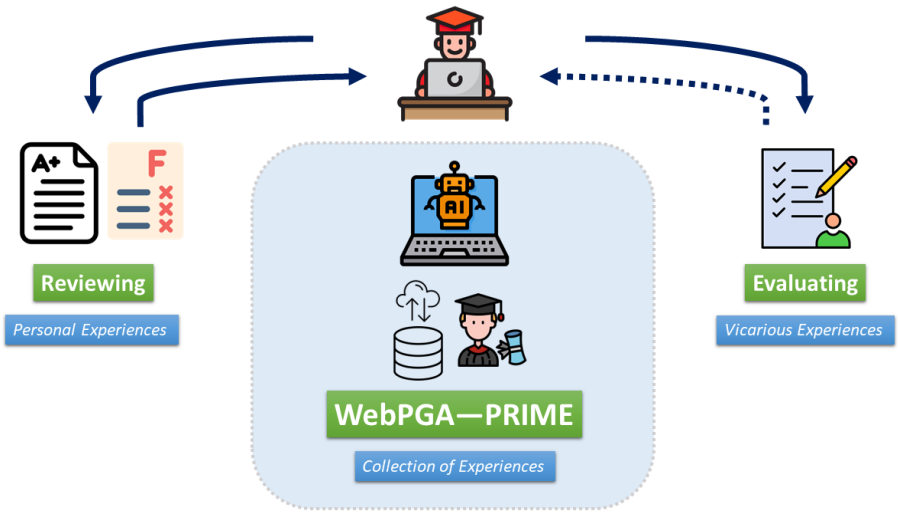


Figure 1. Overview of the WebPGA Ecosystem

of its integration with the physical world of educational assessments, learning analytics was brought from the digital realm to the physical realm. An overview of the various components is illustrated in Figure 1. This dissertation represents a comprehensive collection of the various endeavors undertaken to understand how to help students learn from their own as well as others' mistakes. **Using learning analytics and assessment data to make students cognizant of their misconceptions** is the central theme of this dissertation. Thus, students can only attain a deep understanding of the domain by recognizing their weaknesses and working diligently to improve them. Each chapter describes the studies conducted that shaped subsequent investigations and system improvements based on human-centered design principles[1]. Data from these investigations encompassed both behavioral and performance aspects. Finally, this dissertation concludes with a summary of all the findings and recommendations for future research.

[1]See Norman and Draper (1986) for a detailed discussion of the four principles of human-centered design.

Chapter 2

WEBPGA: AN EDUCATIONAL TECHNOLOGY THAT SUPPORTS LEARNING BY REVIEWING PAPER-BASED PROGRAMMING ASSESSMENTS

2.1 Abstract

Providing feedback to students is one of the most effective ways to enhance their learning. With the advancement of technology, many tools have been developed to provide personalized feedback. However, these systems are only beneficial when interactions are done on digital platforms. As paper-based assessment is still a dominantly preferred evaluation method, particularly in large blended-instruction classes, the sole use of electronic educational systems presents a gap between how students *learn* the subject from the physical and digital world. This has motivated the design and the development of a new educational technology that facilitates the digitization, grading, and distribution of paper-based assessments to support blended-instruction classes. With the aid of this technology, different learning analytics can be readily captured. A retrospective analysis was conducted to understand the students' behaviors in an Object-Oriented Programming and Data Structures class from a public university. Their behavioral differences and the associated learning impacts were analyzed by leveraging their digital footprints. Results showed that students made significant efforts in reviewing their examinations. Notably, the high-achieving and the improving students spent more time reviewing their

mistakes and started doing so as soon as the assessment became available. Finally, when students were guided in the reviewing process, they were able to identify items where they had misconceptions.

This chapter was adapted from Paredes, Y. V., & Hsiao, I.-H. (2021). WebPGA: An educational technology that supports learning by reviewing paper-based programming assessments. *Information*, *12*(11), Article 450.

In today's blended learning environments, paper-based examination is still one of the most popular methods for assessing students' performance. Despite a wide range of computer-based approaches to conducting examinations, the traditional paper-based method still appeals to the teachers due to its flexibility and simplicity. It gives them a straightforward way to manage their class due to the required physical presence. For example, academic dishonesty could be deterred through in-person proctoring. However, the same conventional class management method also presents a challenge. Grading many papers can be time-consuming. It also requires significant effort to generate meaningful and personalized feedback to students while ensuring consistency within and between the graders. Most importantly, with the trend of gradually shifting towards technologically enhanced classrooms, such as smart classrooms or online streaming classes, the traditional blended classes necessitate an upgrade.

Several educational technologies that integrate physical and digital learning activities have started to proliferate. These systems, such as clickers (Trees & Jackson, 2007) and multi-touch tabletops (Martinez-Maldonado et al., 2013), paved the way for advanced learning analytics. However, support for personalized learning in these environments is still limited. Therefore, in this study, a web application called *Web-based Programming Grading Assistant* (WebPGA) was developed to capture and connect multimodal learning analytics from the physical and digital spaces in programming learning. It digitizes paper-based artifacts, such as quizzes and examinations, and provides interfaces for grading and feedback delivery at scale. The system enables students to manage their learning by consolidating assessment content, feedback, and learning outcome. WebPGA also allows for understanding better the students' behaviors in a blended class. Thus, the focus of this chapter is to explore and investigate the impacts

of the technology on students' learning. Specifically, it aims to answer the following research questions:

RQ A.1: In terms of monitoring and reviewing, are there any behavioral differences between high-achieving and low-achieving students?

RQ A.2: Are there any differences in the behavior of students when grouped according to performance trajectories (i.e., whether the student's score in a subsequent examination improved relative to a prior one)?

RQ A.3: What reviewing behaviors are associated with learning?

RQ A.4: How does personalized guidance affect the behavior of students when reviewing?

This chapter is organized as follows. Sections 2.2 to 2.6 discuss the role of assessments in higher education, the importance of feedback in programming learning, the emergence of behavioral analytics, and how personalization can be leveraged in educational systems. Section 2.7 describes in detail the design of the research platform. Section 2.8 provides an overview of the study design and the data collection process. Finally, Section 2.9 presents the findings and discussions.

2.2 Assessments in Higher Education

Assessments play an important role in learning in higher education. It is a process where data about students are collected to identify their strengths and uncover their weaknesses (Hanna & Dettmer, 2004). It also is a tool used to evaluate the effectiveness of the teacher's instructional strategies. Two of the commonly used types of assessments are formative and summative assessments. Formative assessments are low-stakes and typically not graded assessments that provide students feedback on

their current performance (e.g., practice quizzes). They enable students to diagnose and monitor their deficiencies, leading to improved learning. However, for it to be effective, students should be able to see the gap between their current ability and one that is expected of them and close it (Biggs, 1998). On the other hand, summative assessments are high-stakes assessments (i.e., graded) that aim to evaluate students' learning. These two types are viewed as *assessment for learning* and *assessment of learning*, respectively. A third view is *assessment as learning* which promotes students to reflect on their work and be metacognitively aware. Activities could be in the form of self or peer assessment which leads them to identify the next step in learning. Prior definitions, however, did not explain what happens to the assessment, per se. It was only recently given an updated definition to be "assessments that necessarily generate learning opportunities for students through their active engagement in seeking, interrelating, and using evidence" (Yan & Boud, 2021, p. 13). This highlights the importance of the active role of the student in the process.

2.3 Role of Feedback in Learning

A student's academic achievement is affected by several factors, such as learning experience, feedback, teaching style, and motivation. Some of these are more influential than others. Additionally, many of these are not easily quantifiable. Several papers have highlighted the importance of feedback and what constitutes an effective one. The timing of when it is delivered is also essential (Hattie & Timperley, 2007; Kulkarni et al., 2015). The sooner students receive their feedback, the more they can reflect on their learning. Moreover, the availability of immediate self-corrective feedback leads to an increase in the efficiency in reviewing examinations (Dihoff et al., 2004). Students

benefit more from feedback when assigned to individual components (e.g., rubrics), compared to just showing the overall score (Kulkarni et al., 2015). This would allow them to identify their misconceptions quickly. Furthermore, it was found that content feedback had significantly better learning effects than progress feedback (G. T. Jackson & Graesser, 2007). The mere provision of feedback, however, does not guarantee an improvement in students' learning. The student must take an active role in this process, essentially a shift from the *feedback as telling* mentality. Essentially, this conforms to the the proposed framework of Carless and Boud (2018) that underscores the importance of developing student feedback literacy.

2.4 Technology Support in Feedback Generation

Automated grading of assessment is one of the most popular methods employed to generate and deliver feedback at scale. It guarantees the timely release of feedback to students at a lower cost. Such a method has been widely used in several educational fields, such as programming, physics, and mathematics. Examples of these systems include WEB-CAT (Edwards & Perez-Quinones, 2008) and ASSYST (D. Jackson & Usher, 1997). Usually, pattern-matching techniques are used to assess the correctness of the student's work. This is done by performing unit tests and comparing the student's work to an ideal solution. This approach has some drawbacks. In programming learning, the logic and the reasoning of students are being overlooked by the system as it only focuses on the concrete aspects of the solution. As a result, teachers spend extra time reviewing the student's work after an auto-grader has evaluated it to provide personalized and better feedback. One proposed solution to address this is to crowd-source code solution, which will then be suggested to students (Hartmann et al., 2010).

Another approach suggests using student cohorts to provide peer feedback (Denny et al., 2008; Gehringer, 2001). Lastly, parameterized exercises can be used to create a sizable collection of questions to facilitate automatic programming evaluation (Hsiao et al., 2010).

The various feedback generation techniques discussed previously are focused on evaluating digital artifacts. Less is discussed in the context of paper-based programming problems, which can be addressed by digitization. This approach provides several advantages. For example, some default feedback can be kept on the digital pages with the predefined rubrics. Also, submissions can be anonymized, effectively eliminating any grader's biases. It is worth noting that there have been some relevant innovations that attempt to address this problem, such as Gradescope (Singh et al., 2017).

2.5 Behavioral Analytics in Programming Learning

Several studies have explored student modeling. Most intelligent tutors and adaptive educational systems heavily rely on these student models. Student learning is typically estimated using behavior logs. In programming learning, several parameters have been used to estimate students' knowledge of coding. One approach uses the sequence of success when solving programming problems (Guerra et al., 2014). Another approach considers the progression of the student on programming assignments (Piech et al., 2012). Some other approaches include: how students seek programming information (Lu & Hsiao, 2016), compilation behavior when doing assignments (Altadmri & Brown, 2015), troubleshooting and testing behaviors (Buffardi & Edwards, 2013), dialogue structures (Boyer et al., 2011), using snapshot of a code while solving programming problems (Carter et al., 2015).

2.6 Personalized Guidance in Learning

Personalized guidance refers to a group of techniques that provide learners with a straightforward path for learning. This often requires modeling the learning content (domain) and the learning process (interactions with the system), particularly in intelligent educational systems. This allows for material to be presented to learners in a personalized sequence (Chen, 2008). Additionally, it enables the learning process to be adapted so it can scaffold the learning activity (Azevedo & Jacobson, 2008). Changing the link appearances on the learning resources to be able to guide students to the most appropriate and relevant ones (also known as Adaptive Hypermedia) is one of the common techniques in personalized guidance (Brusilovsky, 1996, 1998). This leads to better results and higher satisfaction from learners as it helps them reach the right question at the right time. In the context of self-assessment, this increases the likelihood of students to answer a question correctly (Brusilovsky & Sosnovsky, 2005; Hsiao et al., 2010). This heavily relies on the interaction between the artificial intelligence of the system and the intelligence of the student. The adaptive navigation support method has been used in the social learning context. For example, a system that has open social student model interfaces used greedy sequencing techniques to improve students' level of knowledge (Hosseini et al., 2015a). It led to an increase in the speed of learning of strong students. It also improved the performance of students. It should be noted that the mere presence of personalized guidance in a system may not be enough to provide a learning impact. It always depends on whether students choose to follow the guidance or not (Hosseini et al., 2015b).

2.7 Web-based Programming Grading Assistant (WebPGA)

WebPGA was developed to connect the physical and the digital learning spaces in programming learning. It is an improvement of PGA (Hsiao, 2016), a system that allows the grading of paper-based programming assessments using smartphones. The goal is to facilitate the digitization, grading, and distribution of paper-based assessments in a blended learning environment. Furthermore, it aims to capture the different actions performed by its users.

There are three types of users, namely: teachers, graders, and students. This section discusses in detail the pedagogical foundations and technical implementation of WebPGA. The system is divided into two components, namely the grading interface and the reviewing interface.

2.7.1 Grading Interface

Teachers and graders use the system to grade paper-based assessments and to provide their feedback. They upload the scanned images of the examination papers to the system. The features discussed in this section represent different forms of feedback that can be provided to students. Figure 2 illustrates the grading interface where teachers and graders mainly interact.

2.7.1.1 Image Annotation

The left panel in Figure 2 illustrates the scanned image of the student's paper. Using the provided markers (red or yellow), graders can write directly on top of

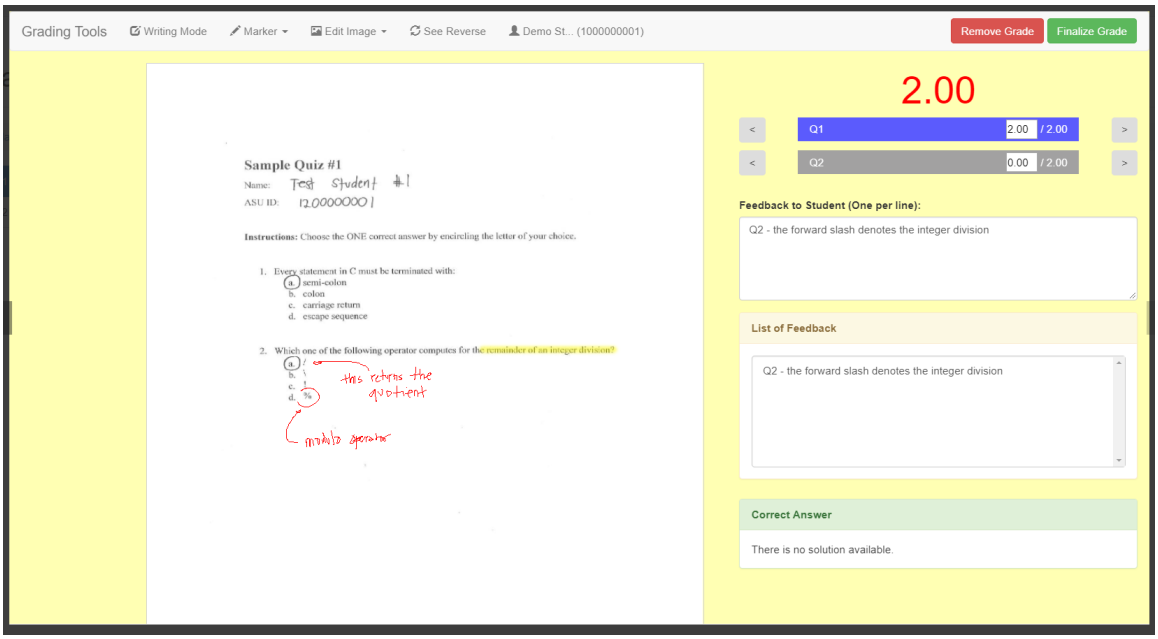


Figure 2. Grading Interface That Supports the Provisioning of Various Feedback to Students

Note: Grading interface provides the tools to assess students' answers. The left panel presents the scanned image while the right panel contains the rubrics and a textbox to provide free-form feedback.

the image. In Ball et al. (2009), such annotations are considered useful feedback to students. In certain instances, this approach is even more convenient than typing in free-form text boxes.

2.7.1.2 Grading Rubrics

The right panel in Figure 2 provides a detailed breakdown of the score obtained by the student. The top compartment displays the overall score. It is followed by a list of rubrics used to assess the work of the student as it is critical that students are informed how their work was evaluated and what was expected of them (Biggs & Tang, 2011). Ideally, these rubrics are associated with the knowledge components that

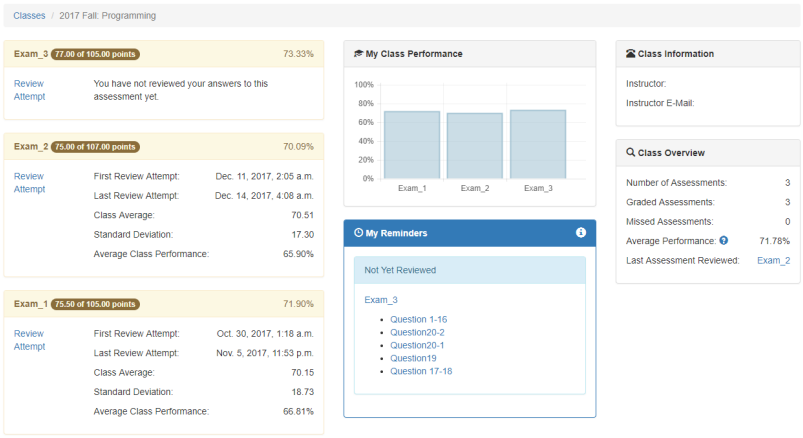
are being evaluated in a question. This makes it easier for students to identify their misconceptions (Kulkarni et al., 2015). A color scheme was employed to distinguish which concepts the students are struggling with easily. The color blue indicates a complete understanding, red indicates partial understanding, and gray represents a misconception.

2.7.1.3 Free-Form Feedback

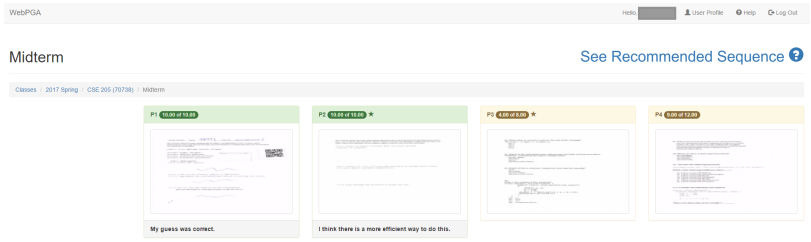
Rubrics alone are not enough forms of feedback. In fact, formative feedback is preferred as it contributes to learning than on correctness alone (Hattie & Timperley, 2007). Therefore, a text box was provided to allow graders to provide free-form feedback to students. This could be a justification of a deduction or a suggestion on how to improve the answer. In large classes, there is a tendency for graders to become inconsistent in the feedback they give. The system stores all the feedback given by all graders for a particular question to address this issue. These are then listed in the list box below. The feedback is arranged according to their frequency (i.e., most used to least used). Such approach was recommended by Biggs and Tang (2011).

2.7.2 Reviewing Interface

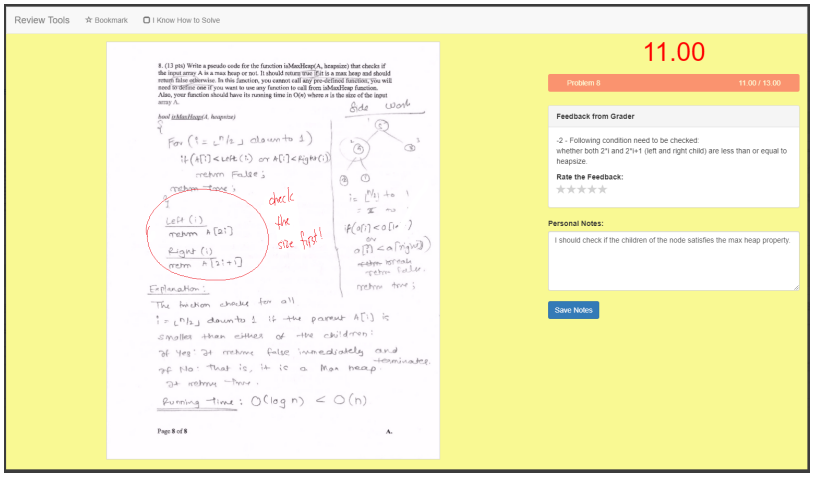
Students mainly benefit from using the system as the delivery of feedback (both summative and formative) becomes more efficient. This allows them to view their scores once they are made available conveniently. Figures 3a–c illustrate the different interfaces students interact with. These various levels uncover students' reviewing behaviors, particularly whether they simply looked at their scores or read the feedback–



(a) Student dashboard provides an overview of the student's performance



(b) Assessment overview lists all the questions of single assessment



(c) Question overview provides a detailed view of how a question was graded

Figure 3. Reviewing Interface Giving Students Feedback With Varying Granularity

a probable action in the system (Mensink & King, 2020). Such granularity allows for distinguishing how students appreciate the varying feedback provided to them by the system.

2.7.2.1 Dashboard

The dashboard (Figure 3a) provides students an overview of their class performance. The left panel lists all the assessments that can be reviewed. It includes information such as the scores, the first and the latest reviews, if applicable. The assessments are arranged in a reversed chronological order. A color scheme was used to highlight the importance of an assessment. The assessment panel is colored in green if the student had a perfect score. Otherwise, it is colored yellow. If the assessment is not for credit, it is colored in blue. When students click on a particular assessment to review, they are redirected to the assessment overview. The middle panel provides students with a bar chart that visualizes how they are performing in class. Below it is a personalized reminder panel which will be discussed in detail later. Lastly, the right panel provides administrative information about the class along with their performance.

2.7.2.2 Assessment Overview

In the assessment overview (Figure 3b), all the questions for a particular assessment are listed along with the scores obtained and personal notes made by the student. A color scheme was used to make the presentation meaningful. Green means the student obtained full credit, yellow means the student obtained partial credits, and red means the student did not obtain any credit. The questions are arranged according to how

they were ordered in their physical counterparts (i.e., when they were administered). However, students can follow the system's personalized recommended sequence (see Section 2.7.2.4) by clicking on the "See Recommended Sequence" link on the upper right portion. When students click on a particular question thumbnail to review, they are redirected to the question overview.

2.7.2.3 Question Overview

In the question overview (Figure 3c), more details about the question are provided to the students. The background color of this page follows the color used in the thumbnail in the assessment overview (green, yellow, or red). The left panel illustrates the image of the answer, including any annotations made by the graders. The right panel provides the overall score for the question, the rubrics and the different scores obtained. This follows the color scheme discussed in Section 2.7.1.2. It then shows the free-form feedback given by the grader and a 5-point Likert scale to rate their perceived quality of the feedback they received. The system records the amount of time spent by the student while in this view.

Three forms of reflection prompts were incorporated: (a) star bookmark to note the importance of or the need to reference a question in the future; (b) checkbox to express explicitly their ability to solve the problem; and (c) free-form text area where they can type elaborated notes. Such features can encourage students to do self-learning on their answers and self-reflect on their reasoning processes that could lead to a deep learning experience (Chi, 2000). Such collections of bookmarks, checkboxes, and notes enable students to be more metacognitively aware of their subject matter knowledge as this captures what they have learned (Roscoe & Chi, 2007).

2.7.2.4 Personalization

One of the system's design goals is to provide some interventions to help students who are falling behind in class. One issue in online learning systems (e.g., learning management systems) is the tendency of feedback to be spatially separated which hinders students from synthesizing them (Winstone et al., 2021). This could be addressed by providing personalized prompts in the system, particularly in the student dashboard and the assessment overview. Students are given personalized, actionable reminders that list all assessments or questions that have not been reviewed (the lower component of the middle panel in Figure 3a). The order of the items in the list is determined using Algorithm 1 which was designed based on prior studies (Paredes, Azcona, et al., 2018; Paredes, Hsiao, & Lin, 2018). The system assigns a higher importance to questions where the student made more mistakes (i.e., the student must review it first). From the list, if the student clicks on the name of an assessment, they are redirected to the assessment overview (Figure 3b) but with the questions arranged using Algorithm 2. On the other hand, if the student clicks on a specific question, they are redirected to the question overview (Figure 3c).

2.8 Methods

To investigate the effectiveness of the educational platform, a retrospective analysis was done on the reviewing behaviors of students by looking at their *review actions*. These are the instances where students interacted with the different views in the reviewing interface, as discussed earlier.

Algorithm 1 Assessment and Question Listing in the Reminders Panel

```
1: procedure GetAssessmentsAndQuestions(S)
2:     for each A ∈ assessments of student S do
3:         Q := list of questions from A which are not yet reviewed
4:
5:         for each q ∈ Q do
6:             q.normalized := q.raw_score_of_student / q.points_worth
7:
8:             if q.normalized = 1 then
9:                 Remove q from Q
10:
11:        if Q.isEmpty() then continue
12:        // Just in case there is a tie, use the next criteria
13:        Sort Q by q.normalized in ascending, q.points_worth in descending
14:
15:    // Display the latest assessment on top
16:    Sort all assessments according to assessment_date in descending
```

Algorithm 2 Recommended Sequence

```
1: procedure GetRecommendedSequence(A)
2:     Q := questions from assessment A
3:
4:     for each q ∈ Q do
5:         q.normalized := q.raw_score_of_student / q.points_worth
6:
7:     // Just in case there is a tie, use the next criteria
8:     Sort Q by q.normalized in ascending, q.points_worth in descending
```

2.8.1 Data Collection

The system data from an Object-Oriented Programming and Data Structures class offered during the Spring 2018 semester in a public university were collected. This 200-level course is the second programming class taken by Computer Science major students. The class was chosen since its instructor signified interest and volunteered to use the system, mainly to facilitate the grading process. This class had a total of 3 examinations and 14 quizzes (five are for credit, and nine are non-credit). There

Table 1. Distribution of Students When Grouped Based on Three Categories

Category	Group	No. of Students
Academic Performance	High-achieving	86
	Low-achieving	71
Performance Trajectory	Exam1-Exam2 Period	
	Improving	53
	Dropping	102
	Retaining	2
	Exam2-Exam3 Period	
	Improving	79
	Dropping	77
	Retaining	1
Personalized Guidance	Guided	46
	Not Guided	111

were 187 students enrolled, but only 157 (83.96%) were included in the analysis as those who dropped the course in the middle of the semester, did not take the three examinations, or did not use the system had to be removed.

2.8.2 Data Processing

Students were labeled and grouped in three different ways to understand how their monitoring and reviewing behaviors affect their learning. The breakdown is summarized in Table 1. First, they were grouped according to their overall academic performance. Then, they were grouped according to their performance trajectory in a given period. Finally, they were grouped according to whether they were guided by the system or not.

2.8.2.1 Overall Academic Performance

The final grades of the students were not included in the data collection. However, the examinations have the highest contribution to the final grade. Therefore, the student's average score for the three examinations was used to determine his or her overall academic performance in place of the final grade. Using the class average ($M = 82.28$, $SD = 10.83$) as the cut-off point, students were classified either as *high-achieving* or *low-achieving*. This cut-off value closely resembles the boundary between the A and B students and the C, D, and E students as set by the instructor.

2.8.2.2 Performance Trajectory

The overall performance only provides a single snapshot of the student. It is interesting to look at the different changes in how the student performed throughout the semester. Therefore, the examinations were used to divide the semester into two equal periods, namely: *Exam1-Exam2* and *Exam2-Exam3*. In a given period, the difference between the scores in the two examinations was computed. This value was referred to as *delta*, which represented the magnitude of improvement or dropping of the student. For that period, a student was labeled *improving* if the delta was positive; *dropping* if negative; and *retaining* if zero. It should be noted that a student may belong to different groups in the two periods.

2.8.2.3 Reviewing Behavior

Among the 21,747 student actions captured by the system, 9,851 (45.30%) were review actions. These actions have their corresponding *duration*, which represents the amount of time a student spent reviewing. Each review action was labeled according to the score obtained by the student in the question that was reviewed. It was labeled *r_ correct* if the student answered the question right. Otherwise, it was labeled *r_ mistake*.

2.8.2.4 Personalized Guidance

The system provides a personalized suggestion to each student, particularly on how and what to review. If a student clicked an assessment or a question from the list on the reminders panel (bottom component of the middle panel in Figure 3a); or clicked on the "See Recommended Sequence" link on the assessment overview (Figure 3b), the student was labeled *Guided*. Otherwise, the student was labeled *Not Guided*.

2.8.3 Data Analysis

In this study, after an assessment was graded, the teacher made an announcement to inform students that the assessment was available for review. This announcement was made using a learning management system.

An assessment was considered reviewed if at least one of its questions was reviewed. Table 2 gives an overview of how students reviewed their examinations. This includes the average class performance, the number of students who reviewed them, and the

Table 2. Overview of Students' Reviewing Behavior

Examination	Avg. Score (%)	Students who Reviewed		Reviewing Delay (days)	
		N	% of Class	M	SD
1	83.30	142	89.87	4.7	14.4
2	78.60	131	82.91	2.4	6.8
3	79.60	100	63.29	0.9	2.2

average time it took students before they reviewed it for the first time (hereinafter referred to as *reviewing delay*). A downward trend can be seen for both the number of students reviewing and their reviewing delay.

2.9 Results and Discussion

2.9.1 The Learning Effects of Reviewing Behaviors

To examine the impacts of reviewing assessments on students' learning, the efforts exerted by the high- and low-achieving students were compared and summarized in Table 3. The reviewing behaviors of the two groups were measured by (1) total number of review actions performed (*review count*) and (2) total time spent reviewing.

2.9.1.1 Impact of Assessment Types: Quizzes and Examinations

The system supports formative and summative assessments. In this class, the instructor administered three types of assessments: non-credit quizzes (used for attendance and the answers of the students are not checked), quizzes for credit (answers

Table 3. Comparison of System Usage and Reviewing Behaviors of High-achieving and Low-achieving Students

Reviewing Behavior	High	Low
Review Count	48.10	47.73
Time Spent Reviewing Assessments (mins)	23.38	25.42
Examination Review Count	24.95	29.39
Time Spent Reviewing Examinations (mins)**	8.42	12.64
Correct (mins)	6.51	7.51
Mistakes (mins)**	1.62	4.73
Review Coverage*	0.73	0.65

Note: $^{**}p < 0.05$ $^{*}p = 0.05$

of the students are checked), and examinations (midterm and final). The non-credit quizzes served as a formative assessment, while the quizzes for credit and the examinations were considered summative assessments. All the review actions performed by the students were logged, regardless of the type of assessment. It is hypothesized that students would pay more attention to assessments that directly contribute to their final grades (quiz for credit and examinations). It is also hypothesized that the non-credit quizzes may affect students' reviewing behavior since they may not have given importance to items that do not count towards their final grades. However, when the overall number of reviewing actions and the time spent of the two groups were compared, no significant difference was found. The results suggested that all students paid the same amount of attention to the graded assessments, regardless of the assessment type. It is important to note that high-achieving students have fewer mistakes to review while low-achieving students have relatively more mistakes to review. Do these students put in the same amount of effort in reviewing the *appropriate* item?

2.9.1.2 High Achievers Focused on Reviewing Their Mistakes

To investigate further the difference of the reviewing efforts of the two groups as well as to answer **RQ A.1**, how these groups reviewed their graded examinations were looked into. Both groups still had a similar number of review actions performed. However, high-achieving students ($M = 8.42$ minutes) spent significantly ($p < 0.05$) lesser time reviewing all their examinations compared to low-achieving students ($M = 12.64$ minutes). There are several possible explanations for this. High-achieving students would have fewer mistakes and may not have reviewed their correct answers, leading to less time on the system. It is also possible that high-achieving students already knew which items to focus on. Lastly, it is also possible that low-achieving students may have struggled to identify which questions to review and therefore spent more time. Spending more time reviewing may not necessarily be an effective strategy. A student may review several times but may not be on items that require their focus—their mistakes. To investigate this, the time spent was subdivided into two categories: on *correct answers* and on *mistakes*. Interestingly, the two groups spent a similar amount of time reviewing their correct answers. However, when reviewing mistakes, low-achieving students spent significantly more time compared to high-achieving students. This was not surprising since low achievers had more mistakes. Therefore, the *review coverage* for mistakes of the two groups were compared. This refers to the proportion of questions that the students actually reviewed. In this case, the percentage of their mistakes that they reviewed. Although just marginally significant ($p = 0.05$), high-achieving students were able to review most of their mistakes compared to the low-achieving students. This would translate into an ineffective reviewing strategy for low-achieving students. They had more mistakes and did not exert enough effort to

review them. This clearly exhibits a bad habit of students since they are unable to take advantage of learning from the feedback they were provided, which could help them correct any of their misconceptions. It is worth investigating in the future if such a trend becomes more pronounced with more examinations. Succeeding analyses will focus mainly on review actions on examinations.

2.9.1.3 Improving Students Reviewed Most of their Mistakes

The previous section looked into the main effects of the aggregated performance of the students throughout the semester. In this section, students were analyzed at a finer granularity—across examination periods. This deeper analysis allowed the dissection of the changes in students' behavior over time and the exploration of the potential various strategies students' employed across different examinations.

Improving students were not necessarily high achievers. The goal is to determine how students differed and what led to the improvement of their grades, essentially answering **RQ A.2**. For each group, the review coverage for both their correct answers ($r\_correct$) and mistakes ($r\_mistake$) were computed. This is summarized in Table 4. The retaining group was omitted because of the negligible number of students. It can be observed that on average, both groups did not review all their answers. For example, the improving students during Exam1-Exam2 period reviewed only 39% of their correct answers and 63% of their mistakes. For both periods, improving students consistently focused on reviewing most of their mistakes, demonstrated by the higher review coverage for mistakes (63% and 32%) compared to correct answers (39% and 15%). This suggests that focusing on your mistake to answer them right the next time may help in improving your grade. In the case of the dropping students,

Table 4. Comparison of Review Coverage Across the Two Periods

Period	Improving		Dropping	
	r_correct	r_mistake	r_correct	r_mistake
Exam1-Exam2	0.39	0.63	0.42	0.64
Exam1-Exam3	0.15	0.32	0.26	0.31

Note: Both groups had $p < 0.01$ for each period except for Dropping during Exam1-Exam2 period.

during the Exam1-Exam2 period, they also focused on reviewing their mistakes as they reviewed 64% of them. However, during the Exam2-Exam3 period, no significant difference was found in their effort in reviewing their correct answers and mistakes. It should be noted that during this period, more assessments were available for review. Interestingly, during this period, no significant difference can be seen between the strategies of the improving and the dropping students (32% and 31%, respectively). This strategy may have worked on the former group but not on the latter group. Possibly, dropping students may have overlooked their mistakes, thus were unable to take full advantage of the feedback they were given. This is an ineffective strategy and intervention strategies should be developed and applied.

2.9.1.4 Spending More Time Reviewing Mistakes is Associated with Improved Performance

A drop of a single point may not have a significant impact on a student's behavior compared to a drop of 10 points. With the current grouping, there would not be any distinction between the two. Therefore, the actual values of the *deltas* were used instead of only the sign. These represent the magnitude of change in the performance of students in a period (*magnitude*). The amount of time spent by students on

reviewing their mistakes was obtained (*effort*). For both periods, a Pearson correlation coefficient was computed to assess the relationship between the two variables. There was a significant positive correlation between the magnitude and the effort for both period ($r = 0.19$, $p < 0.05$ for Exam1-Exam2 and $r = 0.23$, $p < 0.05$ for Exam2-Exam3). This means that students who improved focused on their previous mistakes.

2.9.1.5 Reviewing Promptly is Associated with Academic Performance

Some students attended to their graded assessments as soon as they were made available, while some waited until the last minute before the next examination. To determine the effectiveness of the reviewing strategy, students' reviewing efficiency was examined. This was obtained by getting the average *reviewing delay* for all examinations reviewed by the student. A negative relationship was found ($r = -0.16$, $p < 0.05$) between their academic performance. This means that better-performing students attended and reviewed their graded examinations sooner. This initiative and motivation are among the characteristics of a self-regulated learner that lead to improved academic outcome (Zimmerman, 1990). Being more vigilant in reviewing could potentially be associated with better grades. Another interpretation is that students who obtained better grades started to prepare for an examination early seriously.

The trend on how students attended to their graded assessments is visualized in Figure 4. From this, it can be observed that students reviewed examinations sooner than quizzes (shown by the dips). However, this was not unexpected. This suggests that students were more attentive when the credit at stake was high. The steep downward trend right before examinations (particularly for Exams 2 and 3) could

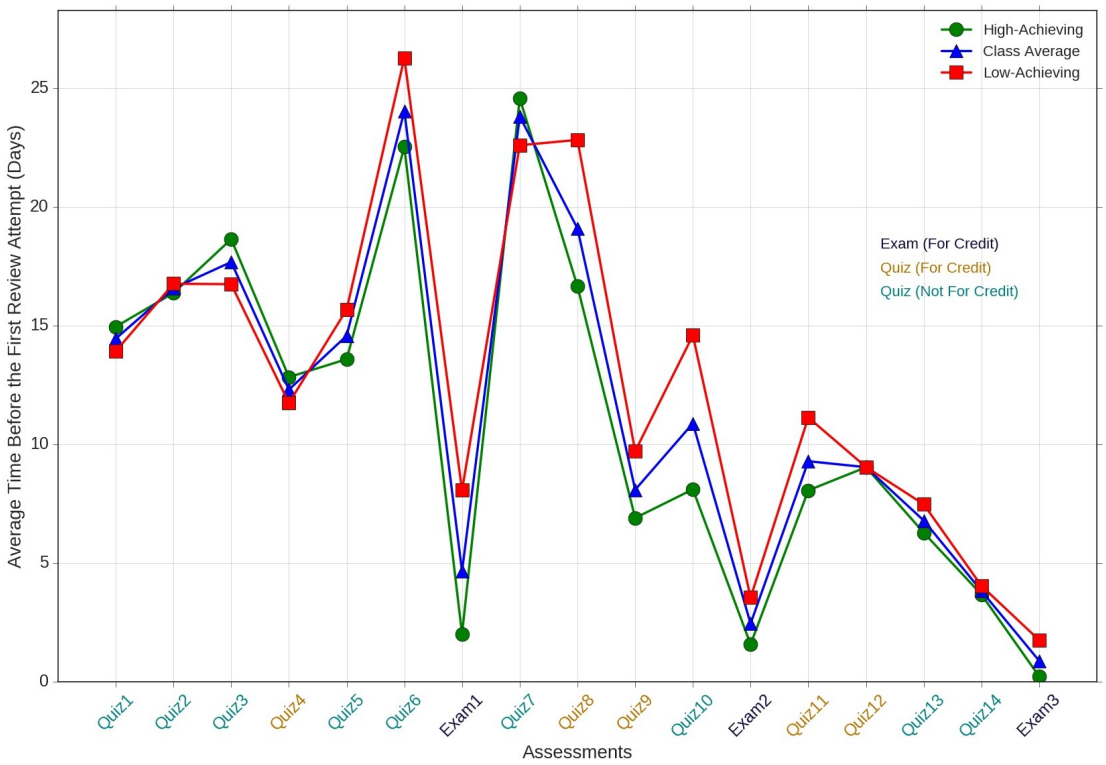


Figure 4. Comparison of Reviewing Delay of High-achieving and Low-achieving Groups Throughout the Semester

be due to students reviewing multiple quizzes before an examination. Eventually, students learned how to use the system, as demonstrated by the overall downward trend. They even started reviewing quizzes sooner, even if the quizzes were not for credit. This is an encouraging note and evidence of how students self-regulate their learning in reviewing assessments. Finally, when the trend lines of the two groups are compared, it can be seen that high-achieving students generally reviewed their assessments sooner (notice that the green line is generally the lowest line throughout the semester).

RQ A.3 can be answered by looking back at the findings in the prior sections to gain insight. Attending to their mistakes promptly was a key characteristic of high achievers and improving students. Such behavior could indicate a willingness

to fix any inconsistencies or misconceptions. Items where they made mistakes are likely to have more feedback provided by the grader. Therefore, spending more time resulted in an improvement in their performance which is consistent with the findings of Zimbardi et al. (2017).

2.9.2 Personalized Guidance Effects: Students Reviewed More Mistakes

The personalized guidance component was introduced to highlight the items that need to be prioritized when reviewing. **RQ A.4** can be answered by looking into whether the students used such feature. Although no significant differences were found in the academic performance of those who were guided and not, a difference in their reviewing behavior was found. It is hypothesized that when students are guided, their learning will improve. However, when the overall academic performance of those who were guided and those who were not was compared, no significant difference was found. It should be noted that the degree of guidance the students received from the system was not measured. Furthermore, students may have been guided at various times throughout the semester. Since the guidance had no impact on their learning, their reviewing behavior was compared, particularly the review coverage for mistakes. Students who were guided ($M = 0.76$, $SD = 0.28$) were able to significantly ($p < 0.05$) review more of their mistakes than those who were not ($M = 0.67$, $SD = 0.32$). The results showed that the personalized reviewing sequences successfully led students to focus on reviewing their misconceptions.

Feedback is indeed essential. For students to realize this, they need to be guided. Despite the potential of personalized guidance, only a few students used it (see Table 1). This raises the question of whether the guidance provided by the system

is enough or visible to them. Each student is different and needs a different form of guidance. Some do not even know that they need help (Aleven & Koedinger, 2000). As discussed by Carless and Boud (2018), there is a need for students to take an active role in the process and to come up with an effective strategy that works for them. Furthermore, the lack of difference in the overall performance between the two groups suggests that both high- and low-achieving students benefitted from this guidance. Additionally, some students who already have an effective reviewing strategy (i.e., highly self-regulated) may not need explicit guidance anymore and therefore did not use the feature.

2.9.3 Subjective Evaluation

At the end of the semester, students were instructed to anonymously answer an online survey to rate their experience using the system. They were also asked to provide some ideas on how the system could be further improved. Only 35 students (22.29%) responded to the survey. Figure 5 shows some of the questions and the students' responses.

2.9.3.1 Usefulness of Features

Respondents indicated that they understood the color scheme used and were aware of most of the system's features. However, some features had low usages, particularly the bookmark and the personal notes. Generally, the respondents were neutral about the usefulness of such features. This could be attributed to the fact

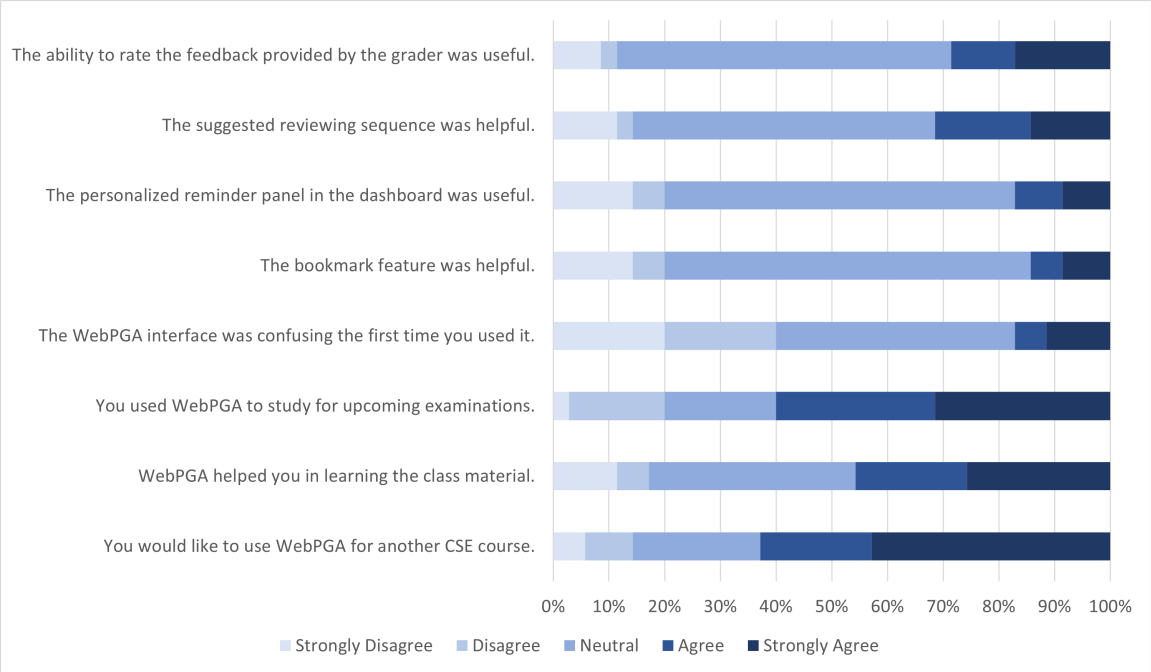


Figure 5. Survey Response of Students to an Anonymous Subjective Evaluation

that some other functionalities that would motivate them to use those features were not yet implemented.

2.9.3.2 Ease of Use

Most of the respondents found it easy to use the system. They became acquainted with it right after the first two quizzes. They indicated that they used the system to prepare for examinations. In fact, most of them wanted the system to be used in their other classes.

2.9.3.3 Future Improvement

Finally, to help improve the system, respondents were asked to provide their suggestions. One of the common responses was to include a way for them to rebut or challenge their grades. Another suggestion was to include a feature that would help them understand a specific question. With these suggestions, more interactions can be captured and could help further understand how students behave.

2.10 Conclusion

This chapter discussed the design of an educational technology that facilitates the digitization, grading, and distribution of paper-based assessments in blended-instruction classes. This system allows for the efficient delivery of feedback to students. It can capture the various interactions of students, providing empirical data on how they review their graded paper-based assessments. Such data can be leveraged to improve the design of existing educational tools. Additionally, it can provide personalized guidance to students on how to review.

A retrospective analysis was conducted to understand the behavioral differences among the different types of students. The reviewing strategies which were associated with improvement and learning were investigated. Results showed that high achievers exerted effort to review most of their mistakes. When analyzed further in finer granularity, students who improved exhibited the same behavior. They reviewed most of their mistakes and spent more time doing so. With the personalized guidance of the system, students were able to review most of their mistakes. Better students reviewed their graded assessments sooner.

This study is subject to several limitations. This investigation focused only on students' voluntarily reviewing behavior to signify one of the self-regulated learning processes: the abstract form of monitoring and reviewing one's learning. More comprehensive scenarios, such as planning, comprehension monitoring, and self-explaining should also be considered. The depth of the guidance the system provided the students was not measured. A better way to quantify this should be explored to determine how it affects students' performance. A more comprehensive algorithm should be considered for the personalized guidance to investigate whether such effect still exists. The sequence of questions that students reviewed could be studied in the future. Sequential pattern mining techniques along with clustering techniques could be used to determine whether different groups of students are exhibiting specific strategies. Students were not taught how to use the system. They had to familiarize themselves on their own. The usability of the system should be studied. Students who did not use the system were dropped from the analysis. However, the participation of these students could potentially provide new insights. This could be done through an interview or the use of self-reporting mechanisms.

The findings have implications for the future development of the system. For feedback to be effective, students must take an active role in the sense-making process to improve their performance (Boud & Molloy, 2013). New functionalities could be introduced to engage the students fully. For example, providing students an opportunity to discuss the feedback with their peers or their teacher. Peer evaluation could also be supported. Ultimately, the goal is to find new ways to make students feedback literate and guide them in the process.

2.11 References

Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? *International Conference on Intelligent Tutoring Systems*, 292–303.

Altadmri, A., & Brown, N. C. (2015). 37 million compilations: Investigating novice programming mistakes in large-scale student data. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 522–527.

Azevedo, R., & Jacobson, M. J. (2008). Advances in scaffolding learning with hypertext and hypermedia: A summary and critical analysis. *Educational Technology Research and Development*, *56*, 93–100.

Ball, E., Franks, H., Jenkins, J., Mcgrath, M., & Leigh, J. (2009). Annotation is a valuable tool to enhance learning and assessment in student essays. *Nurse Education Today*, *29*(3), 284–91.

Biggs, J. (1998). Assessment and classroom learning: A role for summative assessment? *Assessment in Education: Principles, Policy & Practice*, *5*(1), 103–110.

Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. McGraw-Hill Education.

Boud, D., & Molloy, E. (2013). Rethinking models of feedback for learning: The challenge of design. *Assessment & Evaluation in Higher Education*, *38*(6), 698–712.

Boyer, K. E., Phillips, R., Ingram, A., Ha, E. Y., Wallis, M., Vouk, M., & Lester, J. (2011). Investigating the relationship between dialogue structure and tutoring effectiveness: A hidden Markov modeling approach. *International Journal of Artificial Intelligence in Education*, *21*(1-2), 65–81.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-adapted Interaction*, *6*(2-3), 87–129.

Brusilovsky, P. (1998). Methods and techniques of adaptive hypermedia. *Adaptive hypertext and hypermedia* (pp. 1–43).

Brusilovsky, P., & Sosnovsky, S. (2005). Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK. *Journal on Educational Resources in Computing*, *5*(3), Article 6.

Buffardi, K., & Edwards, S. H. (2013). Effective and ineffective software testing behaviors by novice programmers. *Proceedings of the 9th Annual International ACM Conference on International Computing Education Research*, 83–90.

Carless, D., & Boud, D. (2018). The development of student feedback literacy: Enabling uptake of feedback. *Assessment & Evaluation in Higher Education*, *43*(8), 1315–1325.

Carter, A. S., Hundhausen, C. D., & Adesope, O. (2015). The normalized programming state model: Predicting student performance in computing courses based on programming behavior. *Proceedings of the 11th Annual International ACM Conference on International Computing Education Research*, 141–150.

Chen, C.-M. (2008). Intelligent web-based learning system with personalized learning path guidance. *Computers & Education*, *51*(2), 787–814.

Chi, M. T. H. (2000). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. *Advances in Instructional Psychology*, *5*, 161–238.

Denny, P., Luxton-Reilly, A., & Hamer, J. (2008). Student use of the PeerWise system. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, *40*, 73–77.

Dihoff, R. E., Brosvic, G. M., Epstein, M. L., & Cook, M. J. (2004). Provision of feedback during preparation for academic testing: Learning is enhanced by immediate but not delayed feedback. *The Psychological Record*, *54*(2), 207–234.

Edwards, S. H., & Perez-Quinones, M. A. (2008). Web-CAT: Automatically grading programming assignments. *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, 328.

Gehringer, E. F. (2001). Electronic peer review and peer grading in computer-science courses. *Proceedings of the 32nd SIGCSE Technical Symposium on Computer Science Education*, 139–143.

Guerra, J., Sahebi, S., Lin, Y.-R., & Brusilovsky, P. (2014). The problem solving genome: Analyzing sequential patterns of student work with parameterized

exercises. *Proceedings of the 7th International Conference on Educational Data Mining*, 153–160.

Hanna, G. S., & Dettmer, P. (2004). *Assessment for effective teaching: Using context-adaptive planning*. Allyn & Bacon.

Hartmann, B., MacDougall, D., Brandt, J., & Klemmer, S. R. (2010). What would other programmers do: Suggesting solutions to error messages. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1019–1028.

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81–112.

Hosseini, R., Hsiao, I.-H., Guerra, J., & Brusilovsky, P. (2015a). Off the beaten path: The impact of adaptive content sequencing on student navigation in an open social student modeling interface. *Proceedings of the 17th International Conference on Artificial Intelligence in Education*, 624–628.

Hosseini, R., Hsiao, I.-H., Guerra, J., & Brusilovsky, P. (2015b). What should I do next? adaptive sequencing in the context of open social student modeling. *Proceedings of the 10th European Conference on Technology Enhanced Learning*, 155–168.

Hsiao, I.-H. (2016). Mobile grading paper-based programming exams: Automatic semantic partial credit assignment approach. *Proceedings of the 11th European Conference on Technology Enhanced Learning*, 110–123.

Hsiao, I.-H., Sosnovsky, S., & Brusilovsky, P. (2010). Guiding students to the right questions: Adaptive navigation support in an e-learning system for Java programming. *Journal of Computer Assisted Learning*, *26*(4), 270–283.

Jackson, D., & Usher, M. (1997). Grading student programs using ASSYST. *Proceedings of the 28th SIGCSE Technical Symposium on Coomputer Science Education*, 335–339.

Jackson, G. T., & Graesser, A. C. (2007). Content matters: An investigation of feedback categories within an ITS. *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, 127–134.

Kulkarni, C. E., Bernstein, M. S., & Klemmer, S. R. (2015). PeerStudio: Rapid peer feedback emphasizes revision and improves performance. *Proceedings of the 2nd ACM Conference on Learning @ Scale*, 75–84.

Lu, Y., & Hsiao, I.-H. (2016). Seeking programming-related information from large scaled discussion forums, help or harm? *Proceedings of the 9th International Conference on Educational Data Mining*, 442–447.

Martinez-Maldonado, R., Dimitriadis, Y., Martinez-Monés, A., Kay, J., & Yacef, K. (2013). Capturing and analyzing verbal and physical collaborative learning interactions at an enriched interactive tabletop. *International Journal of Computer-Supported Collaborative Learning*, *8*(4), 455–485.

Mensink, P. J., & King, K. (2020). Student access of online feedback is modified by the availability of assessment marks, gender and academic performance. *British Journal of Educational Technology*, *51*(1), 10–22.

Paredes, Y. V., Azcona, D., Hsiao, I.-H., & Smeaton, A. F. (2018). Learning by reviewing paper-based programming assessments. *Proceedings of the 13th European Conference on Technology Enhanced Learning*, 510–523.

Paredes, Y. V., & Hsiao, I.-H. (2021). WebPGA: An educational technology that supports learning by reviewing paper-based programming assessments. *Information*, *12*(11), Article 450.

Paredes, Y. V., Hsiao, I.-H., & Lin, Y.-L. (2018). Personalized guidance on how to review paper-based assessments. *Proceedings of the 26th International Conference on Computers in Education*, 26–30.

Piech, C., Sahami, M., Koller, D., Cooper, S., & Blikstein, P. (2012). Modeling how students learn to program. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 153–160.

Roscoe, R. D., & Chi, M. T. H. (2007). Understanding tutor learning: Knowledge-building and knowledge-telling in peer tutors' explanations and questions. *Review of Educational Research*, *77*(4), 534–574.

Singh, A., Karayev, S., Gutowski, K., & Abbeel, P. (2017). Gradescope: A fast, flexible, and fair system for scalable assessment of handwritten work. *Proceedings of the Fourth ACM Conference on Learning @ Scale*, 81–88.

Trees, A. R., & Jackson, M. H. (2007). The learning environment in clicker classrooms: Student processes of learning and involvement in large university-level courses using student response systems. *Learning, Media and Technology*, *32*(1), 21–40.

Winstone, N., Bourne, J., Medland, E., Niculescu, I., & Rees, R. (2021). "Check the grade, log out": Students' engagement with feedback in learning management systems. *Assessment & Evaluation in Higher Education*, *46*(4), 631–643.

Yan, Z., & Boud, D. (2021). Conceptualising assessment-as-learning. *Assessment as learning* (pp. 11–24). Routledge.

Zimbardi, K., Colthorpe, K., Dekker, A., Engstrom, C., Bugarcic, A., Worthy, P., Victor, R., Chunduri, P., Lluka, L., & Long, P. (2017). Are they using my feedback? the extent of students' feedback use has a large impact on subsequent academic performance. *Assessment & Evaluation in Higher Education*, *42*(4), 625–644.

Zimmerman, B. J. (1990). Self-regulated learning and academic achievement: An overview. *Educational Psychologist*, *25*(1), 3–17.

Chapter 3

MODELING STUDENTS' ABILITY TO RECOGNIZE AND REVIEW GRADED ANSWERS THAT REQUIRE IMMEDIATE ATTENTION

3.1 Abstract

Students utilize various resources to prepare for an examination, such as lecture materials, homework, or previous quizzes or tests. Reviewing graded tests allows students to develop their metacognitive skills. However, a lack of proper guidance, exacerbated by a lack of maturity, hinders fully realizing the benefits of learning from past mistakes. In this paper, we investigated students' reviewing strategies. We analyzed the clickstream data of students taking a Computer Science Education course. Using Hidden Markov models (HMMs), we modeled the reviewing behaviors of high-performing and low-performing students. Our preliminary findings suggest that the two groups share some similar strategies but also have some that are particular to the group.

Students prepare for examinations using various resources that are made available to them. In an earlier survey, students indicated that apart from lecture materials, they also reexamine previous quizzes or tests (Paredes et al., 2017). Students used it as a practice opportunity anticipating that a similar question would come out in the examination. Reviewing assessments enables students to demonstrate and enhance their metacognitive skills, such as monitoring mistakes or evaluating a learning strategy's success and adjusting if necessary. Knowing how they performed in a graded assessment allows them to formulate a plan to address their misconceptions. This paper aims to determine whether students can identify the questions that require their immediate attention. Specifically, it aims to answer the following research questions:

RQ B.1: Do students review questions based on their performance?

RQ B.2: What reviewing patterns can be uncovered?

These questions can be answered by looking at how students interact with an educational technology that captures their reviewing behaviors. These strategies are captured in the form of clickstream data. Many approaches can be employed to model and interpret such behaviors. In this paper, we modeled students' clickstream behaviors using Hidden Markov models (HMMs) and presented our preliminary findings.

Earlier works have examined the distribution of the students' review actions and how this affects their succeeding examination performance (Paredes, Azcona, et al., 2018; Paredes et al., 2019). Moreover, when students review their graded tests, they benefit from being guided in identifying which items to focus on (Paredes, Hsiao, & Lin, 2018). However, these earlier investigations did not consider the dataset's sequential and temporal dimensions. The analyses focused only on the frequency of user actions and did not account for the transitions between them. Therefore, this current work aims to address the said limitation.

HMM is among the popular approaches to analyzing and modeling clickstream data (Rabiner, 1989). Beyond the educational data mining domain, many works have leveraged this technique to understand behavioral patterns (e.g., common transitions as visitors navigate an e-commerce website; Liu et al., 2017). An advantage of this approach is that it incorporates the temporal information of the data as opposed to simple clustering (Perera et al., 2008).

3.2 Methods

A total of 88,111 actions from clickstream data of 317 students enrolled in an Object-Oriented Programming and Data Structures class were analyzed. These interactions were captured using the educational tool WebPGA (Paredes et al., 2019). The course had a total of 17 paper-based assessments. Three of them were examinations, while the other 14 were quizzes. Two of the quizzes were for credit, while the rest were not. Students had to answer these quizzes and were awarded full points regardless of the correctness of their answers, as these were used for attendance.

Although the system can capture multiple student interactions, this preliminary analysis was limited to three specific actions. These actions represent the three levels of how a student can review an assessment as illustrated in Figure 6. The first level is the dashboard or class overview (Figure 6a), where students are presented with a list of all the assessments administered in class and the scores they obtained. The second level is the assessment overview (Figure 6b), where students are shown all the questions from the selected assessment. Their scores for the individual questions are shown at this level. From here, students can choose a question to review, which leads them to the third level or the question overview (Figure 6c). Students can see

fine-grained information about the question in the third level, such as the rubrics used to assess their answer, detailed feedback from the grader on why such a score was given, and written annotations on the digital paper.

3.2.1 Data Pre-Processing

The students used the platform throughout the semester at their convenience. They were informed via announcements in the learning management system (i.e., Blackboard) immediately after an assessment was graded. Each student's overall performance was computed by averaging the student's scores in the three examinations. Lastly, students were classified as high-performing or low-performing using the class average ($M = 0.83$, $SD = 0.12$) as the cut-off point.

Each question had varying difficulty. To determine this, how the entire class performed was examined. The average score for all the answers to a question was computed. The higher the value, the easier the question is. This information was used to add context to the reviewing behavior of the students. The score obtained by the student was compared to the question's difficulty. If the student obtained a higher score, a review action on this graded answer was labeled a non-urgent question review; otherwise, it was labeled an urgent question review. The rationale behind this heuristic is that students should attend to the questions they did not satisfactorily do the soonest.

Each student is represented by a single sequence enumerating the various actions performed on the system. The system can identify a group of actions performed in a single session. Therefore, a symbol was introduced to indicate the beginning of a new

Table 5. Symbols Representing the Various Actions Performed by the Students

Symbol	Description
D	Viewing the class dashboard that shows an overview of the student's scores on all the assessments.
A	List all the questions of an assessment and the scores obtained by the student in each question. Allows them to choose a question to review in detail.
N	Reviewing a graded answer considered non-urgent. The student's score is above the threshold based on the question's difficulty.
U	Reviewing a graded answer considered urgent. The student's score is below the threshold based on the question's difficulty.
X	Reviewing an ungraded question. The question's difficulty is unknown. A student can still learn from this item.
S	Used as a marker to denote the beginning of another session in the student's sequence.

session. The average sequence length was 125 actions. Table 5 provides a summary of the various symbols used for the analysis.

Based on how the system was design, a typical workflow always begins with the class dashboard (Figure 6a), where students choose an assessment to review. Afterward, students choose a particular question to review (Figure 6b). Students can navigate to the next question using the next or previous buttons; or close the current question window, go back to the assessment overview, and choose another question from the list (Figure 6c). The system has a personalized notification panel on the top and is always visible regardless of where the student is. It enables students to navigate directly to specific questions not yet reviewed.

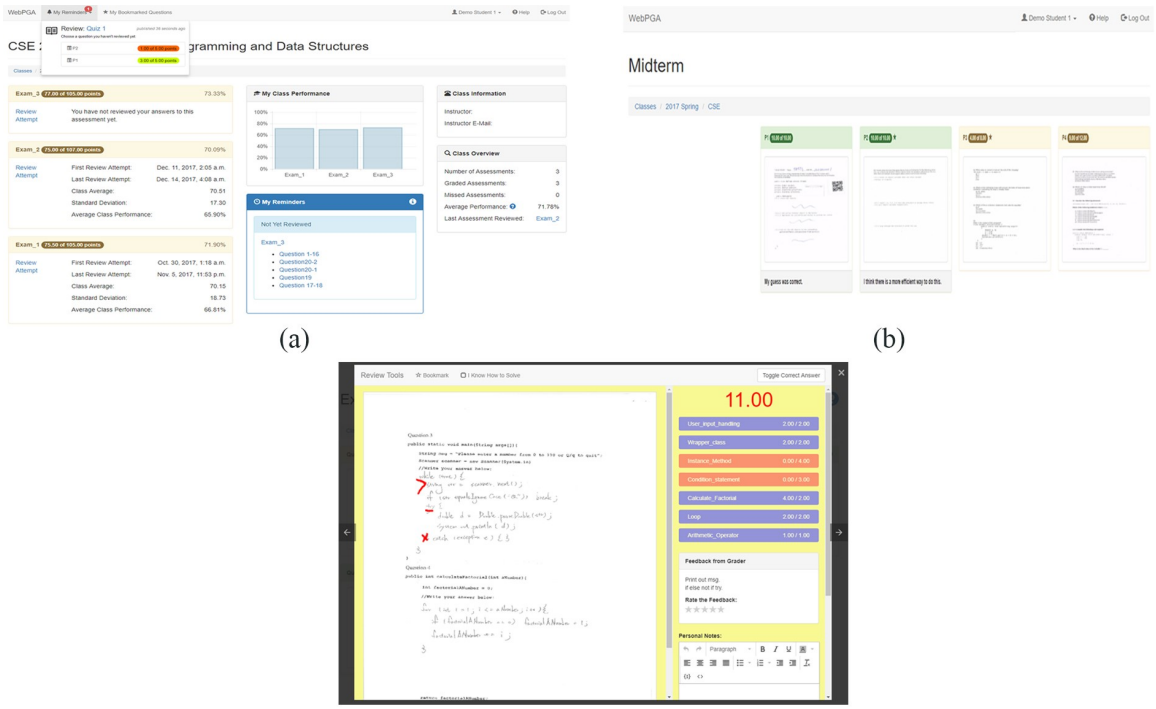


Figure 6. Three Levels of How Students Review Assessments

Note: (a) dashboard or class overview, (b) assessment overview, and (c) question overview

3.2.2 Hidden Markov Model

One common approach to modeling sequential data is through HMM (Rabiner, 1989). Two HMMs were developed to model the sequences of the two student groups, one for each group, and explore any similarities. The number of hidden states (HS) was a parameter that needed to be estimated. The parameter was set to four based on a similar early work where the Akaike Information Criterion (AIC) was used to determine the optimal number of hidden states (Hsiao et al., 2017). As shown in Table 2, each HS represents a strategy where the emission probabilities of each action are identified. The most probable action of a strategy is highlighted. Essentially, an

Table 6. Emission Probabilities of the Two HMMs

Group	Strategy	D	A	N	U	X	S
High	HS1	**0.70**	-	-	-	-	0.29
	HS2	-	**0.95**	-	-	0.03	0.03
	HS3	0.21	0.06	0.21	0.16	**0.36**	-
	HS4	-	0.15	**0.63**	0.22	-	-
Low	HS1	**0.59**	0.12	0.01	-	0.28	-
	HS2	-	**0.79**	0.09	0.12	-	-
	HS3	0.09	**0.60**	-	-	0.04	0.27
	HS4	0.03	-	0.31	**0.66**	-	-

Note: The most probable action for a strategy is highlighted in bold. Values less than 0.01 were omitted.

HS encapsulates the combination of actions that are likely to be done by the student. The transition probabilities between strategies (HS) of the two models are illustrated in Figure 7. Due to the system's design, the prior probability of the HS1 for both models is 1.00. It simply means that all sequences always begin with navigation from the dashboard.

3.3 Preliminary Results and Discussion

The hidden states reflect the students' reviewing strategies. As evident in the two groups' prior probabilities, students would always start their review from the dashboard. However, the high-performing group's emission probability for the HS1 strategy is limited only to actions D and S. This means they do not go into the details of any assessments at that moment. On the other hand, the HS1 strategy of the low-performing group shows more actions aside from the dashboard. They would go

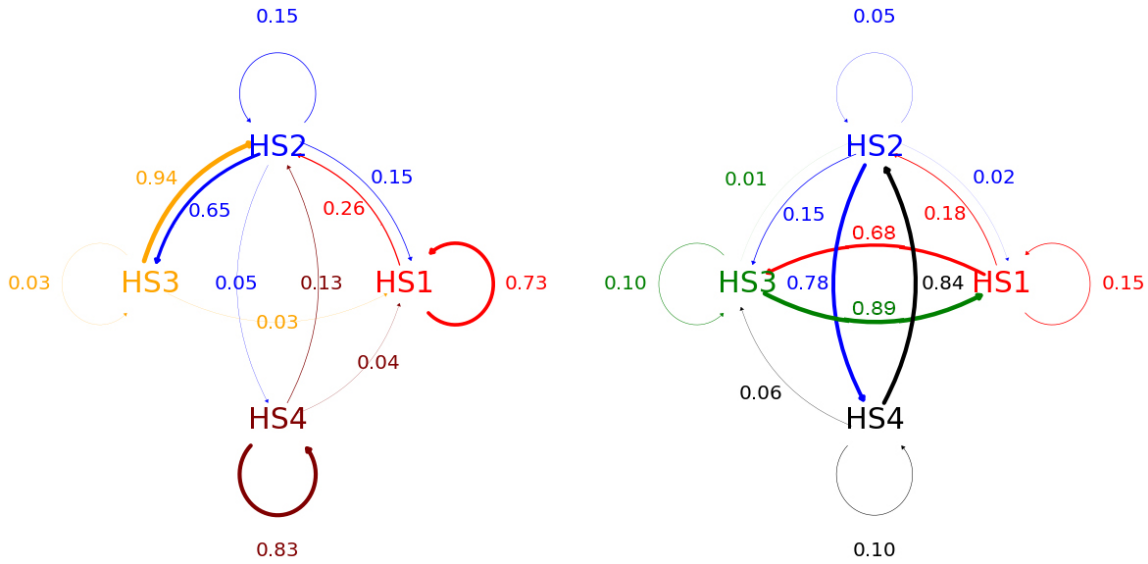


Figure 7. Visualiation of the Transition Probabilities of the Two HMMs

Note: The HMMs of the high-performing students (left) and low-performing students (right) both have four hidden states (HS), each representing a strategy.

ahead and review ungraded questions. This action by the low-performing group could indicate that they were using the notification panel.

3.3.1 High-Performing Students

High-performing students are likely to repeat their HS1 strategy of checking their overall performance from time to time. However, there are instances where they would change strategy and go into the details of an assessment, then later details of various questions as evidenced by their transition from HS1→HS2→HS3. Interestingly, in their HS3 strategy, the emission probability is high on ungraded questions, which suggests that they exert effort to review questions that were not graded to help them prepare for an exam. The loop in HS3→HS2→HS3 indicates that these students consciously determine which questions to review next instead of simply relying on the built-in

navigation buttons. It possibly suggests the ability of these students to recognize which of their graded answers to review next. This strategy could be a potential indicator of the student's awareness of planning on how to address their misconceptions. The HS4 strategy, which focused on reviewing the non-urgent questions, had a lower likelihood of happening since the only way to reach HS4 is through itself or from HS2.

3.3.2 Low-Performing Students

Low-performing students, like the high-performing students, had a high probability of transitioning to a strategy involving the assessment overview, HS2 or HS3 (more probable). A closer look into the more probable transition HS3 strategy's emission probability shows the presence of seeing the session marker. It suggests that these students often stopped reviewing at the assessment level and did not proceed further to the question level. It is even more pronounced in the following transition of HS3→HS1→HS3, meaning they would only log in to the system to check their scores without the intention of knowing where they made mistakes or learning from the feedback provided by the grader. The strategy for reviewing questions, particularly urgent ones, in detail HS2→HS4→HS2 involves a loop. These states can only be reached from the HS1 strategy.

3.4 Conclusion

This chapter examined the potential of modeling students' ability to recognize questions requiring immediate attention as they review and prepare for an upcoming examination. It also investigated whether the two student groups had different

strategies in this process. One of the limitations of this analysis involves estimating the parameters of the HMM. Although we followed the AIC method, several approaches can be explored that use information from the data. For example, Li and Biswas (2002) proposed a Bayesian approach to estimate the number of hidden layers based on the data. Two other approaches to analyzing sequential data include clustering students who had a similar distribution of actions they performed. Another is leveraging sequential pattern algorithms (e.g., Generalized Sequential Pattern; Srikant & Agrawal, 1996) to identify frequently performed actions. Differential pattern mining (Kinnebrew et al., 2013) which focuses on sequences specific only to certain student groups, is also a promising direction. Finally, instead of focusing on what actions are frequently performed on the system, another perspective is to examine each student group's distinct actions.

The clickstream data used in this study focused only on what was available on the system. This data can be used to complement other clickstream data from other systems, such as learning management systems. In effect, it would allow for a better understanding of the students, as shown in the work of Gitinabard et al. (2019). With the shift of most activities to online due to the Covid-19 pandemic, it is worth investigating if similar trends can be found in assessments administered electronically.

The current models can be incorporated into the system, allowing future studies to investigate how students would benefit from personalized interventions to improve reviewing behaviors. By analyzing the students' clickstream data in real-time, tailored suggestions in the form of notifications can be shown to students, making them aware of their current strategy. The same can be used to make them understand the strategies of successful students, hopefully enabling other students to emulate such behaviors.

Gitinabard, N., Barnes, T., Heckman, S., & Lynch, C. F. (2019). What will you do next? a sequence analysis on the student transitions between online platforms in blended courses. *Proceedings of the 12th International Conference on Educational Data Mining*, 59–68.

Hsiao, I.-H., Huang, P.-K., & Murphy, H. (2017). Uncovering reviewing and reflecting behaviors from paper-based formal assessment. *Proceedings of the 7th International Learning Analytics and Knowledge Conference*, 319–328.

Kinnebrew, J. S., Loretz, K. M., & Biswas, G. (2013). A contextualized, differential sequence mining method to derive students' learning behavior patterns. *Journal of Educational Data Mining*, *5*(1), 190–219.

Li, C., & Biswas, G. (2002). A bayesian approach for structural learning with hidden markov models. *Scientific Programming*, *10*(3), 201–219.

Liu, Z., Wang, Y., Dontcheva, M., Hoffman, M., Walker, S., & Wilson, A. (2017). Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 321–330.

Paredes, Y. V., Azcona, D., Hsiao, I.-H., & Smeaton, A. F. (2018). Learning by reviewing paper-based programming assessments. *Proceedings of the 13th European Conference on Technology Enhanced Learning*, 510–523.

Paredes, Y. V., & Hsiao, I.-H. (2022). Modeling students' ability to recognize and review graded answers that require immediate attention. *Proceedings of the 30th International Conference on Computers in Education Volume II*, 85–90.

Paredes, Y. V., Hsiao, I.-H., & Lin, Y.-L. (2018). Personalized guidance on how to review paper-based assessments. *Proceedings of the 26th International Conference on Computers in Education*, 26–30.

Paredes, Y. V., Huang, P.-K., & Hsiao, I.-H. (2019). Utilising behavioural analytics in a blended programming learning environment. *New Review of Hypermedia and Multimedia*, *25*(3), 89–111.

Paredes, Y. V., Huang, P.-K., Murphy, H., & Hsiao, I.-H. (2017). A subjective evaluation of web-based programming grading assistant: Harnessing digital footprints

from paper-based assessments. *Joint Proceedings of the 6th Multimodal Learning Analytics Workshop and the 2nd Cross-LAK Workshop*, 23–30.

Perera, D., Kay, J., Koprinska, I., Yacef, K., & Zaïane, O. R. (2008). Clustering and sequential pattern mining of online collaborative learning data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(6), 759–772.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements. *Proceedings of the 5th International Conference on Extending Database Technology*, 1–17.

Chapter 4

CAN STUDENTS LEARN FROM GRADING ERRONEOUS COMPUTER PROGRAMS?

4.1 Abstract

Learning from erroneous examples involves the intentional inclusion of errors as part of the learning process. Prior works, mostly from the field of mathematics, have investigated how this can be used in blended learning environments to help students. Due to the Covid-19 pandemic, most learning activities have shifted to online, motivating us to study and utilize students' use of an existing grading platform. Students were tasked to evaluate various degrees of erroneous answers as their learning opportunities, resembling program debugging. The grading process was engineered to supply feedback to students by revealing the actual marks and remarks to help them address their misconceptions and prepare them for an upcoming exam. This study presents our findings from clickstream data of students taking a synchronous online Computer Informatics class. How different students approached the activity was looked into: the amount of time spent and the difference of their assigned grade to that of a subject expert's. Although it is still inconclusive whether students learned from erroneous computer programs, we found that students who were proactive

in seeking feedback had better midterm scores than those who were not.

This underscores the importance of feedback in this learning process.

The Covid-19 pandemic has caused several disruptions, especially in learning in physical classrooms. The majority of the learning activities had to be shifted online or at home. With this impersonal environment, students and educators have more than ever desired innovative and engaging educational technologies. Fortunately, our research lab has designed and has been actively using and researching an online platform called WebPGA (see Chapter 2; Paredes & Hsiao, 2021b; Paredes et al., 2019). It bridges cyber and digital learning analytics for blended learning environments. The capability to support learning and teaching permitted the deployment and utilization of the platform to cope with the challenges of learning online during the pandemic. The system was leveraged by offering a new learning activity by involving students with new roles to participate in independent learning activity (Hsiao & Brusilovsky, 2011). Students were empowered to become graders assessing answers to past exams to mimic peer assessment, allowing them to exercise several metacognition skills, such as assessment and feedback-giving. The benefits to students are twofold. They learn from giving feedback and from seeing worked-out examples. This prepares students for an upcoming exam as they are provided actual questions from past exams. It enables them to gauge their knowledge, highlight some of their misconceptions about specific topics, and self-regulate their learning.

4.2 Learning from Peer Assessments

Peer assessment is a formative strategy where a learner gets to evaluate or comment on a peer's work (Black et al., 2003; Topping, 2017). Teachers in higher education mostly use this learning activity to ease their workload in grading. Research has shown that students benefit from this activity regardless of whether they are the

receiver or the giver of the feedback (Black et al., 2003; Hayes et al., 1987; Ion et al., 2019; Li et al., 2020). Students benefit from being exposed to other's work, allowing them to create a mental representation of successful or unsuccessful work. Students can interpret the assessment criteria of what a good performance is. This helps them adjust their actions to meet the expected results and make them more engaged in learning (Ion et al., 2019). In this learning activity, students adopt goals such as problem detection, diagnosis, and searching for strategies to fix the problems (Hayes et al., 1987). Students who consistently provided feedback outperformed those who simply rated the quality of the peer's work (Patchan & Schunn, 2015). Ion et al. (2019) found a high association between providing feedback and improvement in the current task and in transferring the knowledge to future tasks. Furthermore, students had better-perceived learning experiences and an increased sense of commitment to their own learning. Teachers play an essential role in this process, particularly by modeling ways to identify strengths and weaknesses from the work of others (Berg, 1999). Li et al. (2020) did a meta-analysis of peer assessment and found the importance of rater training. Research has been focused on training students to give task-related feedback as these motivate them to learn and improve in their work (Kamins & Dweck, 1999). Students can also be guided by using prompts or checklists of criteria or through regular practice.

4.3 Learning from Worked-Out Examples

In well-structured domains such as mathematics or computer programming, worked-out examples play an important role in the learner's initial acquisition of cognitive skills (VanLehn, 1996). It provides a description of the problem, a step-by-step

solution, and a final answer. Several studies have shown its effectiveness in improving student's learning (Renkl, 2002; Sweller & Cooper, 1985) and found it to be more effective compared to learning through problem-solving (Sweller & Cooper, 1985). However, its utilization does not guarantee that students would learn from it. It depends on the students' ability to explain the worked-out example to themselves as they make sense of the new information (i.e., self-explanation effect) (Chi, 2000; Renkl, 2002). Worked-out examples are not limited only to illustrating correct answers. In certain situations, errors are intentionally incorporated into the examples (i.e., erroneous examples). These mistakes can either be pointed out or left to be figured out. This presents learners an opportunity to detect inconsistencies and violations between their mental models and the normative model, which leads them to reflect, self-question, or self-explain (Rushton, 2018). Prior works, mostly in mathematics, have explored the impact of erroneous examples on student's learning (Isotani et al., 2011; Melis, 2005; Tsovaltzi et al., 2010). The literature is still inconclusive on its effectiveness as although it may improve student's performance, the improvement was not statistically different from other conditions that did not use erroneous examples (Griffin, 2019; Isotani et al., 2011). Despite this, in the domain of programming learning, this approach is popularly used as a learning exercise called *programming debugging*. Students are deliberately exposed to mistakes in a programming code and are expected to think critically and troubleshoot it to make it work. This learning activity presents as a learning opportunity for students (Brandt et al., 2010; Hsiao & Brusilovsky, 2011). This is a critical and indispensable skill one has to be equipped to succeed in programming. Therefore, in this case, it is worth looking into how erroneous examples play a role in programming learning. Thus, the focus of this chapter is to

explore and investigate on the feasibility of evaluating erroneous examples using an educational technology. Specifically, it aims to answer the following research questions:

RQ C.1: Do students learn from evaluating erroneous answers?

RQ C.2: Do students leverage feedback provided to them during the learning activity?

RQ C.3: What behaviors do students exhibit when evaluating erroneous answers?

RQ C.4: How do students benefit from receiving feedback during the learning activity?

In this chapter, erroneous example is explored in the form of answers (that contain errors) to programming problems. This analysis looked at how students evaluated such answers using WebPGA. Students not only get to see actual questions, but they also get to see actual answers and the actual grades or marks given to those answers. Section 4.4 provides an overview of the study design and the data collection process. Afterward, Section 4.5 presents the findings and discussions.

4.4 Methods

This section provides a discussion of the classroom study conducted. Particularly, it discusses the characteristics of the participants. Additionally, the erroneous examples used in the study are also discussed.

4.4.1 Participants

The behaviors of 63 non-Computer Science major students enrolled in a synchronous online Introductory Computer Informatics class offered during the fall semester of 2020 were looked into. Class instruction was delivered through Zoom, while resources and assessments were posted on Canvas. The topics covered were primarily focused on basic web programming using JavaScript. It covered database management concepts and SQL toward the end. The class had two midterm examinations. Students answered a pre-class survey to gauge their prior programming experience. 59 students responded. Most of them were familiar with Java (i.e., have heard of it) where the majority (58%) had less than a year of programming experience.

4.4.2 Research Platform and Study Design

In this study, students were assigned the role of being graders on WebPGA (see Chapter 2 for a detailed discussion of the original rationale and design of the research platform). Students were tasked to evaluate answers to past examination questions. This includes explaining why such an answer merits the grade they have assigned. They were given rubrics along with their corresponding weights to serve as their guide (Figure 8A). Students were made aware that they could grade an answer multiple times. The system can be accessed anytime and anywhere and can capture the amount of time spent by the students while performing the activity. It could also determine whether the students sought help by requesting feedback from the system for a particular answer. It can also identify how the answers were graded (i.e., which answer was graded first, next, and last) as each grading attempt is recorded

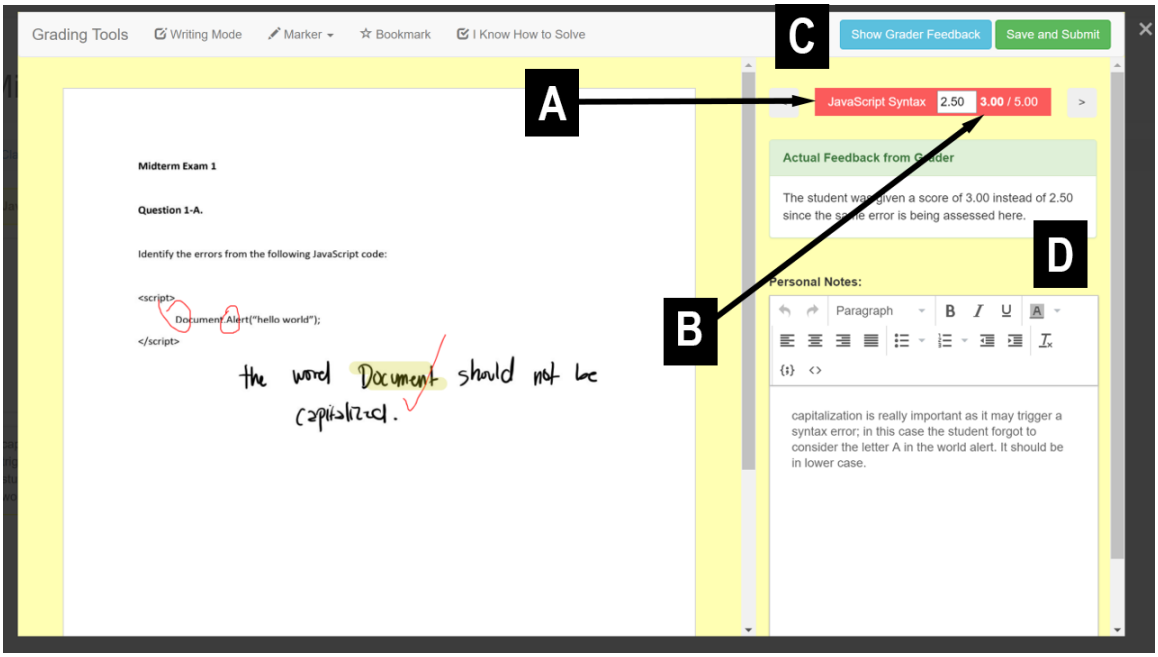


Figure 8. Grading Interface Used by Students Capable of Providing Two Levels of Feedback

Note: The two levels are *actual marks* (B) and *actual remarks* (D).

separately. During the class meeting prior to the first midterm, the system was introduced as a supplemental examination preparation tool. Students were told that usage or non-usage of it would not have any bearing on their final grades.

Students were given multiple answers to grade and were given the autonomy to determine the order on how to grade them. Figure 8 shows a screenshot of the grading interface. The system provides two levels of feedback. After grading an answer for the first time, a **Reveal Actual Marks** button on the upper right portion will appear. This will uncover the *actual marks* (Level 1) the answer received from a subject expert (Figure 8B, right next to the student's assigned grade) and will make the **Show Grader Feedback** button visible (Figure 8D). When the **Show Grader Feedback** button is clicked, the *actual remarks* (Level 2) the expert gave will be shown on the

Table 7. Overview of Actual Scores of Worked-Out Answers

Activity	Mistakes	QP1	QP2	QP3
A	M	4/8	5/8	2/7
	F	8/8	7/8	5/7
B	F	4/5	4/5	5/5
	M	0/5	1/5	1/5

Note: Activities A and B were administered prior to Midterms 1 and 2, respectively.

screen (Figure 8D). The student can either choose to seek feedback (either level) or continue grading the next answer.

Two activities (A and B) were given in class. For each activity, students were given three pairs of programming questions (QP). Each QP contains two versions of answers to the same question. One version has more mistakes (M) while the other has fewer (F). Both activities had six answers that needed to be graded. Activity A was given prior to Midterm 1. In this activity, the answers were arranged as follows: A-QP1-M, A-QP2-M, A-QP3-M, A-QP1-F, A-QP2-F, and A-QP3-F. Answers with more mistakes were shown first. Activity B was given prior to Midterm 2. In this activity, the order was swapped to counterbalance any order effects. Answers with fewer errors were shown first: B-QP1-F, B-QP2-F, B-QP3-F, B-QP1-M, B-QP2-M, and B-QP3-M. Table 7 summarizes the design along with the actual scores.

4.5 Results and Discussion

In this study, the impact of the learning activity was explored. A total of 11,029 clickstream data was analyzed to reveal students' strategies as they performed the tasks. In addition, students' responses to a survey to obtain their subjective evaluation of the system were examined.

Table 8. Student Performance Based on Activity Completion

	Midterm 1			Midterm 2		
	N	*M*	*SD*	*N*	*M*	*SD*
Complete	50	92.36	15.53	37	85.03	11.85
Incomplete	13	83.08	29.24	26	79.54	18.48

Note: Completion is defined based on whether all six questions from the activity were graded.

4.5.1 Did the Learning Activity Help the Students?

It is hypothesized that this learning activity would help students prepare and address any of their misconceptions as they are given the opportunity to see questions as well as answers to past exams. To answer RQ C.1, students were grouped based on whether they completed the activity prior to taking their midterm exam. The midterm scores of the two groups are summarized in Table 8. No significant differences in the students' scores for both Midterm 1 ($p = 0.14$) and Midterm 2 ($p = 0.09$) was found. This finding is similar to Griffin (2019) and Isotani et al. (2011). However, it can be observed that the scores of those who completed the activity were more coherent, as represented by having a standard deviation of almost half of those who did not. This could also be a good indicator that exposing students to erroneous examples may not be detrimental to the students' performance, at least when learning to program. It should be noted that this exploratory study focuses only on a single activity performed before an exam. More exploration should be done to know whether this trend persists when students are provided with multiple opportunities to see erroneous examples on the platform.

4.5.2 Exploring How Students Grade

The following sections provide a discussion on how to answer RQ C.2 and RQ C.3. Particularly, these look into whether leverage feedback during the grading process, how they improve their speed over time, whether they take personal notes, and how students' performance were in general.

4.5.2.1 Students Did Not Take Full Advantage of Feedback

For both activities, students had the opportunity to solicit feedback by either revealing the answer's *actual marks* (Level 1) or *actual remarks* (Level 2). Given that there were two versions of answers for every question, it was hypothesized that students would grade the first version, request feedback before grading the second version, and finally request feedback again. Such is referred to as the *immediately reviewed* behavior. It is possible to grade the first and the second versions separately then opt to seek feedback only after both have been graded. Such is referred to as the *delayed review* behavior. Any other behaviors exhibited were grouped as *neither*. This includes those who may have graded both versions but chose not to solicit feedback to either versions or both. Surprisingly, when students do solicit feedback, they would only do it to two out of the six questions in an activity. Students assumed that since the feedback for one version was already seen, the other version's feedback could be ignored. This could be an indicator that students are not taking full advantage of the feedback provided to them. This might affect how students reflect or self-explain this inconsistency between their mental model and the normative model. *What motivates students to seek help? Are they aware if they need help?* Among the six different

QPs, only 7.67% of the students *immediately reviewed*. 40.48% did a *delayed review*. Most of the students (51.85%) belonged to the *neither* group. Such counterintuitive findings is speculated to may have been driven by the design of the user interface. This merits further investigation.

4.5.2.2 Students Became Faster in Grading

How does the order of presentation of the answers affect the students' behaviors? In this section, how much time students spent grading was looked into as this could reflect how they strategize their problem detection and diagnosing approach to fix the problem. In Activity A, students took significantly more time grading the answers with more mistakes (Figure 9, left). This could be due to students trying to identify the errors and typing in their notes. Students became more critical as they figured out what to look for when assessing an answer. On the other hand, there was no significant difference in the time spent grading in Activity B (Figure 9, right). Seeing questions with fewer mistakes may be similar to seeing the answer key. Therefore, students are likely to be less critical as students are already aware of what to look for when grading. This could suggest that the order of presentation may have an impact on how students grade.

4.5.2.3 Students Were Taking Personal Notes

As students assess answers, they write in their own feedback or personal notes as a form of self-explanation (Chi, 2000) which hopefully would transfer knowledge to future tasks (Ion et al., 2019). The amount of time spent grading may have been

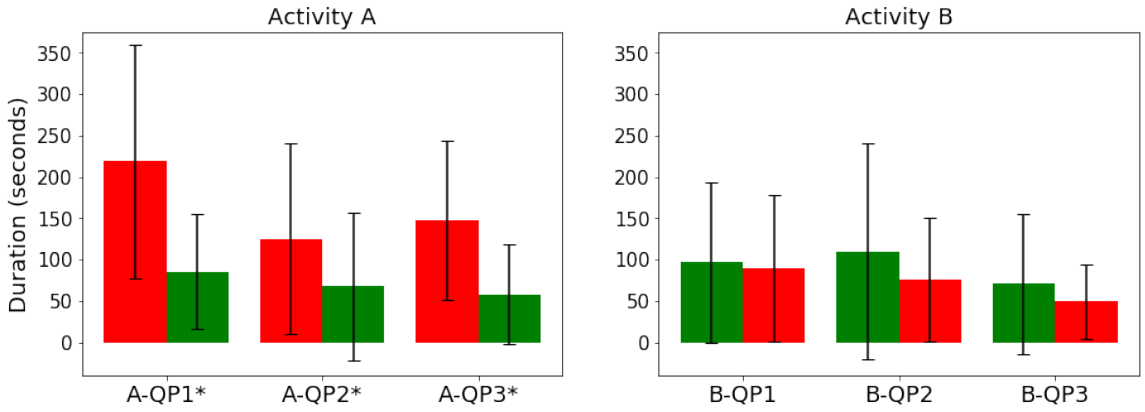


Figure 9. Comparison of Time Spent Grading Between Activities

Note: A difference in time spent grading was only seen when answers with more mistakes (red) were presented first instead of those with fewer mistakes (green); $^{*}p < 0.05$

influenced by these notes' length (in characters). Unsurprisingly, a significant positive correlation between the length of the notes and the amount of time they spent grading a question was found for both Activity A ($r = 0.54$) and Activity B ($r = 0.56$). The amount of time spent is composed of multiple factors: (1) analyzing the problem at hand, (2) figuring out the mistake, (3) identifying the solution, (4) providing a justification why such an answer deserves a particular grade. The average length of the notes between the two activities were compared to examine the impact of how the answers were arranged. Students had significantly longer notes in Activity A ($M = 88, SD = 103$) compared to Activity B ($M = 59, SD = 81$). Upon closer look at Activity A, students wrote longer notes on answers with more mistakes. This behavior was not seen in Activity B, where the arrangement of the questions was reversed. This suggests that the order may have mattered on how students took down notes. If exposed to more mistakes first, they will be more critical in note-taking. When exposed to fewer mistakes first, they will be more lenient and likely to write shorter notes.

4.5.2.4 Students Tend to Overgrade Answers with More Mistakes

In this study, the *actual marks* were considered as the grades assigned by a subject expert. The expert's marks were compared with the students' marks for every answer by computing their differences and rescaled them to $[-1, 1]$. Ideally, the value should be close to zero. A positive value indicates the student assigning a higher grade than the actual (*overestimated*). A negative value indicates the opposite (*underestimated*). Among the six QPs, we found that regardless of the arrangement of the answers (i.e., more mistakes showed first or the other way around), students were not good at grading answers with more mistakes as they tend to overestimate the grade (Figure 10). This is despite them being provided rubrics that could aid them in the process. Students need to be informed and guided on what constitutes a crucial part of a question to help them answer a similar question in the future. This highlights the importance of teachers' role in modeling what to look for when assessing answers (Berg, 1999). In their absence in this activity, the two levels of feedback that the system could provide was used as a proxy to help them do better the next time. This overestimation of grade assignment is a concerning behavior as this could be a projection of their own performance and a reflection of overconfidence in their skill (Kruger & Dunning, 1999; Magnus & Peresetsky, 2018; Serra & DeMarree, 2016) or may indicate a knowledge gap. Therefore, how this behavior correlates to their exam performance was looked into. For each student, the average grade differences for the six questions in the activity was computed and associated it with the student's score on the midterm exam corresponding to the activity. A significant negative correlation between the average grade difference and the midterm exam scores was found for both

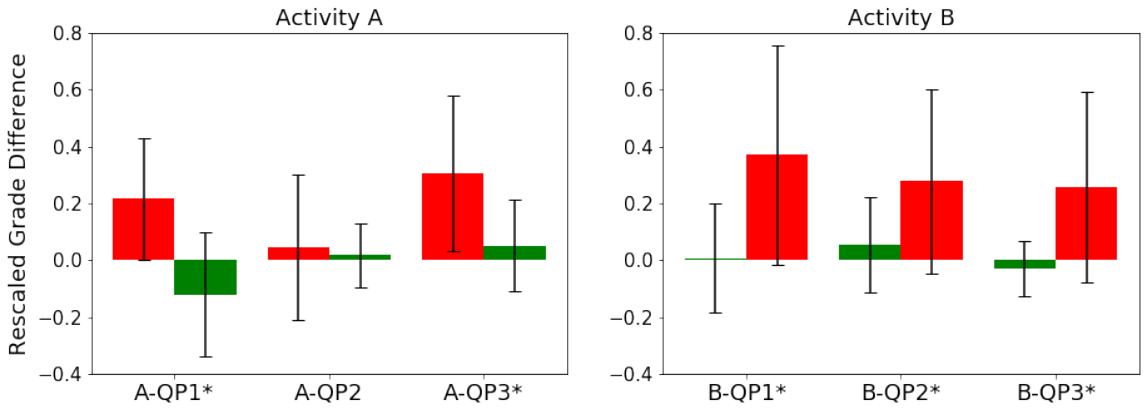


Figure 10. Comparison of Students' Grading Calibration in the Two Activities

Note: Students have a tendency to overgrade answers with more mistakes (red) than those with fewer mistakes (green); $^{*}p < 0.05$

Activity A ($r = -0.34$) and Activity B ($r = -0.56$), suggesting that those who did not overestimate did better in the exams.

4.5.3 The Role of Feedback

The following sections provide a discussion on how to answer **RQ C.4**. Particularly, these look into how soliciting feedback is associated with their midterm performance as well as their calibration in the grading process.

4.5.3.1 Students Who Actively Sought Feedback Did Better in Exams

The system provides two levels of feedback: (1) by showing the *actual marks* of the worked-out answer; and (2) by showing the *actual remarks* written by a subject expert (i.e., an explanation of why such grade was given or what was being looked for in the question). Students had to click a button to receive feedback. As noted earlier,

Table 9. Midterm Performance of Students According to Whether They Solicited Feedback From the System

Activity	Feedback	Proactive			Inactive			Hedges' g
		N	M	SD	N	M	SD	
A	Actual Marks	38	96.58	9.54	25	81.12	25.76	0.87
	Actual Remarks	32	96.44	10.32	31	84.26	24.03	0.66
B	Actual Marks	30	87.03	10.86	33	78.88	17.31	0.56
	Actual Remarks	26	88.46	9.88	37	78.76	16.81	0.67

Note: Activities A and B were administered prior to Midterms 1 and 2, respectively.

not everyone requested feedback. Students were grouped based on this behavior: the *proactive group* are those who actively sought feedback to at least one of the six worked-out answers in an activity, while the *inactive group* are those who did not. The midterm exam scores immediately following the activity (i.e., Midterm 1 for Activity A; Midterm 2 for Activity B) were compared. On average, the *proactive group* ($M = 92.34, SD = 10.96$) had a significantly higher midterm score than the *inactive group* ($M = 80.05, SD = 20.17$) with a Hedges' g effect size of 0.9. Additionally, Table 9 provides a breakdown of the performance on the two midterm examinations. These findings highlight the important role of feedback. Moreover, as students are exposed to erroneous examples, they must be guided throughout the process. The mistakes have to be pointed out to them, and correct answers have to be provided. This finding is encouraging as it shows the potential of this learning activity to help them prepare for their exams. It also trains them on how to debug a program, which is a critical skill one has to master to succeed in computer programming.

4.5.3.2 Validating One's Performance Led to Assigning Grades Closer to Expert's

Some students opted not to reveal what the *actual mark* was, a failure to validate their performance. What could be the underlying reason for such behavior? *Did the student already know what the answer was or what the assigned grade should be, based on his or her current understanding? Was the student simply doing the activity for the sake of compliance?* The quality of their grade assignments were looked into. Students who actively validated their performance had grade assignments that were closer to experts (i.e., scaled difference close to zero) ($M = 0.07, SD = 0.24$). On the other hand, those who did not validate their performance had a significantly higher difference on their grade assignments ($M = 0.20, SD = 0.33$) with a Hedges' g effect size of 0.46. This is more apparent in answers with more mistakes. Those who confirmed their gradings appeared to have more coherent and close-to-reality grade assignments similar to that of an expert, and vice versa. Choosing not to receive feedback would have been acceptable if they can grade properly. However, it seems that these students need guidance. This would have been a good opportunity to correct their misconceptions. This underscores the importance of student training as pointed out by (Li et al., 2020). This behavior needs to be addressed as it may affect students as they answer a similar question on their exam.

4.5.4 Subjective Evaluation

At the end of the semester, students were asked to answer an optional and anonymous survey to solicit their subjective evaluation of the system. Only 58 students (92.06%) of the 63 responded. The results to select questions are shown

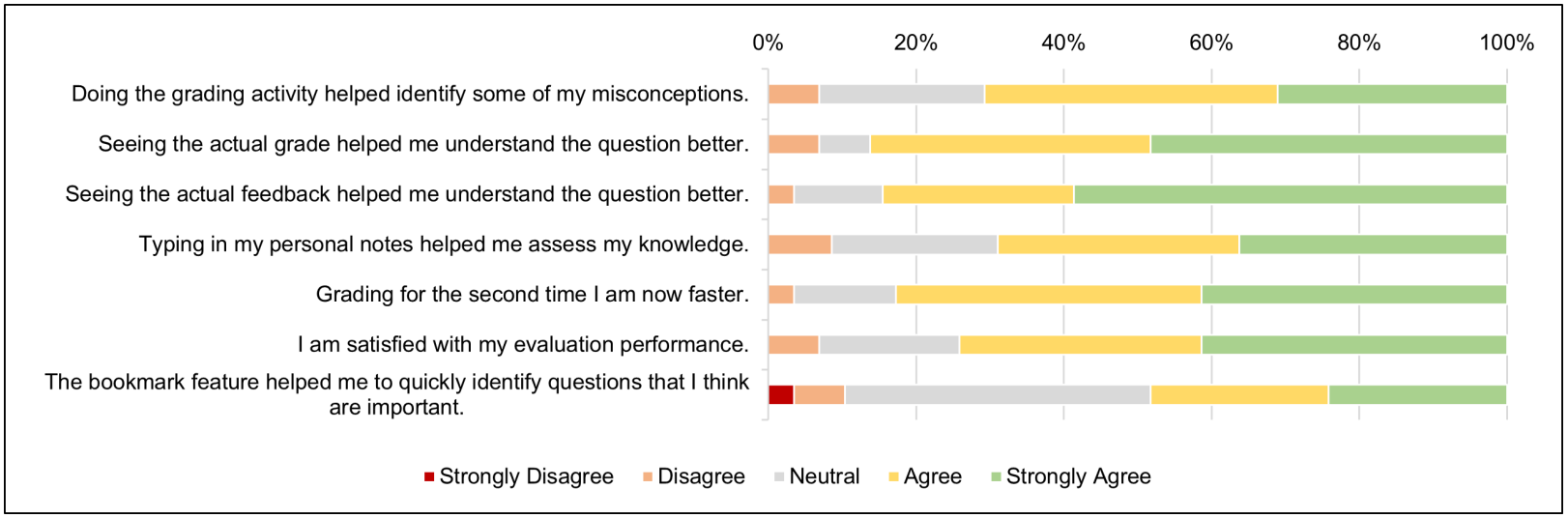


Figure 11. Select Survey Response of Students to an Anonymous Subjective Evaluation of Their Experience During the Grading Activity

in Figure 11. The majority (70.69%) answered that the grading activity has helped them identify some of their misconceptions. Furthermore, most of them said that the ability to see the actual mark (86.21%) of a worked-out answer, as well as the actual remarks (84.48%), has helped them understand the question better. Most students responded that typing in their justification for the grade they assigned to an answer has helped them assess their knowledge (68.97%) and that grading the same question the second time would be faster as they already know what to look for (82.76%). Overall, students were satisfied (74.14%) with how they graded the answers using the system. When asked which they think would help them prepare for an upcoming exam, most would prefer to see a past exam that contains both correct and incorrect answers (72.41%). Finally, 67.24% of the respondents indicated that they want to use the system in future courses, while 65.52% wanted to use the system when preparing for an exam.

4.6 Conclusion

This chapter presented an exploratory analysis of students' behaviors as they grade answers to programming questions from examinations of a previous offering of the same course. It was hypothesized that students could benefit from a learning activity that integrates peer assessments and worked-out erroneous examples, particularly in programming learning, as this closely resembles program debugging, a crucial skill in computer programming. This learning activity was performed on WebPGA prior to every midterm examination to help students prepare and review. To help students become familiar with the learning process, the system offers two levels of feedback: (1) by revealing the *actual marks*, and (2) the *actual remarks* given by a subject expert.

Since the activities were not mandatory, not all students completed the learning activities. Consistent with some studies, there was no significant difference in terms of exam performance between those who completed and those who did not. However, those who performed the activity had more coherent exam scores. Although the learning activity's impact is still inconclusive, a closer look at those who used the system highlighted the importance of feedback. Students who solicited feedback had significantly better scores in their exams. This suggests that being proactive in seeking feedback, especially when working with erroneous examples, could lead to a better learning outcome. Students would greatly benefit from this when consistently guided and trained.

In the future, a within-student analysis should be done to determine the impact of the learning activity or the system. Also, students were providing their notes while doing the activity. These notes can be leveraged to represent students' knowledge about a topic. Lastly, the analysis can be expanded to provide deeper insights to understand why students overestimate grades, especially those with more mistakes, and how this could be addressed.

4.7 References

Berg, E. C. (1999). The effects of trained peer response on ESL students' revision types and writing quality. *Journal of Second Language Writing*, *8*(3), 215–241.

Black, P., Harrison, C., & Lee, C. (2003). *Assessment for learning: Putting it into practice*. McGraw-Hill Education.

Brandt, J., Dontcheva, M., Weskamp, M., & Klemmer, S. R. (2010). Example-centric programming: Integrating web search into the development environment. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 513–522.

Chi, M. T. H. (2000). Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. *Advances in Instructional Psychology*, *5*, 161–238.

Griffin, J. M. (2019). Designing intentional bugs for learning. *Proceedings of the 2019 Conference on United Kingdom & Ireland Computing Education Research*, Article 5.

Hayes, J. R., Flower, L., Schriver, K. A., Stratman, J. F., & Carey, L. (1987). Cognitive processes in revision. *Advances in applied psycholinguistics* (pp. 176–240). Cambridge University Press.

Hsiao, I.-H., & Brusilovsky, P. (2011). The role of community feedback in the student example authoring process: An evaluation of AnnotEx. *British Journal of Educational Technology*, *42*(3), 482–499.

Ion, G., Martí, A. S., & Morell, I. A. (2019). Giving or receiving feedback: Which is more beneficial to students' learning? *Assessment and Evaluation in Higher Education*, *44*(1), 124–138.

Isotani, S., Adams, D., Mayer, R. E., Durkin, K., Rittle-Johnson, B., & McLaren, B. M. (2011). Can erroneous examples help middle-school students learn decimals? *Proceedings of the 6th European Conference on Technology Enhanced Learning*, 181–195.

Kamins, M. L., & Dweck, C. S. (1999). Person versus process praise and criticism: Implications for contingent self-worth and coping. *Developmental Psychology*, *35*(3), 835–847.

Kruger, J., & Dunning, D. (1999). Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, *77*(6), 1121–1134.

Li, H., Xiong, Y., Hunter, C. V., Guo, X., & Tywoniw, R. (2020). Does peer assessment promote student learning? a meta-analysis. *Assessment and Evaluation in Higher Education*, *45*(2), 193–211.

Magnus, J. R., & Peresetsky, A. A. (2018). Grade expectations: Rationality and overconfidence. *Frontiers in Psychology*, *8*, Article 2346.

Melis, E. (2005). Design of erroneous examples for ActiveMath. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 451–458.

Paredes, Y. V., & Hsiao, I.-H. (2021a). Can students learn from grading erroneous computer programs? *Proceedings of the 2021 International Conference on Advanced Learning Technologies*, 211–215.

Paredes, Y. V., & Hsiao, I.-H. (2021b). WebPGA: An educational technology that supports learning by reviewing paper-based programming assessments. *Information*, *12*(11), Article 450.

Paredes, Y. V., Huang, P.-K., & Hsiao, I.-H. (2019). Utilising behavioural analytics in a blended programming learning environment. *New Review of Hypermedia and Multimedia*, *25*(3), 89–111.

Patchan, M. M., & Schunn, C. D. (2015). Understanding the benefits of providing peer feedback: How students respond to peers' texts of varying quality. *Instructional Science*, *43*(5), 591–614.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction*, *12*(5), 529–556.

Rushton, S. J. (2018). Teaching and learning mathematics through error analysis. *Fields Mathematics Education Journal*, *3*, Article 4.

Serra, M. J., & DeMarree, K. G. (2016). Unskilled and unaware in the classroom: College students' desired grades predict their biased grade predictions. *Memory and Cognition*, *44*(7), 1127–1137.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, *2*(1), 59–89.

Topping, K. (2017). Peer assessment: Learning by judging and discussing the work of other learners. *Interdisciplinary Education and Psychology*, *1*(1), 1–17.

Tsovaltzi, D., Melis, E., McLaren, B. M., Meyer, A. K., Dietrich, M., & Goguadze, G. (2010). Learning from erroneous examples: When and how do students benefit from them? *Proceedings of the 5th European Conference on Technology Enhanced Learning*, 357–373.

VanLehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology*, *47*, 513–539.

Chapter 5

PRIME: INTELLIGENTLY RECOMMENDING APPROPRIATE ITEMS TO STUDENTS TO SUPPORT LEARNING FROM OTHER'S MISTAKES

In the modern classroom, systems for capturing student performance data have become increasingly available. Over the years, vast amounts of student data have been collected which provide an abundance of information just waiting to be discovered. By construing this as a collection of students' learning experiences, future students will be able to gain insight from this information. The size of this dataset makes it difficult for an individual to navigate through and explore it independently. Therefore, personalized recommendations are desired to tailor to the user's needs, as is the case with recommender systems. Two major types of recommendations are often made by technology-enhanced learning systems: those that enhance knowledge and those that provide remedial advice (Bauman & Tuzhilin, 2018). Recommendations from the former group aim to expand the student's knowledge by suggesting the next steps. In contrast, recommenders belonging to the latter group are concerned with filling in the student's gaps.

From the results of Chapter 4, it was found that when students were asked to evaluate an erroneous example, some sought feedback by reviewing the actual score or comments provided by an expert. This process results in the development of a student's evaluative judgment, which ultimately leads to an improvement in the calibrator's accuracy. Eventually, students became familiar with the process, which ultimately resulted in them taking less time to complete the activity. It was unclear how exactly students learn from such activity, or at all. This is because there was no difference

between the test performance of students who completed the activity and those who did not. The problem may be due to a lack of consideration for students' needs. In an introductory programming course, some students possess greater knowledge than others. Like in the literature of worked examples, the value of viewing such examples diminishes as an individual develops expertise (Kalyuga, 2007; Kalyuga et al., 2001). A problem-solving activity would be more beneficial to these students. In contrast, novices would benefit from examples that are on par with their level of mastery. The recommendation should therefore consider the student's current level of mastery.

During the learning process, guidance is the process by which a learner is directed or influenced toward taking certain actions. It is something that should be inherent in educational systems. Every learner has their own unique motivation, goals, preferences, knowledge level, interests, and traits (Brusilovsky, 1996). The design of such systems using a one-size-fits-all approach may be ineffective. It should be able to provide learners with tailored guidance and adapt to their individual needs. A system such as this is called an adaptive environment since it monitors the learner and makes the necessary adjustments to enhance learning (Shute & Zapata-Rivera, 2012). Adaptations can be expressed on a micro-, macro-, and meta-level (Folsom-Kovarik et al., 2019). To provide effective guidance, it is necessary for a variety of components to work in concert.

Based on the findings in Chapter 4, the present study aims to customize the selection of erroneous examples that students will evaluate. To accomplish this, a recommendation engine that analyzes multiple pieces of information could be incorporated to determine the most appropriate item. Specifically, it takes into account the student's deficiencies. As in the previous study, students are assisted in preparing for a forthcoming test. By using minimal information from teachers

regarding the blueprint of an upcoming test, the system seeks to predict each student's item-by-item performance. Based on these predicted performances, erroneous examples (i.e., anonymous answers of a different student) will be presented for the student to evaluate to learn from the errors of others. To make students aware of their own weaknesses, items that resemble their own performance are provided to them. The present study hypothesizes that students will be made aware of their misconceptions when a personalized recommender can intelligently provide them with an erroneous example for them to evaluate. This will prepare them for further assessment. It will be necessary to develop a system for forecasting one's performance to achieve this. While this task is similar to those that have been performed in other research, the current context prevents it from using existing methods.

While this chapter addresses the limitations identified in the previous chapter, the evaluation focuses primarily on the framework's feasibility. An independent study, such as a Wizard of Oz study, is believed to be able to test the learning effect. Thus, the purpose of the chapter is to explore and investigate how past student performance data can be utilized to provide appropriate learning exemplars for future students. Specifically, it aims to answer the following research questions:

RQ D.1: Using performance data on complex multi-topic test items from a classroom setting, how can the growth of the mastery level of students be modeled?

RQ D.2: How can a student's outcomes on individual items be predicted on a test that contains items that allow partial credit?

RQ D.3: Having knowledge of the potential outcomes of a test, what innovative learning opportunities, particularly in the domain of computer programming, can be provided to students to assist them in preparing?

This chapter is organized as follows. Sections 5.1 to 5.2 discuss the motivation for the proposed learning activity and discuss various techniques for predicting student performance. In Section 5.3, the dataset used for the experiment is described in detail, followed by a comprehensive introduction of the PRIME framework in Section 5.4. Afterward, Sections 5.5 to 5.6 provide a brief overview of the experiment and the evaluation results.

5.1 Evaluating Erroneous Answers as a Learning Opportunity

Based on a survey among teachers of introductory computer programming, it was found that students are given access to past tests to prepare for an upcoming test (Sheard et al., 2013). In some classes, the teacher may even provide students with copies of these tests directly. In other instances, students utilize online services such as Chegg to obtain copies of these tests illegally. In addition, it was found that teachers tended to use test questions from past examinations and slightly revise them. The likelihood of obtaining questions from textbooks and the internet is low. These findings suggest the value of reviewing old versions of the tests. The act of simply providing access to students may be considered a passive activity. Based on the ICAP framework, if students are engaged in learning rather than passively receiving information, they will have a more meaningful learning experience and ultimately learn more (Chi & Wylie, 2014). Data collected in the past can be repurposed to assist a new cohort of students in preparing for a forthcoming examination. When students have the opportunity to experience a similar task before taking an actual test, they not only gain a greater understanding of the topic but also develop a greater degree of confidence in their ability to accomplish it or increase their self-efficacy (Bandura,

1997). The motivation of students for performing an activity is typically determined by the expected benefits they will receive (Wigfield & Eccles, 2000). For example, students tend not to value such additional activities if they already know the material. Due to this, students should be provided with a level of difficulty that is appropriate to their current level of mastery. Lastly, the supplementary activity has practical value for students since they can gain knowledge from it and see possible test items that may appear on the test.

5.1.1 Benefits of Peer Assessment

This type of learning opportunity may be viewed as an opportunity for learners to evaluate the work of their peers, known as peer assessment. It offers students a variety of opportunities to practice and integrate what they have learned geared toward developing a mastery of the domain (Ambrose et al., 2010). Literature has pointed out the educational benefits of the activity, particularly for those who perform the evaluation, as it fosters evaluative judgment and enhances judgment accuracy or calibration (Nicol et al., 2014). It facilitates the feedback-giving process, especially when students attempt to troubleshoot the work of a peer or even to offer solutions, which is a cognitively engaging activity based on the ICAP Framework (Chi & Wylie, 2014). Through exposure to numerous works of varying quality, students gain an understanding of the abstract concept of quality (Sadler, 2010). Students can compare their mental solution to the problem at hand with that of the other student (McConlogue, 2015). This allows them to evaluate their own work by comparison, thereby strengthening their internal feedback mechanism. It is helpful for students to observe the work of others to gain an understanding of alternative strategies,

such as another approach to solving the problems (Atkinson et al., 2000). To fully benefit from the process, students must first be trained in the process (Nicol, 2021). Overestimation of performance is common among students. It is necessary to devise interventions to improve a student's calibration. Knight et al. (2022) found that when students improved their calibration or even underestimated, their performance on the subsequent quiz improved.

5.1.2 Benefits of Worked Examples

Seeing other people's answers can also be construed as worked examples. Providing students with worked examples is one of the ways to help them develop strategies, especially when they are learning new skills (McLaren & Isotani, 2011; Renkl, 2014). When viewing actual student answers, students will find that most contain errors. These examples will therefore be considered erroneous if they are provided to other students. In recent years, research has explored the benefits of exposing students to erroneous examples, particularly in mathematics (Isotani et al., 2011; Melis, 2005; Tsovaltzi et al., 2010) or computer programming (see Chapter 4). This learning exercise mimics the process of "programming debugging", an intricate and challenging activity requiring a strategy (Winslow, 1996), which is a crucial skill in the field of computer programming. There is a need to assist students with debugging errors and explain how to correct them (Pillay, 2003). By studying other people's mistakes, students can avoid committing the same mistakes themselves (Ohlsson, 1996). By recognizing their own mistakes or those of others, students can reflect on their own performance and make the necessary adjustments. This develops metacognitive skills in students, such as self-monitoring. As such, it facilitates the processes of assimilation

and accommodation as they attempt to make sense of information that appears at odds with their schema or existing knowledge (Piaget & Inhelder, 1969). Additionally, this activity aligns with the main principles of social learning theory, where learning is seen as a social process (Bandura, 1977). Learning occurs by observing others' behaviors. Specifically, it argues that students learn by imitating others' actions and then observing the consequences of those actions. It may also trigger students to encounter prediction errors as they encounter mistakes, resulting in a difference between expected results and actual results. Learning is believed to result from this process. The past answers of students can be construed as a collection of different approaches to answering a question. Students can replicate or dismiss such behaviors based on what they observe, for instance, if they observe how certain answers are being rewarded or penalized on a test. The owner of the incorrect answers corresponds to what Bandura (1977) refers to as a model, whereas the incorrect answers are what is being observed. The outcome is the grade received for the answer. Thus, students may be able to gain knowledge from such vicarious experiences and even increase their self-efficacy as a result (Bandura, 1997).

5.2 Student Performance Prediction

The learning activity aims to provide students with examples that are tailored to their needs. As a matter of fact, this is one of the guidelines for designing and building effective adaptive and intelligent systems (Brusilovsky, 1996, 2001). What is an appropriate example? Since the purpose of this activity is to make students aware of their misconceptions, an appropriate example is one that addresses topics on which the student has deficiencies. Learning is a mental process, so knowing

these deficiencies can only be determined by understanding the student's domain mastery. Additionally, these measurements can only be determined from observables, in this case, the results of the test. Moreover, because the learning activity includes identifying the student's performance in the future, the task can also be framed as a prediction of student performance. An overview of some approaches that can be employed to identify student strengths and weaknesses is provided in this section.

Student performance prediction has become a well-researched area as student data has become more prevalent and machine learning has become more popular. Students' performance is measured both quantitatively and qualitatively (Brahim, 2022) with varying levels of granularity. The degree of granularity is significant as it provides more information and could greatly affect the intervention. Researchers have previously identified students at risk, identified appropriate next tasks, predicted completion rates, and predicted performance on advanced courses, among other things (Bauman & Tuzhilin, 2018; Romero & Ventura, 2020). Most of the literature predicts summative performance metrics such as course grades and exam scores; however, the concept of "knowledge gain" is gaining a lot of popularity (Hellas et al., 2018). Most studies use statistical and machine learning methods, with few using a combination of both. These models rely on various features, attributes, or characteristics of the student to make these predictions. The performance of students is thought to be influenced by a variety of factors. Many factors are involved, ranging from demographics to student behavior (Khan & Ghosh, 2021). Many studies use demographic data, some use performance data (from either the current or previous course), some use characteristics (self-efficacy), and some use engagement data (such as time on task). There are also works that combine these factors.

It is possible to categorize these works based on their objectives, as illustrated in

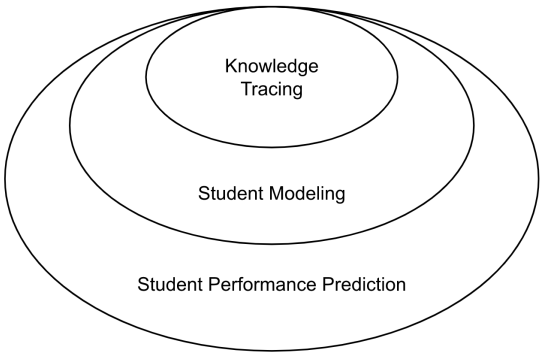


Figure 12. Synthesizing and Categorizing the Various Works on Student Predictions

Figure 12. Student performance prediction is the umbrella term that encompasses everything. Work that incorporates individual student characteristics is classified as student modeling. Attributes may include cognitive, emotional, and self-regulation characteristics. Finally, knowledge tracing encompasses works that focus on knowledge and mastery.

5.2.1 Student Modeling

Self (1990, as cited in Chrysafiadi & Virvou, 2013) emphasizes the importance of effective student models. In any adaptive education system, models are an essential component (Pelánek, 2017). It aims to capture various aspects of students to obtain a holistic picture of them. It is possible to collect information directly from the student or indirectly through proxy data, such as interactions, that could assist in developing the model. Using these data, latent characteristics of students, such as mastery or knowledge, can be estimated. The data is used to provide tailored pedagogical interventions and suggest appropriate learning resources for the students. Multiple techniques have been proposed and incorporated into existing educational

systems over the years. Table 10 combines and summarizes the current approaches to developing student models as surveyed by Chrysafiadi and Virvou (2013) and Zapata-Rivera et al. (2020). In conjunction with overlay student models, which are commonly used to represent students' mastery of a domain, this study leverages fuzzy and machine-learning methods.

5.2.2 Knowledge Tracing

As far as the granularity of student performance is concerned, relying solely on overall scores provides only limited insight into a student's strengths and weaknesses. Estimating a student's mastery of a subject domain is a common task associated with student modeling. Having a clear understanding of the student's current mastery level, particularly of a knowledge component, enables the system to provide appropriate suggestions or actions to maximize the student's learning. An example of this would be the system looking at a student's previous responses to determine whether they are ready to learn a new concept. The process of knowledge tracing is used to accomplish this. Knowledge tracing is extensively studied in EDM and AIED research, particularly in order to determine what students know and does not know. A knowledge tracing task is divided into two major components: the estimation of knowledge and the prediction of response (N. Liu et al., 2022). According to Khan and Ghosh (2021), most EDM research focuses primarily on success prediction using either regression or classification. There are very few studies that focus on predicting actual marks, as 46% of the 140 surveyed focused on success prediction, 35% on final grades, and 19% on score prediction. The purpose of the current study is therefore to make predictions at the item level or score predictions. In this manner, the strengths and weaknesses of

Table 10. Overview of Various Approaches to Developing Student Models

Approach	Description
Overlay model	Commonly used to represent mastery of topics where the student model is a subset of the domain model. This alone cannot make inferences of knowledge integration with prior knowledge of the learner, therefore has to be combined with other approaches.
Stereotype model	Groups students based on certain characteristics that are shared. Allows the system to start quickly. Prone to errors and requires human intervention for updates.
Perturbation model	Extension of overlay model which includes possible misconceptions (i.e., mal-knowledge). Leads to better remediation and development of bug library (i.e., buggy model). However, developing the bug library is a laborious process.
Machine learning	Uses learning algorithms to infer behavior through various characteristics or from the data. Multiple observations enable the development of predictive models to predict future actions.
Cognitive theories	Relies on various established theories on human cognition that attempts to explain behavior during the learning process.
Constraint-based	Uses constraints to represent both domain and student knowledge. Tends to be computationally simple as it does not require an expert module.
Fuzzy model	Assumes that student modeling is a complex and non-straightforward task. Capable of handling uncertainty.
Bayesian networks	Uses directed acyclic graphs representing probabilistic dependence or causal relationships among variables. Capable of handling uncertainty using probabilities. One limitation is it is time-consuming to make.
Ontology-based	Supports representation of abstract concepts and properties to facilitate reusability and extendability in different application contexts.

Table 11. Overview of Various Approaches to Knowledge Tracing

Approach	Description
Probabilistic[a]	Mastery of a topic is estimated based on historical performances or practice opportunities and by including parameters for guess, slip, learning, and unlearning.
Logistic	Diagnose mastery by applying a logistic function on estimated parameters.
Deep Learning	Uses deep learning algorithms to derive predictive models based on student performance.
Others	Approaches from other fields such as psychometrics and combinatorics.

[a] As in Q. Liu et al. (2021) but Markov processes in Gervet et al. (2020).

a student can be more precisely identified. The next step is to survey and identify existing approaches that attempt to accomplish a similar objective.

There are already a variety of approaches in the literature. In fact, a taxonomy has recently been proposed to group the various models. Based on a technical perspective, Q. Liu et al. (2021) classified models into three categories: probabilistic, logistic, and deep learning. Gervet et al. (2020) presented a similar classification within the categories of Markov processes, logistic regression, and deep learning. In addition, these families of models were evaluated on nine real-world datasets to identify the characteristics and properties that contributed to their performance. In addition to the three categories identified earlier, knowledge tracing is also evident in the Psychometrics literature, which is noteworthy. A summary of the different approaches can be found in Table 11.

5.2.2.1 Probabilistic

In the probabilistic approach, it is assumed that student learning follows a Markov process. The latent knowledge state can be estimated from the observed performance. A common example is the Bayesian knowledge tracing (BKT; Corbett & Anderson, 1994), which is a special case of a hidden Markov model. It has two parameters, namely the transition and the emission probabilities.

5.2.2.2 Logistic

In the logistic approach, various factors associated with learning interactions are leveraged to estimate both the student and knowledge component parameters. A logistic function is used to convert these estimates into probabilities of students' mastery. The popular examples are the learning factor analysis (LFA; Cen et al., 2006) and performance factor analysis (PFA; Pavlik et al., 2009). In PFA, performance on the current item is predicted based on success and failure on previous items that address the same knowledge components. In AFM, the probability of success is proportional to the combination of the student's ability, the difficulty of skills associated with the item, and the amount of learning gained from each attempt. On the other hand, PFM improves on AFM that it includes evidence of learning from previous attempts while discarding student ability altogether. An example in Thaker et al. (2020), PFA was used alongside knowledge states to identify the state of the student and provide recommendations for remedial readings in an online reading platform to provide an adaptive recommendation.

5.2.2.3 Deep Learning

In this approach, deep learning techniques are exploited to develop models that can directly learn from data. It can learn non-linear relations as well as perform feature extraction. One prominent example is deep knowledge tracing (DKT; Piech et al., 2015). Another example that uses deep learning models is the open-ended knowledge tracing framework which is capable of estimating students' mastery of computer science concepts and generating a code that is predicted to be the output of the student (N. Liu et al., 2022). However, their work is focused solely on computer programs. Another is Knowledge proficiency tracing (KPT; Y. Chen et al., 2017) which borrows from the idea of knowledge space and matrix factorization while incorporating the theories on learning and forgetting. Notably, a common limitation of employing a deep learning approach is the interpretability of such models.

5.2.2.4 Others

Item Response Theory (IRT) is a family of approaches aiming to uncover a latent trait based on item difficulty and student ability interaction. It has several variants, particularly regarding the number of estimated parameters (e.g., Rasch, 2-PL, 3-PL). One issue often associated with the classical IRT is the unidimensional nature of the latent trait. To get a granular idea of students' mastery, it is necessary to go beyond an overall latent trait. Extensions to IRT began supporting multiple latent traits (e.g., MIRT). Recently, one work compared the prediction performance at the item level of IRT to machine learning methods (Park et al., 2022). Despite the promising results of their explanatory IRT models, the dataset used was limited to dichotomous items.

Again, in introductory computer programming classes, tests often emphasize code writing where multiple concepts are being dealt with by the student simultaneously (Daly & Waldron, 2004).

Cognitive Diagnostic Model (CDM) is a family of approaches that can attempt to uncover latent subskills of students based on their item performance. In a way, it addresses the limitation of IRT by supporting multiple subskills instead of assuming a single latent trait. One example is G-DINA (de la Torre, 2011). Information about the item using a Q-matrix is used to estimate students' mastery based on their observed performances. However, unlike IRT, where a student is described by a continuous latent trait, in CDM, a student is characterized by a vector of discrete latent traits that represents their mastery of the various subskills. It is worth noting that researchers have proposed a framework for polytomous latent subskills by extending current approaches (J. Chen & de la Torre, 2018).

Knowledge Space Theory (KST) is another approach to assessing student knowledge based on combinatorics (Doignon & Falmagne, 1999). Arguably, this one is fairly complex and inaccessible among all the surveyed approaches. The core idea is the determination of the student's knowledge state, which is a subset of problems from a domain that the individual is capable of solving (Falmagne et al., 1990). The collection of knowledge states is known as a knowledge structure, of which knowledge space is a special kind. Knowing a student's knowledge state allows for inferences on efficient assessment procedures such as identifying which problem to solve based on the student's knowledge boundary (Wang et al., 2017). One prominent example that leverages this approach is Assessment and Learning in Knowledge Spaces (ALEKS)[2].

One technique that has been done in the student performance prediction literature,

[2]https://www.aleks.com/

particularly for student modeling, is matrix factorization (Thai-Nghe et al., 2011; Thai-Nghe et al., 2012; Thai-Nghe & Schmidt-Thieme, 2015). Matrix factorization is popularly used in the recommender system area and has gained popularity during the Netflix prize time (Koren et al., 2009). The objective is to factorize a given matrix (i.e., ratings) into two, resulting in uncovering the latent attributes of two entities (i.e., users and movies). One advantage of such an approach is that it implicitly encodes parameters such as slip and guess (Thai-Nghe et al., 2012). Their earlier work contextualized matrix factorization to a performance prediction problem given the following correspondences: ratings-performance, users-students, and movies-questions (Thai-Nghe et al., 2011). An extension to such work was the incorporation of biases to account for student effects and question effects. Additionally, they accounted for the temporal component of the data that resulted in the notion of tensor factorization (Thai-Nghe et al., 2012). Finally, by supporting the ability to incorporate knowledge of multiple relationships and adjust their importance, it was possible to integrate these into the prediction process (Thai-Nghe & Schmidt-Thieme, 2015). This means that in addition to student performance data, information about an item can be integrated into the process. For example, if we have prior knowledge regarding the knowledge components associated with a question, we can utilize this information to improve the model's performance. It can also incorporate information about whether the student has mastered a required knowledge component. Such an approach was found to be simpler and performed better than the Bayesian Knowledge Tracing approach when they compared the root-mean-square error (RMSE) (Thai-Nghe & Schmidt-Thieme, 2015).

5.2.3 Limitations of Existing Models

Most previously surveyed models were mostly designed for dichotomous items (i.e., correct or incorrect) that are typically associated with multiple-choice questions. Using this information, such models can predict whether the student can answer a different question correctly. Although this approach can also be leveraged in the context of the present work, in reality, most of the teacher-made questions are not necessarily binarily graded. Given the complex nature of the questions employed in paper-based tests, teachers often give partial credits to students to reflect the severity of the student's mistake. Simply reducing these continuous values to a binary value would result in a loss of information or context that would otherwise provide critical information, for example, a question's complexity or difficulty (Pelánek, 2017). There is also some work that deals with polytomous responses going beyond binary correctness, such as option tracing models (An et al., 2022; Ghosh et al., 2021). These model student responses and predict which among the choices the student is likely to select as an answer. In a way, this is similar to what this present work seeks to accomplish. However, these predictions are mainly for multiple-choice questions. Most questions are open-ended for teacher-made tests, particularly in computer programming; thus, these models cannot be directly applied to the current context. Although there is work that attempts to forecast student code generation, the approach is limited only to program code (N. Liu et al., 2022). There are still other types of questions beyond it. Therefore, it is critical that the modeling accounts for partial grading. In this study, student performance is normalized to account for the varying question format and their varying points. The raw score is divided by the total points possible, resulting in

a value in the range of zero and one, inclusive. This is then construed as the student's partial grade for that particular question.

In terms of the results of the existing models, most of those focus mostly on predicting a student's success in future tasks. These probabilities often are used to make the necessary adaptations for the student. For example, if the policy follows Vygotsky's (1978) zone of proximal development, the system will most likely select an appropriate task where the success rate is optimal. However, in the present work, the dataset contains complex questions that support partial grading. Defining what success means becomes a subjective task requiring experts to label it manually. To the best of our knowledge, only limited work has been done on predicting student performance at a finer-grained level where the normalized score at the item level is predicted. When dealing with normalized scores, the given illustration can still be accomplished as these normalized scores are directly comparable to each other. Additionally, the probability of success cannot be easily translated to a normalized score as they are two different constructs. This means if a model outputs a probability of success of $P(S) = 0.75$, certainly, it does not follow that the correctness of the student's work is 0.75 as well. Given the nature of the current dataset (i.e., performance scores), obtaining a probability value will be ambiguous. Furthermore, normalized scores are easier to interpret than success probability as the former is concrete, which could lead to actionable findings. Although there is some work that attempts to reverse the process where evidence from the dataset is estimated based on the success probability, this is only done in Bayesian models (Keith, 2021). As one of the system's objectives is to identify similar performances of students from the past, it is critical that the metrics involved in the process are comparable. This suggests that it is necessary to devise another mechanism, which this work proposes as a novel contribution.

Another limitation of the existing models is the manner in which the data is received. In particular, these data are received one at a time, depending on the level of granularity. For example, when a student performs a step, the model interprets it to determine what steps should be performed next (VanLehn, 2006). However, the context of exams, particularly summative ones, involves students answering multiple questions in a single session before feedback is provided. Due to the nature of the exam, typically it is not possible to provide multiple opportunities for independent practice, unlike in other systems. Due to these factors, leveraging the learning curve plot can be challenging. Thus, it is necessary to consider such factors as the order in which the model receives input will determine how the model estimates the student's mastery. This will help ensure that the proposed solution is feasible in the current context. As has been noted by Pelánek (2017), certain learner modeling approaches are not fundamentally appropriate for detecting mastery. However, they are helpful to gain insights for offline analyses.

5.3 Educational Assessment Dataset

This section provides an in-depth discussion of how the dataset used for the student modeling was curated. The dataset was obtained from an Introductory Computer Programming course offered to information management students from the years 2018 to 2020. This course has been taught by the same instructor and followed the same syllabus. Three exams were administered each semester covering a predefined set of topics. Due to the nature of the domain, the coverage of the topics was cumulative. Recent topics are given higher importance in terms of point assignments. In total, there were nine exams in the dataset. Table 12 provides an overview of the dataset.

Table 12. Descriptive Statistics of the Examinations Across the Years

	2018 ($N = 124$)			2019 ($N = 123$)			2020 ($N = 124$)		
	E1	E2	E3	E1	E2	E3	E1	E2	E3
Max Points	100	100	105	100	93	100	100	89	100
No. of Questions	18	21	21	18	18	19	16	12	15
No. of KCs	39	53	43	40	50	49	40	35	43
No. of Attempts	126	124	124	126	124	123	128	127	126
Average Question Difficulty	0.74	0.68	0.80	0.84	0.72	0.73	0.86	0.72	0.77
Point Contribution									
Dichotomous	0.24	0.30	0.43	0.30	0.39	0.45	0.30	0.27	0.30
Polytomous	0.76	0.70	0.57	0.70	0.61	0.55	0.70	0.73	0.70
Topic Distribution									
01 - intro	0.06	0.03	0.02	0.02	0.08	0.07	0.02	0.06	0.05
02 - fundamental data type	0.56	0.08	0.10	0.51	0.12	0.06	0.59	0.19	0.15
03 - decision	0.20	0.11	0.08	0.29	0.04	0.02	0.16	0.07	0.04
04 - loop	0.18	0.07	0.04	0.18	0.04	0.03	0.23	0.07	0.02
05 - methods	-	0.22	0.10	-	0.13	0.03	-	0.08	0.07
06 - arrays and array list	-	0.21	0.07	-	0.19	0.06	-	0.17	0.02
08 - objects and class	-	0.28	0.33	-	0.39	0.38	-	0.37	0.25
09 - inheritance and interface	-	-	0.28	-	-	0.35	-	-	0.40
Cumulative Point Distribution									
01 - intro	0.55	0.82	1.00	0.12	0.58	1.00	0.17	0.58	1.00
02 - fundamental data type	0.76	0.86	1.00	0.74	0.91	1.00	0.65	0.84	1.00
03 - decision	0.51	0.79	1.00	0.83	0.94	1.00	0.62	0.85	1.00
04 - loop	0.63	0.86	1.00	0.72	0.88	1.00	0.74	0.94	1.00
05 - methods	-	0.69	1.00	-	0.80	1.00	-	0.50	1.00
06 - arrays and array list	-	0.75	1.00	-	0.74	1.00	-	0.88	1.00
08 - objects and class	-	0.44	1.00	-	0.49	1.00	-	0.57	1.00
09 - inheritance and interface	-	-	1.00	-	-	1.00	-	-	1.00

Note: N indicates the number of students who took all the three examinations.

5.3.1 Data Collection and Processing

The number of student attempts is presented for each exam. A few students had to be removed from the dataset due to them not taking the three required exams. Only students who were able to take the three exams were retained and are indicated below the year in Table 12. There were a total of 158 unique questions in the dataset. It should be noted that, as mentioned in the literature, there are instances where questions from previous years were reused (Sheard et al., 2013). In this dataset, even if two questions are very similar or identical, they were counted separately.

Whenever a new test is created, the instructor uses the Test Authoring interface of the system (Figure 13) to provide the following information: the question and its corresponding KCs. The questions take various forms, such as multiple-choice, program tracing, or program writing. Multiple-choice questions are often graded in a binary fashion, while the others involve partial grading. In complex questions, multiple KCs are often assigned. Along with the KCs are their corresponding subjective weights (i.e., score). For this particular class, all questions were written by a single instructor and were typically based on the main textbook reference specified in the syllabus. The exams were designed to be answered for two hours on paper without external references. Consistent with the findings of Sheard et al. (2013), the questions written contain a mix of various types, mainly composed of multiple choice, filling in the blanks, and code writing, thereby suggesting that multiple KCs were being assessed in several questions. The perceived importance of a KC to successfully answer a question is determined based on the points assigned by the instructor.

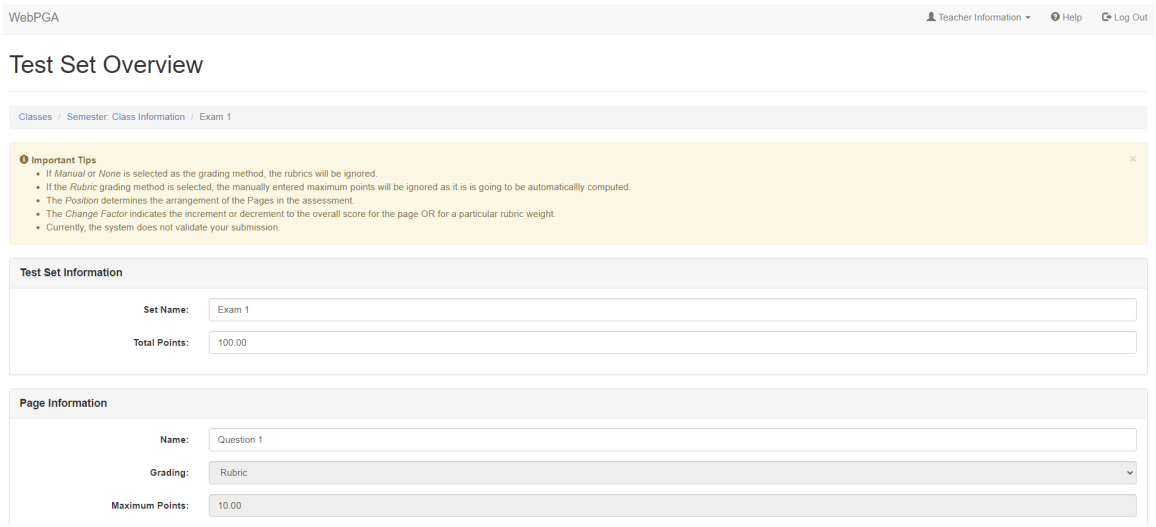


Figure 13. Test Authoring Interface Used for Creation of New Assessments

5.3.2 Using Human Judgment for Topic Labeling

Figure 14 illustrates an example of a student's work being assessed using a predefined set of KCs and their associated weights. Despite the tests being written by a single instructor, the terminology used for the KCs had slight variations. Some terms were inconsistently used although referring to the same concept. Given the varying granularity of the KCs, a common level for the 392 KCs had to be identified for the analysis. Thus, a coarse-level knowledge unit was determined as a *topic*. Human judgment was leveraged to identify which topic a particular KC was associated with. The course covered eight major topics based on the syllabus and the reference book *Java for everyone: Late objects* by Horstmann (2013), namely:

- 01 - intro
- 02 - fundamental data type
- 03 - decision
- 04 - loop
- 05 - methods
- 06 - arrays and array list
- 08 - objects and class
- 09 - inheritance and interface

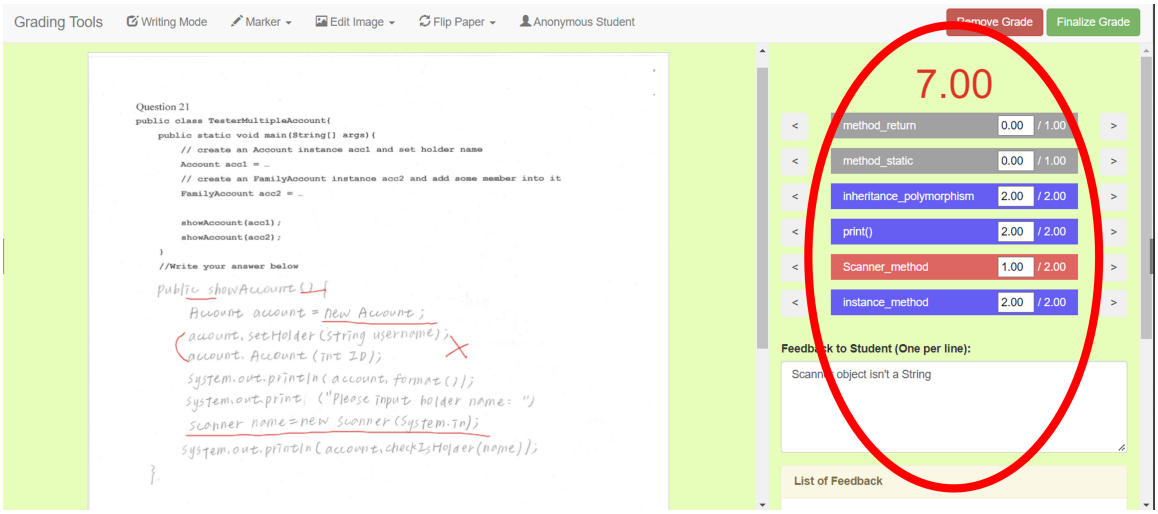


Figure 14. Example of Student Work Evaluated Using a Set of Predefined KCs

It is worth noting that these topics are similar to that found by Tew and Guzdial (2010) when they surveyed both the literature and several introductory programming courses across various universities as they attempted to develop a guide for developing validated assessments.

To validate the process, two subject-matter experts with prior teaching experience (10 and 14 years) independently analyzed the 392 rubrics and assigned each to a single topic. After the initial pass, the two experts compared their labels. Any disagreements were resolved through discussion until a consensus was reached. This re-labeled data was used for the study. The proportion of the topics being assessed in a test is summarized in Table 12. As expected in a typical introductory computer programming course, advanced topics build upon a solid understanding of fundamental topics. Therefore, it is expected that exams early in the semester do not have complete coverage of all topics.

5.3.3 Consistency of Exam

Although a single instructor has been handling the same course over the years, it is worth investigating whether the topic coverage of these tests was consistent over time. One approach to measuring consistency is by comparing the distribution of the various test at a given time to their corresponding test from a different year. This means the distribution of topics of *Exam 1* from *Year 1* will be compared to that of *Exam 1* from *Year 2* and so on. This can be done through the Kolmogorov–Smirnov test. It is a nonparametric test that allows testing whether two samples were drawn from the same probability distribution, thereby quantifying the difference. For a two-sample test, the general idea is to obtain the cumulative frequency distribution for each sample and get the absolute difference. Afterward, the maximum absolute difference becomes the $D_{\text{statistic}}$. As with other hypothesis tests, the next step is to obtain the D_{critical} value using the following formula for confidence level $\alpha = 0.05$:

$$D_{\text{critical}} = 1.36 \sqrt{\frac{n_1 + n_2}{n_1 n_2}} \tag{5.1}$$

where n_1 and n_2 represent the sample sizes. A detailed computation can be found in Appendix A. Given the computed values, it can be seen that the instructor has consistently written and used similar tests when assessing students.

5.3.4 Descriptive Statistics

The following section provides some descriptive statistics regarding the dataset. It gives an insight into the complexity and difficulty of each question based on the number of topics associated with each question and the composition of the types of questions. Also, it provides an overview of the students included in the dataset.

5.3.4.1 Question Complexity

One way of measuring question complexity is based on the number of associated topics. On average, each question is associated with 1.85 ($SD = 1.50$) topics. Another approach is to look into the number of KCs associated with each question. On average, each question is associated with 4.96 ($SD = 4.83$) KCs. Another approach is to look into the total points associated with a question. On average, each question has a total possible score of 5.61 ($SD = 4.53$) points. The correlation between the number of associated topics to a question and the point association was looked into where a significant positive correlation ($r = 0.80$) was found. Unsurprisingly, this suggests that the more topics being assessed in a question, the more points are at stake. Table 12 provides a detailed breakdown of each exam. Finally, a metric was added to enable questions to be compared based on the points assigned and the number of KCs being assessed. This metric, denoted by ϕ is computed by dividing the maximum points associated with a question by the number of KCs associated with the question.

5.3.4.2 Question Difficulty

The difficulty of a question can be estimated using the performance of the student who answered it, defined mathematically as the proportion of those who answered the item correctly (van de Watering & van der Rijt, 2006). Essentially, it is a normalized value that ranges from 0 to 1 where a value close to 0 indicates that the question is very difficult. In fact, some studies suggest using the term *easiness* due to the direction of the values (Pavlik et al., 2009). Due to the presence of partial credit, the normalized score is used instead of a simple count of the number of students

who correctly answered an item. A breakdown of the average question difficulty for an exam is provided in Table 12. A Kruskal–Wallis test was performed to check the consistency of the average question difficulty of each exam belonging to different semesters (i.e., Exam 1 over the three years, and so on). No significant difference was found (Exam 1, $p = 0.16$; Exam 2, $p = 0.23$; Exam 3, $p = 0.24$) suggesting that there was consistency over the years. The average difficulty on all 158 questions from the three years taken by 371 students is 0.76 ($SD = 0.17$).

5.3.4.3 Grading Assignment and Point Contribution

As in the literature, how questions are graded plays an important role in the modeling approach. The following provides a breakdown of the number of questions that were graded dichotomously (i.e., correct or not). On average, in a given examination, a proportion of 0.33 of the total points is from dichotomous questions while 0.67 is for polytomous which necessitates partial grading. Table 12 provides a breakdown of the proportion for each exam.

5.3.4.4 Cumulative Point Distribution

Given the cumulative nature of the number of topics being assessed in this course over time, it was necessary to determine the distribution of each topic at a given time point. Essentially, this value becomes the ceiling value a student can possibly attain for that topic at that given time point. For example, during the *E1* time point of *2018* as shown in Table 12, a student can obtain a maximum of 0.55 for the *01 - intro* topic, indicating that the student has fully mastered the available material at that

given moment if the student obtains a level of 0.55. However, since it is still early in the semester, it is still premature to conclude that the student has already fully mastered the topic. Besides, there are still other opportunities where the student will be assessed in the future (e.g., *E2* and *E3*).

5.3.4.5 Overview of the Students

The dataset encompasses students taking an Introductory Computer Programming course and is mainly composed of first-year students. A total of 48,469 transactions were collected pertaining to the KC-level performance of the students. These were derived from 371 students over the course of three years. On average, a student was evaluated using 130.64 KCs with varying weights throughout the semester. Similar to the earlier chapters, the actual final grades of the students were inaccessible. Thus, student outcome was simply estimated using a proxy measure, which is the average performance of the student in the three exams denoted by λ. Figure 15 provides an overview of the distribution of the λ of the students grouped according to year. The line indicates the median while the green dot indicates the average. A Kruskal–Wallis test was performed to check the consistency of the average overall performance of students belonging to different semesters. No significant difference was found ($p = 0.49$) suggesting that there was consistency between the average performances of the students over the years. In succeeding sections, students were labeled as either high-performing or low-performing based on a certain cutoff point. The cutoff point used was the average λ of the 371 students ($M = 0.78$, $SD = 0.11$).

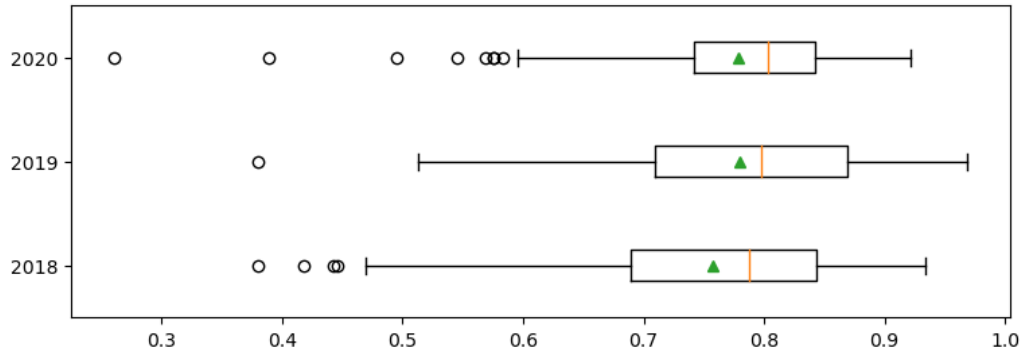


Figure 15. Distribution of Overall Performance

5.4 Personalized Recommender of Items to Master and Evaluate (PRIME) Framework

5.4.1 Problem Formulation

It has been shown in Chapter 4 that validating one's evaluative judgment by soliciting feedback while evaluating erroneous examples was associated with improved performance on a forthcoming test. Nevertheless, the study had one limitation in that students were given the same set of examples to evaluate (i.e., hypothetical student answers with mistakes). The current mastery levels of the students were not taken into account. This might partially explain why the completion of the learning activity had no significant impact on their test performance. To enhance the learning activity further, students should be provided with *appropriate* examples to evaluate. This means that the difficulty level of the activity is just the right amount (i.e., Goldilocks principle) or within what Vygotsky (1978) termed as zone of proximal development.

This work examines a coarse-grained knowledge unit referred to as topics, which is composed of multiple concepts (i.e., KCs). It is important to explain how the term is operationalized within this research context. As the aim of the exercise is to make students aware of their misconceptions, an example is considered appropriate

if it addresses topics in which they have deficiencies. Additionally, to ensure that the difficulty level is appropriate, the example must have a score which is similar to what students are predicted to obtain for a similar example. By using the learning activity, students are given the illusion that they are evaluating another person's answer, however the system is assuming that the same student would have the same answer if they were given the same question. As a result, students implicitly evaluate their own work, which fosters what Sharp (2012) refers to as stealth learning. Through this activity, students are made aware of somebody else's performance that resembles their own performance, thus helping them become more aware of their own deficiencies. Furthermore, it uses a knowledge gap-based remedial recommendation system that makes recommendations intended to close these gaps (Bauman & Tuzhilin, 2018; Thaker et al., 2020).

Given the study context, it is then important to identify the different components of the system to achieve the goal. The goal is to find an appropriate example to evaluate given a student's forecasted performance on an upcoming test. This goal can be subdivided into two subgoals, namely: (1) forecast the student's future performance on a test and (2) find an answer to recommend to the student. These two subgoals will then be discussed in detail in the following subsections. An overview of the steps is illustrated in Figure 16.

5.4.2 Forecasting of the Student's Future Performance

There are numerous approaches to predicting future performance in the literature. One common approach involves knowledge tracing where a student model estimates the mastery of a given knowledge component. However, as previously mentioned, the

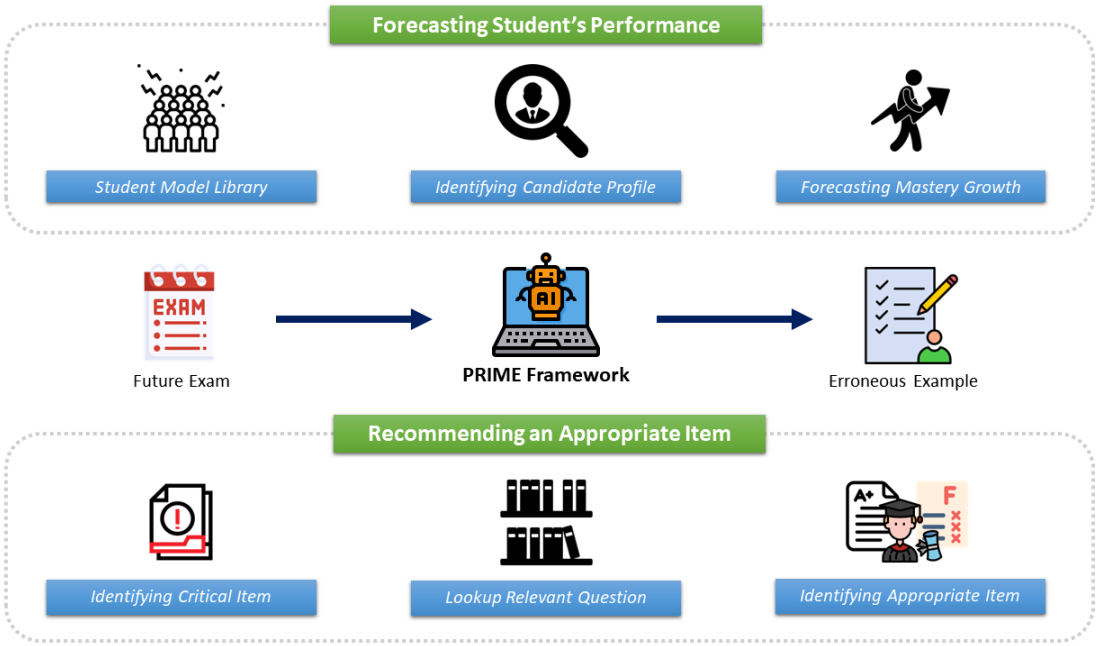


Figure 16. Overview of the PRIME Framework

context of the study involves complex questions that mostly involve partial credits. One way to solve this problem involves looking at the following analogy. Over the years, the teacher sees a lot of students with varying backgrounds and performances taking the same class. It is more likely that in the future, a new student will exhibit a similar behavior that has already occurred in the past. Based on this, it is then possible to assume that this new student might perform somewhat similarly to this other student from the past. Therefore, in this work, to be able to forecast a student's future performance, the system relies on a library of student models and identifies a candidate student model.

5.4.2.1 Populating the Student Library

In developing the student library, a model was developed for each student. This section provides an overview of the principled approach to modeling student mastery.

5.4.2.1.1 Rescaling and Defining Uniform Time Points

Despite having a similar number of examinations in a semester over the years, the time when these were administered varied by a few days. If these differences were unaccounted for, it would be difficult to develop certain models and incorporate additional assessments in future analyses. Therefore, these dates had to be transformed and rescaled to ease the process. Basically, the entire semester is viewed as a timeline and all dates are transformed into a value from 0 to 1 to indicate the relative time point progression with respect to the total number of days in the semester. For example, if a semester has a total of 110 days and an exam was administered on the 40th day, this translates to a time point $\pi = 0.36$. Throughout this work, a time point will be denoted by π.

5.4.2.1.2 Extracting Student Knowledge Mastery Levels

Certain assumptions were made in estimating students' mastery levels. In this work, the knowledge domain is assumed to be represented by a set of questions from all the examinations that were given to the students. Such an approach was motivated by knowledge space theory where a set of domain problems or items constitute the knowledge domain (Doignon & Falmagne, 1999; Falmagne et al., 1990). Mastery is

determined based on the student's response on the question (Doignon & Falmagne, 1999). Additionally, as typical in other studies, course content was believed to reprent domain knowledge and KCs to be learned (Khan & Ghosh, 2021). Because of the cumulative nature of the computer programming domain, it is necessary to identify an approach that captures this. In other words, this value should reflect the accumulated knowledge of a student over time from varying pieces of evidence (i.e., performance data). To achieve this, it is necessary to identify an upper bound for a given time point π to allow for the value to be normalized and comparable with other values. Using historical data, such an upper bound can be identified. Consider the curated dataset described earlier in Section 5.3. It can be seen that despite having been collected at different years, there was consistency in terms of the average distribution of topics and points among the tests created as illustrated in Section 5.3.3.

With the proposed approach, a student's mastery level for a topic at a given time point π, denoted by $\delta(\pi)$, is the ratio of the cumulative sum of all the points obtained up until π that are associated with the topic to the total possible points that could be earned associated with the topic (5.2a). Such information is readily available for offline analysis. However, for online analysis, this can only be estimated as long as both the mastery level and the cumulative sum of points at time point π are known (5.2b). Using historical data, it is possible to estimate the mastery level for any given time point.

$$\delta(\pi) = \frac{\text{points}_{\pi}}{\text{total points}} \tag{5.2a}$$

$$\hat{\text{total points}} = \frac{\text{points}_{\pi}}{\delta(\pi)} \tag{5.2b}$$

To illustrate this, assume that at an arbitrary time point π, it is expected that based on the past, the mastery level for a given topic is $\delta = 0.25$ and that by this point, the maximum points that could be earned is 30. Therefore, the total points,

or the maximum possible points that could be earned by the end of the semester, can be estimated. In this case, the total is estimated to be 120 points. Note that this estimation is subject to change and must be recalculated as new data becomes available, particularly the cumulative points.

$$\begin{aligned} \hat{\text{total points}} &= \frac{\text{points}_{\pi}}{\delta(\pi)} \\ &= \frac{30}{0.25} \\ &= 120 \end{aligned}$$

Using this estimated value, the student's mastery level at time point π can now be calculated. For example, if the student has only accumulated 15 points, this would translate to a mastery level of 0.13 at time point π.

$$\begin{aligned} \hat{\delta}(\pi) &= \frac{\text{points}_{\pi}}{\hat{\text{total points}}} \\ &= \frac{15}{120} \\ &= 0.13 \end{aligned}$$

With this approach, the mastery level has an upper bound of that of the ideal case (i.e., the student always gets a perfect score) which in this case is 0.25. Unless $\pi = 1$, the mastery level only provides an incomplete snapshot of the student, particularly only up until π. As $\pi \to 1$, it follows that $\delta(\pi) \to 1$ along with the confidence on the value obtained. A similar situation was observed in an earlier study, in which the accuracy of predictions increased with time as more information was accumulated (Paredes et al., 2018). Employing this approach makes mastery level monotonically increasing. A similar approach was employed by Sosnovsky and Brusilovsky (2015) to model a student's current level of knowledge on a topic referred to as "average of sums of averages". As opposed to their study, this study allows students to make only

one attempt at answering test items. Interestingly, modifying their approach yields a similar result to that discussed previously.

5.4.2.1.3 Modeling Topic Mastery Growth

Quantifying domain knowledge mastery is a challenging task (Lorenzetti et al., 2016, as cited in Khan & Ghosh, 2021). In this operationalization of knowledge mastery levels, the values range from 0 to 1. Unlike approaches from other studies that represent mastery as a probability of success, this work represents mastery as a real value reflecting the accumulation of various pieces of evidence throughout the semester. It is worth noting that this approach assumes that students do not have any prior knowledge at the beginning of the semester as indicated by the $\delta(0) = 0$ mastery level illustrated in Figure 17. Considering that the course investigated was an introductory course typically offered to first-year students, such an assumption is supposed to be adequate. Throughout the section, the discussion refers to a single topic. The various mastery levels of the students were estimated using (5.2a). For an offline analysis, it is unnecessary to estimate the total points as done in (5.2b) because this value is known. Therefore, all scores obtained by the student throughout the semester were transformed into their corresponding mastery levels, with respect to the known total points, during the time point when they were administered. An example is illustrated in Figure 17, top.

The next step involves modeling the growth of the mastery level over time. Regarding these data points as snapshots of a student's state enables the development of a growth model. An approach capable of modeling this is the logistic growth curve (S-curve) or also referred to as Verhulst model (Verhulst, 1845) illustrated in

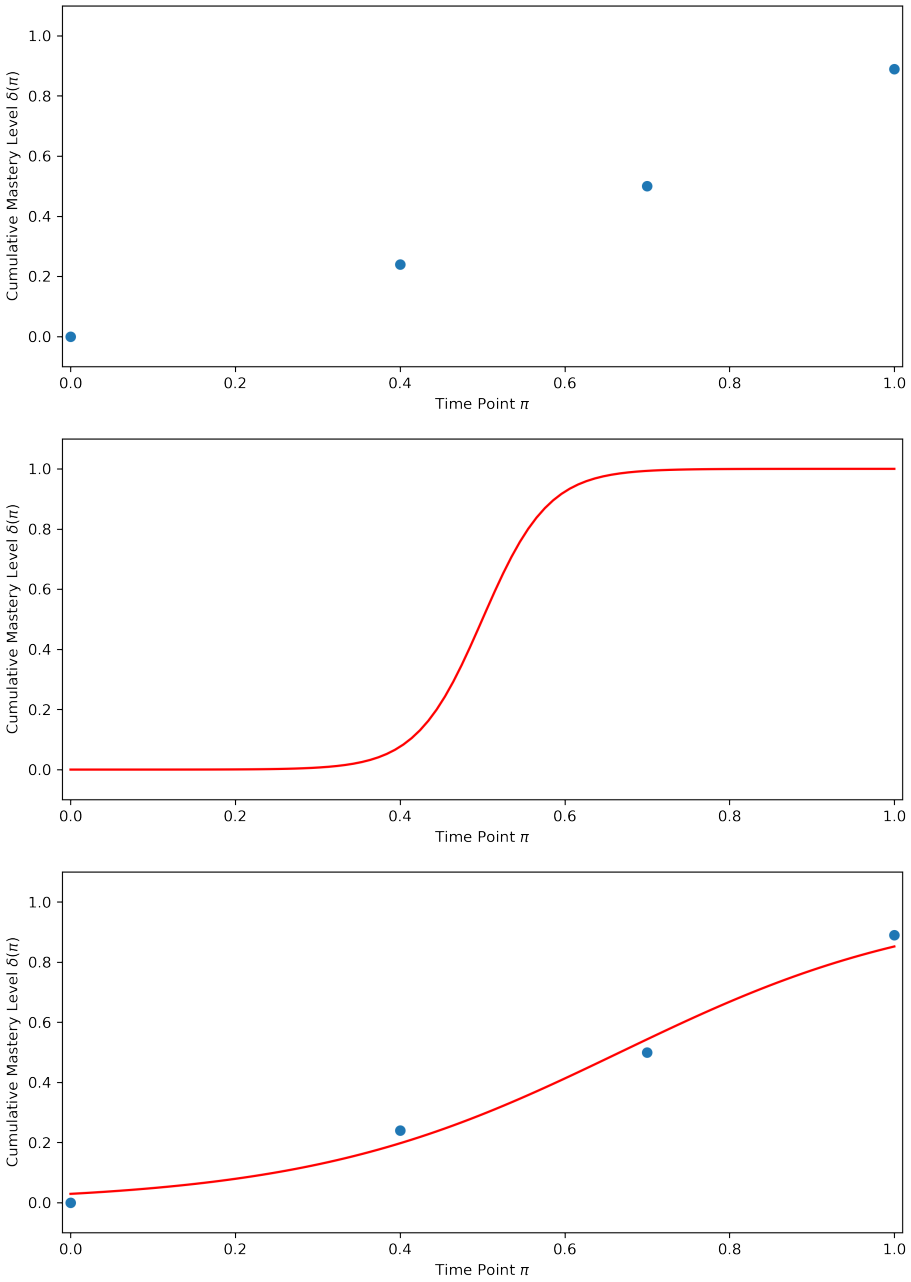


Figure 17. Fitting a Growth Curve Using the Mastery Level Over Time

Figure 17, middle. The general form is:

$$f\,(x) = \frac{k}{1 + \exp\,(\,-r\,[\,x - m\,]\,)} \tag{5.3}$$

where k is the carrying capacity (i.e., maximum sustainable population), x is a time point, r is the growth rate representing the curve's steepness, m is the time point at which the curve is at the midpoint. $f\,(x)$ is the population size at time x. This model is known for its S shape and is commonly used in the fields of Biology and Economics. In this work, y is treated as the mastery level. Furthermore, since its value is bounded ranging from 0 to 1, it can be assumed that $k = 1$. As a result, this Sigmoid function only needs to estimate two parameters from the data, namely the growth rate (r) and at what time point the mastery level reached 0.50 (m). Thus, It assumes that an individual doing a new task begins slowly. Gradually, the individual becomes proficient and ultimately, a plateau is reached. It is a progression of discovery geared towards the limit.

As an illustration, consider the previous example shown in Figure 17, bottom. By performing curve fitting (more details later), the optimal parameters for the data points can be identified. In this case, the values for $r = 5.26$ and $m = 0.67$ were obtained suggesting that the student's growth rate is 5.26. Additionally, the student was able to attain a level of mastery of 0.50 at $\pi = 0.67$.

5.4.2.1.4 Development of Unified Student Model

The preceding section discussed how mastery of a single topic can be modeled. As a result, each student has a total of t independent sigmoid models, one for every topic. Considering how all these models are associated with the same student, introducing a common parameter will unify the t independent models and will account for the fixed

effect of the student. As a result, a unified student model (which will be referred to as β function) was developed based on (5.3). It integrates the independent models and introduces a student latent trait θ. The function is:

$$\beta\left(\mathbf{X}\right) = \sum_{i=1}^{t} \frac{\mathbf{X}_i}{1 + \exp\left(-\theta\,\mathbf{R}_i\left[\mathbf{X}_{t+1} - \mathbf{M}_i\right]\right)} \tag{5.4}$$

where $\mathbf{X}$ is a consolidation of a one hot encoding of the topic (first t elements) and the time point (last element). $\mathbf{R}_{1\times t}$ and $\mathbf{M}_{1\times t}$ correspond to the growth rate and the time point at which the mastery level reached 0.50, respectively, for each of the t topics. Finally, θ is a scalar value representing the interaction between the student's latent trait and the student's growth rate for a particular topic.

$$\mathbf{X}_{1\times(t+1)} = \begin{pmatrix} x_1 & x_2 & \cdots & x_t & \pi \end{pmatrix}$$

5.4.2.1.5 Estimating Individual Growth Model Parameters

The β function allows for the estimation of the parameters of the students who took the course. The collection of parameters serve as the core of the student model library that can be used to make predictions to a new cohort of students. Similarly, this library represents the various types of students (i.e., profiles) that have been encountered in the past and may be encountered again in the future.

A total of $2t+1$ parameters need to be estimated: one common parameter associated with the student and two parameters associated with each topic (growth rate and time point). These parameters can be estimated by performing curve fitting. In this work, the `curve_fit` function from the SciPy[3] library was used for the curve fitting. In this process, (5.4) is referred to as the basis function. The optimal parameters are

[3]https://scipy.org/

determined through nonlinear least squares estimation. The `curve_fit` function will be passed the following input: $\mathbf{X}'$ and $\mathbf{Y}$. The former is a vector of $\mathbf{X}$ while the latter is a vector of the actual mastery levels. Both have a length of n data points. The error function, denoted by $e(\cdot)$, is computed based on the root-mean-square deviation shown in (5.5)

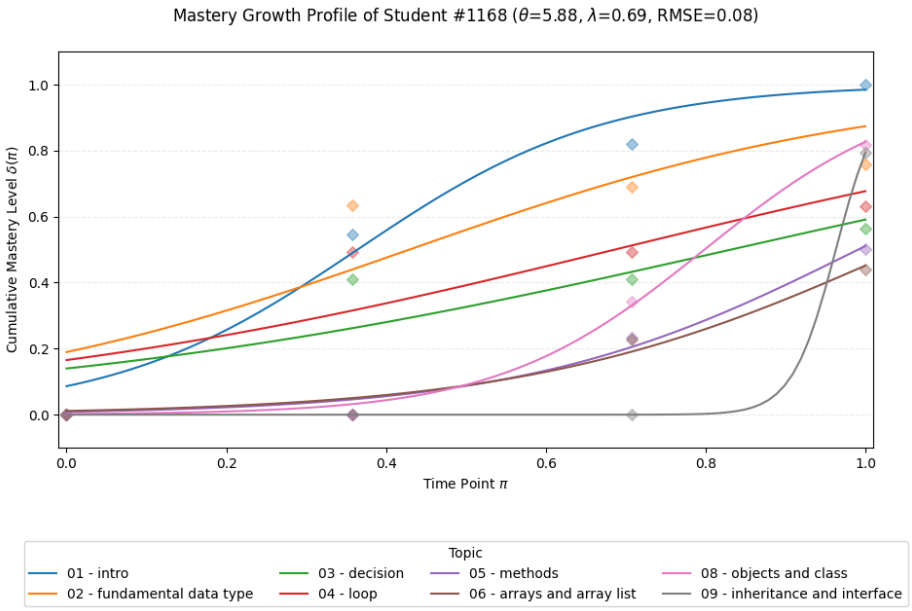
$$e(\beta; \mathbf{X}', \mathbf{Y}) = \sqrt{\frac{\sum_{i=1}^{n} \left(\mathbf{Y}_i - \hat{\beta}(\mathbf{X}'_i) \right)^2}{n}} \tag{5.5}$$

5.4.2.1.5.1 Example Student

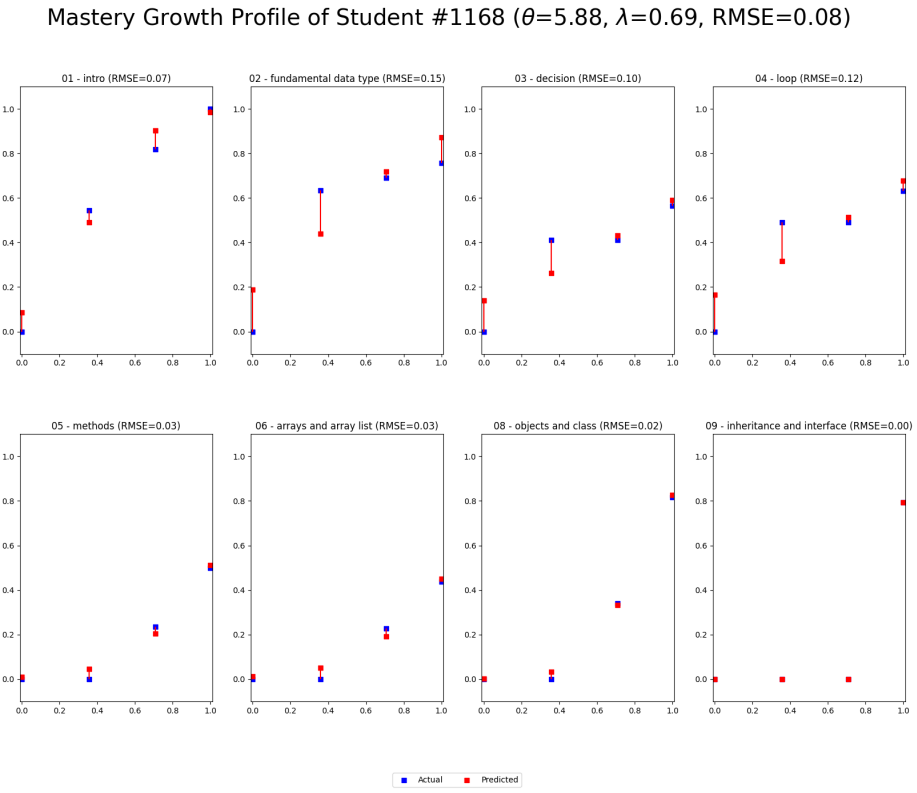
As a demonstration, consider a student with a profile of $\lambda = 0.69$ along with their performance data captured by the system. The lines in Figure 18 represent the estimated mastery level of the student for each topic over time. On the other hand, the points represent the actual mastery level. It is also possible to visualize the errors according to topics. The latent trait of this student was $\theta = 5.88$.

5.4.2.1.5.2 Ideal Student

It is also noteworthy to see what the growth curve of an ideal student looks like. This ideal student, which will be referred to as ω, simply is a hypothetical student who obtains a perfect score in all learning opportunities ($\lambda = 1$). As can be seen in Figure 19, the curves tend to uncover which topics were mostly the focus during a particular time point as some curves are closer to each other. The latent trait of this student was $\theta = 7.81$.

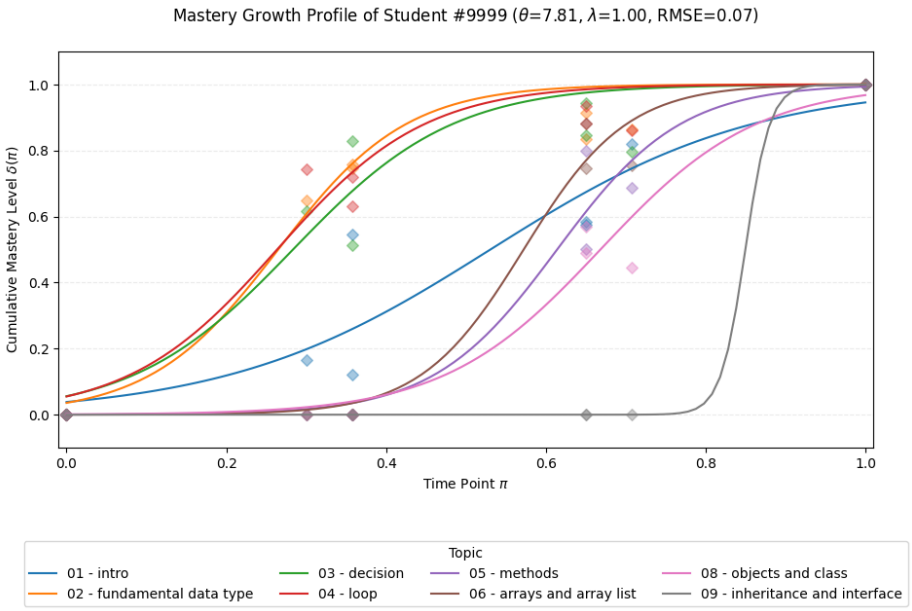


(a)

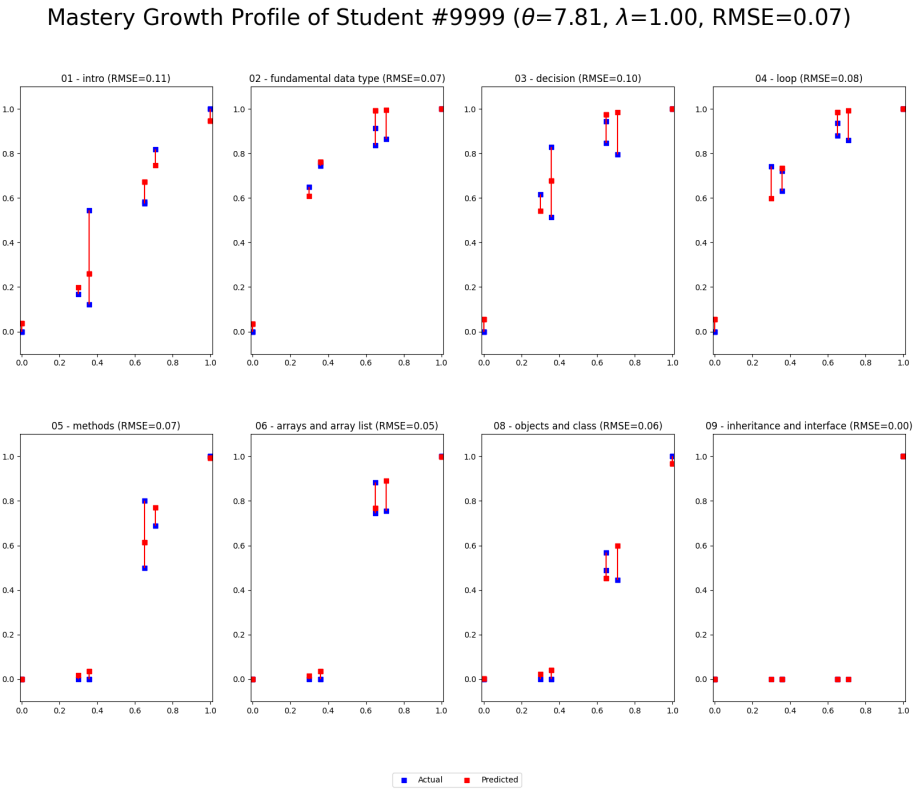


(b)

Figure 18. Profile Overview of an Example Student



(a)



(b)

Figure 19. Profile Overview of an Ideal Student ω

5.4.2.1.5.3 Interpreting the Latent Trait

It is assumed that θ is the student's general learning rate. However, because this value was only estimated, a possible interpretation was sought. Hence, students were classified either as high-performing or low-performing based on their overall performance λ. The average of all the λ values was used as the cut-off value ($M = 0.77, SD = 0.11$). Interestingly, the result of a Mann–Whitney U test suggests that on average, the θ of the high-performing students ($M = 6.38, SD = 4.74$) was significantly ($p < 0.01$) higher than the low-performing students ($M = 5.18, SD = 1.96$).

5.4.2.2 Identifying a Candidate Student Model

The previous section has demonstrated that the student model is capable of forecasting the mastery level of a student once the parameters are known. However, these parameters can only be identified once all the performance data had been collected which is at the end of the semester. If this framework is to be deployed to be used during the class, it will struggle to make predictions due to the incompleteness of the data. To address this, identifying a candidate model can be framed as a recommender problem where a neighborhood-based approach is employed. Essentially, the models of previous students can be leveraged and compared for similarity to account for these missing parameters. With the student model library $\mathbf{L}$ already developed, forecasting the outcomes of future or next-term students (denoted by γ) may be performed as new data arrives. This scenario is considered an instance of online analysis as a complete picture of γ only becomes available at the end of the semester. This section enumerates the various steps.

5.4.2.2.1 Step 1: Determining Key Time Points

As forecasting student outcomes rely on existing evidence, it is necessary to define two key time points to accomplish the task, namely π_e and π_q. The former is the latest time point at which evidence of the student's performance exists while the latter is the time point at which the system is being queried to do the forecasting. Usually, π_q is the time when the next test will be administered. It is assumed that $0 \leq \pi_e < \pi_q \leq 1$. Otherwise if $\pi_q \leq \pi_e$, it is simply reduced to a simple lookup. Figure 20 illustrates an example where the system already has evidence of student performance (yellow region) and attempts to forecast the student's performance (blue region). The yellow line denotes π_e while the blue line denotes the π_q.

5.4.2.2.2 Step 2: Fitting Evidence to an Existing Student Model

The next step involves searching and retrieving a model β_s from the library $\mathbf{L}$ that best fits the available data points of a new student γ. Finding a student from the past that closely resembles this new student can help inform how γ would likely perform in the future. This can be accomplished by minimizing the the error between the model's output and the actual data (i.e., points in the yellow region) based on (5.5). An example is illustrated in Figure 21, top.

$$\min_{\forall s \in \mathbf{L}} \; e\left(\beta_s \,; \mathbf{X'}_\gamma, \mathbf{Y}_\gamma\right) \tag{5.6}$$

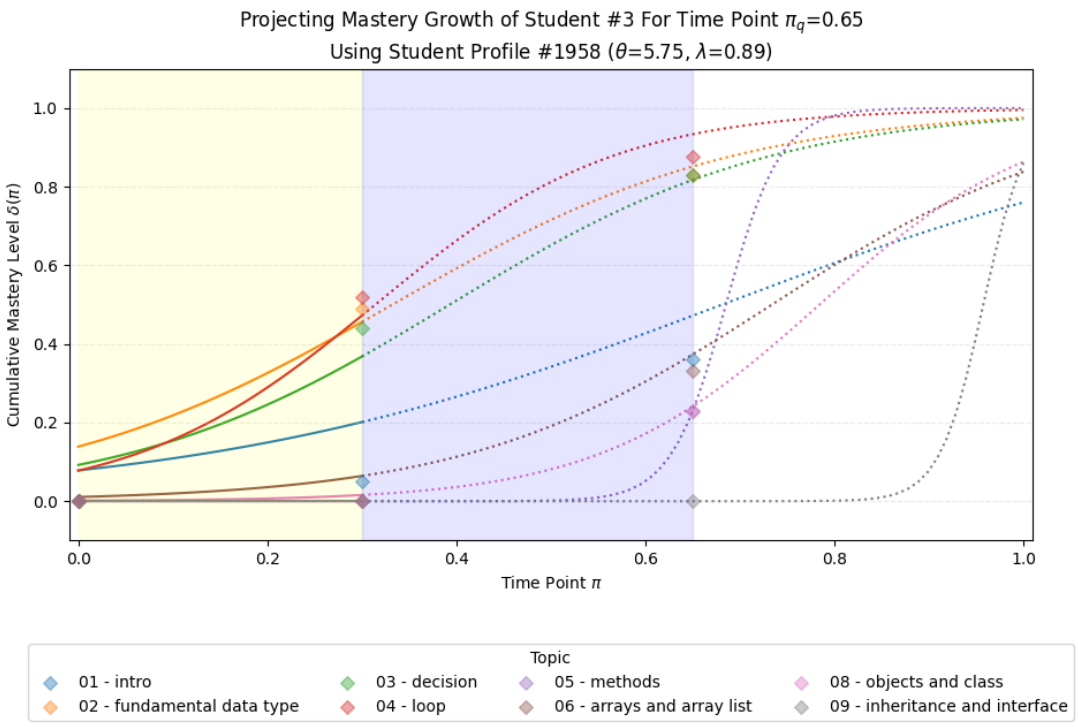


Figure 20. Time Point Regions

5.4.2.2.3 Step 3: Finding a Similar Student Profile

Besides fitting the existing data points of a new student, another approach to measuring similarity can be based on the mastery level of the student at a given time point $\delta(\pi)$ for each topic denoted by $\mathbf{S}_\pi$.

$$\mathbf{S}_{\pi\ 1\times t} = \begin{pmatrix} \delta_1(\pi) & \delta_2(\pi) & \cdots & \delta_t(\pi) \end{pmatrix}$$

By relying on recent information, this would translate to identifying a student s from the library $\mathbf{L}$ that minimizes the distance between the profile of s and γ at time point π_e of γ (5.7).

$$\min_{\forall s \in \mathbf{L}} d(s, \gamma, \pi_e) \tag{5.7}$$

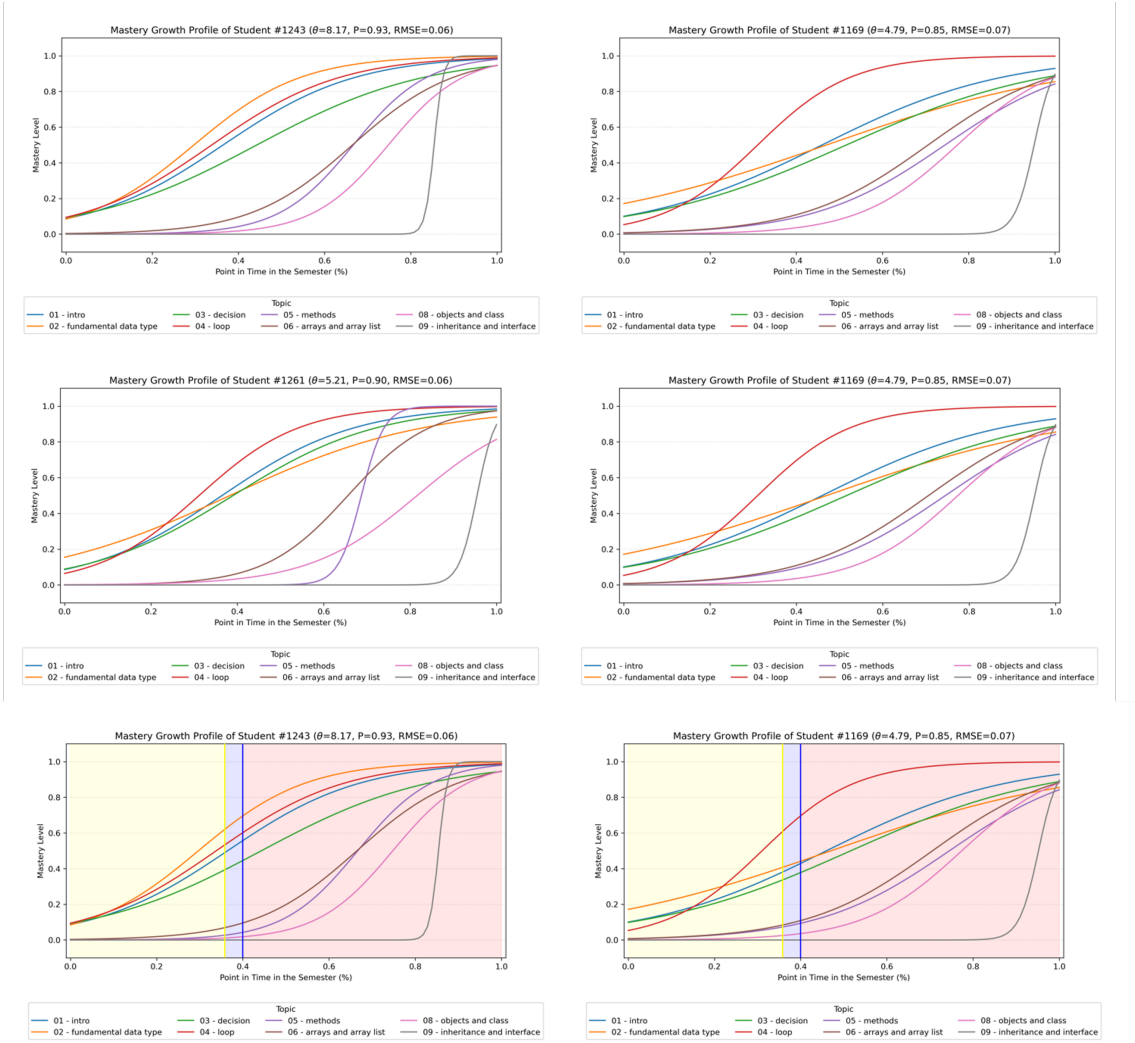


Figure 21. Comparison of Sample Output of the Three Student Similarity Metrics

Euclidean distance is used to measure the similarity (5.8). A value of 0 indicates an absolute identity. An example is illustrated in Figure 21, middle.

$$d\left(s,\gamma,\pi_e\right)=\sqrt{\sum_{i=1}^{t}\left(\mathbf{S}_{\pi_e s_i}-\mathbf{S}_{\pi_e \gamma_i}\right)^2} \tag{5.8}$$

5.4.2.2.4 Step 4: Combining and Weighing the Two Metrics

Insofar as identifying similar students from the library is concerned, the two previously discussed metrics can be seen as independent from each other as each focuses on a different aspect of the student. Simply relying on the fitted student model imposes a strong assumption that the current student will exhibit similar behavior in the future despite the lack of strong evidence (i.e., no data points beyond π_e). On the other hand, relying only on profile similarity disregards the overall aspect of the student model as it focuses solely on a particular time point (π_e). Therefore, it is important to integrate both information while accounting for such uncertainty. This could be achieved by assigning weights to the metrics based on the available evidence. Intuitively, the fitted student model is given importance up until π_e. For the remaining period $(1 - \pi_e)$, the system gives more importance to the similarity of the student profiles to account for the uncertainty (5.9).

$$\min_{\forall s \in \mathbf{L}} \left[(\pi_e \cdot e\left(\beta_s\,; \mathbf{X'}_{\gamma}, \mathbf{Y}_{\gamma}\right)) + ((1 - \pi_e) \cdot d\left(s, \gamma, \pi_e\right)) \right] \tag{5.9}$$

An example is illustrated in Figure 21, bottom. The yellow region highlights the area in which evidence exists and therefore is given importance. The blue region highlights the area in which the system is being queried to predict the likely performance of the student. Typically, it is based on when the new test will be administered in class. Finally, the red region highlights the area in which the system does not have any information about the student and therefore has a very high degree of uncertainty.

In summary, given the current mastery level of a student γ, identifying a candidate profile from the library can be framed as a recommender problem. The system implicitly recommends a profile that is likely to have a very similar trajectory to that of the student γ. Formally, to forecast the performance of a new student γ at time

point π_q, a similar student s from the library $\mathbf{L}$ is identified. Afterward, this model β_s will be used under the assumption that both students will have similar outcomes based on a similar normalized gain. A popular metric for assessing the quality of recommendations is to determine whether the item being recommended is relevant to the user. However, in this case, defining what is relevant for the student is not and cannot be explicitly defined. Therefore, the nature of the profiles chosen by the system needs to be looked into.

5.4.3 Identifying an Appropriate Example to Recommend

Given that the system is now capable of making projections of a student's mastery level at any given time point, the next part of the framework involves identifying and recommending an appropriate student answer to a given question. This entails identifying the types of questions that form part of an upcoming test. Recall that regardless of when the course was offered, the same syllabus was followed thereby making the content and the timing effectively identical. In addition, there is a tendency for teachers to reuse questions from prior years. Therefore, if the system is provided with information of an upcoming test, it should be possible to find a similar question from the past. The rationale behind this is that the system cannot explicitly tell the students what exact questions will come out in the upcoming test. However, the system can tell the students that a certain proxy question is believed to be similar enough and has already been administered in the past. Therefore, in this work, to be able to identify an appropriate answer to recommend, the system first uses the forecasted performance to determine the *critical item* or the question in the upcoming test where the student is likely to perform the poorest. Then, it uses the information of

the critical item to identify a proxy from the past. Finally, it looks for student answers to the proxy question where the associated score closely resembles the projected score of the current student γ.

5.4.3.1 Populating the Question Library

To be able to identify a proxy question, it is necessary to develop a library which contains all the questions written for the course. Whenever teachers create a new test, they supply the system information pertaining to the various KCs and the point assignment for each item. Using this information, the topic distribution for a question can be derived. One way to represent a test is through a $\mathbf{Q}$ matrix. This matrix contains the distribution of topics required to answer the question correctly as illustrated in (5.10). Each value represents the strength of the influence of a topic to getting the question correctly. Somewhat, this can be closely associated to the test blueprint or table of specifications (Mehrens & Lehmann, 1991). The $\mathbf{Q}$ matrix developed by other researchers often contains binary values (Barnes, 2011; de la Torre, 2011; Tatsuoka, 1983). Similar to Brewer (1996), in this work, the values are continuous from 0 to 1, where a non-zero value represents the degree of importance of mastery of the topic as tests in introductory programming courses typically contain questions that simultaneously assess multiple topics (Sheard et al., 2013). Essentially, the value corresponds to the weight of importance of a topic (Guo et al., 2014). The row represents the various questions, while the column represents the various topics (or knowledge components) covered in the course. Each row sums to one. Similarly, another matrix can be used $\mathbf{Q}'$ which is identical to $\mathbf{Q}$ except that the row values were not normalized. Thus, instead of a distribution of topics, $\mathbf{Q}'$ contains the raw

points associated to each topic for each question. Dividing each row with its row-wise sum transforms it to $\mathbf{Q}$. Finally, because teachers provide the information about the tests, both $\mathbf{Q}$ and $\mathbf{Q}'$ are considered to be is defined by subject-matter experts.

$$\mathbf{Q}_{q\times t} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,t} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{q,1} & \alpha_{q,2} & \cdots & \alpha_{q,t} \end{pmatrix} \tag{5.10}$$

5.4.3.1.1 Question Representation

Now, focusing on one particular question, this step entails determining how to quantify the similarity between two questions. In constructing a question, teachers provide the weights (or importance) associated with the various knowledge components (KCs). According to Koedinger et al. (2012), every question or problem can be represented by a set of domain KCs or coarse-grained topics. In representing a question, all KCs were grouped together according to topics and the weights were aggregated since KCs may belong to the same topic (refer to the earlier example in Figure 14). Afterward, these aggregated weights were normalized to make it possible to compare multiple questions based on $\mathbf{I}$. The values of the first t elements range from 0 to 1, and sum to 1, effectively representing the topic distribution of a question. Some questions may be more complex than others. This information is discarded in the normalization process. To preserve and incorporate this, a metric is introduced to represent a question's complexity, called point per KC. It simply is the average point or weight associated with a KC, denoted by ϕ, regardless of the topic.

One issue that may arise when computing for the similarity given the current

question representation would be the scale of ϕ as it currently lacks an upper bound. To address this, the value is rescaled through min-max normalization. sklearn's[4] `MinMaxScaler` was applied to bound the values from 0 to 1. This rescaled value $\hat{\phi}$ becomes the last element of $\mathbf{I}$. It is worth noting that $\hat{\phi}$ is dynamic as these have to be recomputed whenever questions are added due to the dependence on the minimum and maximum values. However, based on the dataset, if the ϕ of the new questions are within the range of 1.00 to 6.00, then, no recalculation is necessary. Essentially, $\mathbf{I}$ is the row of the question from $\mathbf{Q}$ plus $\hat{\phi}$.

$$\mathbf{I}_{1\times(t+1)} = \begin{pmatrix} i_1 & i_2 & \cdots & i_t & \hat{\phi} \end{pmatrix}$$

5.4.3.2 Identifying the Critical Item

The next step is identifying the *critical item*, an item from the upcoming test where the student is believed to perform the worst given their projected mastery levels. Afterward, a proxy question based on the critical item will be identified from the question library. As a reminder, this is an online process where certain information only becomes available at certain time points π.

The next step is identifying the *critical item*, an item from the upcoming test where the student is believed to perform the worst given their projected mastery levels. Afterward, a proxy question based on the critical item will be identified from the question library. As a reminder, this is an online process where certain information only becomes available at certain time points π.

[4]https://scikit-learn.org/

5.4.3.2.1 Step 1: Modifying the Performance Forecasting Formula

Recall that the mastery level for a given time point $\delta(\pi)$ of the student can be estimated using (5.2a). Because the goal is to identify the performance of the student in an upcoming test at time point π_q, (5.2a) is modified to distinguish the cumulative points already earned up until π_e from the score projected to be earned at π_q. The modified formula is given in (5.11b).

$$\delta(\pi_q) = \frac{\text{points}_{\pi_q}}{\text{total points}} \tag{5.11a}$$

$$= \frac{\text{points}_{\pi_e} + \text{score}_{\pi_q}}{\text{total points}} \tag{5.11b}$$

As both (5.2a) and (5.11b) pertain only to a single topic, the following convention is employed for clarity and to account for any given topic τ:

$$\text{CM}(s, \tau, \pi_q) = \frac{\text{CP}(s, \tau, \pi_e) + \text{PE}(\mathbf{Q}', s, \tau, \pi_q)}{\text{TP}(\tau)} \tag{5.12}$$

This convention simply generalizes the formula to any given student s for topic τ using the following functions: $\text{CM}(\cdot)$ for cumulative mastery, $\text{CP}(\cdot)$ cumulative points, $\text{PE}(\cdot)$ for points earned in a test given its $\mathbf{Q}'$ matrix (raw points), and $\text{TP}(\cdot)$ for total points.

5.4.3.2.2 Step 2: Estimating the Course Total

Due to the online nature of the process, the final total is unknown unless $\pi = 1$. Thus, the system relies on historical data as done in (5.2b) based on the ideal student $s = \omega$ to estimate the total points for the topic. So, (5.12) is rewritten to solve for $\text{TP}(\cdot)$. Additionally, in the case of the ideal student ω, the $\text{PE}(\cdot)$ is equal to:

$$\text{PE}(\mathbf{Q}', \omega, \tau, \pi_q) = \sum_{i=1}^{q} \mathbf{Q}'_{i,\tau} \tag{5.13}$$

As an illustration, consider two arbitrary values for π_e and π_q, such that $\pi_e < \pi_q$. Based on historical data of an ideal student ω, it was found that the mastery level is $\delta(\pi_q) = 0.30$. Moreover, it is known that a student can obtain a maximum possible points 20 by π_e. Based on the $\mathbf{Q}'$ matrix of the upcoming test at π_q, the maximum score possible is 10 points. With all these, it is possible to estimate the total points using:

$$\hat{\mathrm{TP}}(\tau) = \frac{\mathrm{CP}(\omega, \tau, \pi_e) + \mathrm{PE}(\mathbf{Q}', \omega, \tau, \pi_q)}{\hat{\mathrm{CM}}(\omega, \tau, \pi_q)} \tag{5.14a}$$

$$= \frac{\mathrm{CP}(\omega, \tau, \pi_e) + \sum_{i=1}^{q} \mathbf{Q}'_{i,\tau}}{\hat{\mathrm{CM}}(\omega, \tau, \pi_q)} \tag{5.14b}$$

$$= \frac{20 + 10}{0.30} \tag{5.14c}$$

$$= 100 \text{ points} \tag{5.14d}$$

5.4.3.2.3 Step 3: Identifying Raw Topic Performance

Presently, the total points can be estimated. Also, the mastery level of a student can be projected based on the candidate student model as identified in Section 5.4.2.2. Using the candidate student model, the normalized gain on the mastery level between π_e and π_q is calculated as defined by Hovland et al. (1949, as cited in Sosnovsky & Brusilovsky, 2015) and provided in (5.15).

$$\Delta\delta_\tau = \frac{\mathrm{CM}(s, \tau, \pi_q) - \mathrm{CM}(s, \tau, \pi_e)}{1 - \mathrm{CM}(s, \tau, \pi_e)} \tag{5.15}$$

This allows for the computation of the forecasted mastery level $\mathrm{CM}(\cdot)$ by adding the obtained normalized gain to γ using a rewritten version of the normalized gain

formula given in (5.16).

$$\mathrm{CM}\left(\gamma, \tau, \pi_q\right) = \Delta\delta_\tau - \Delta\delta_\tau \cdot \mathrm{CM}\left(\gamma, \tau, \pi_e\right) + \mathrm{CM}\left(\gamma, \tau, \pi_e\right) \tag{5.16}$$

Using these, the points earned in the upcoming test $\mathrm{PE}\left(\cdot\right)$ can now be forecasted. Note that for now, these points pertain to a topic and not a question. By rewriting (5.12) the following is obtained:

$$\mathrm{CM}\left(s, \tau, \pi_q\right) = \frac{\mathrm{CP}\left(s, \tau, \pi_e\right) + \mathrm{PE}\left(\mathbf{Q}', s, \tau, \pi_q\right)}{\mathrm{TP}\left(\tau\right)} \tag{5.17a}$$

$$\mathrm{TP}\left(\tau\right) \cdot \mathrm{CM}\left(s, \tau, \pi_q\right) = \mathrm{CP}\left(s, \tau, \pi_e\right) + \mathrm{PE}\left(\mathbf{Q}', s, \tau, \pi_q\right) \tag{5.17b}$$

$$\mathrm{PE}\left(\mathbf{Q}', s, \tau, \pi_q\right) = \mathrm{TP}\left(\tau\right) \cdot \mathrm{CM}\left(s, \tau, \pi_q\right) - \mathrm{CP}\left(s, \tau, \pi_e\right) \tag{5.17c}$$

Continuing the example provided, assume a student $s = \gamma$ earned 15 of the 20 possible points so far. Thus, $\delta\left(\pi_e\right) = 0.15$. This suggests that $\delta\left(\pi_e\right) \leq \delta\left(\pi_q\right) \leq 0.25$. These boundaries were identified based on the minimum and maximum score that the student could obtain in the upcoming test (0 and 10, respectively). Because this value is provided by the candidate student model of γ, it is possible for the value to fall outside of the range. In such event, the value is clipped to restrict it within the boundary. A similar approach was employed by Gong et al. (2010) to account for the overestimation of their models which resulted to negative learning rates. Similarly, Huang et al. (2020) performed a clipping on their student score prediction models to limit the values between 0 and 1 on values obtained from their knowledge proficiency tracking.

5.4.3.2.4 Step 4: Computing the Topic Performance Weights

Recall that the steps identified earlier will be performed for each topic τ. Once the forecasted topic scores are computed, these will be divided by their corresponding

maximum possible scores based on the test's $\mathbf{Q}'$ matrix (i.e., the score of the ideal student ω). This step results in the construction of the topic performance weights $\mathbf{W}$ which contains the weights associated with the performance of the students on each topic. This is similar to the simple weighted student overlay model where topic-weight pairs represent the mastery of the student of a topic (Brusilovsky, 2003).

$$\mathbf{W}_{1\times t} = \begin{pmatrix} w_1 & w_2 & \cdots & w_t \end{pmatrix} \tag{5.18a}$$

$$= \begin{pmatrix} \dfrac{\mathrm{PE}\left(\mathbf{Q}', s, \tau_1, \pi_q\right)}{\mathrm{PE}\left(\mathbf{Q}', \omega, \tau_1, \pi_q\right)} & \dfrac{\mathrm{PE}\left(\mathbf{Q}', s, \tau_2, \pi_q\right)}{\mathrm{PE}\left(\mathbf{Q}', \omega, \tau_2, \pi_q\right)} \cdots \dfrac{\mathrm{PE}\left(\mathbf{Q}', s, \tau_t, \pi_q\right)}{\mathrm{PE}\left(\mathbf{Q}', \omega, \tau_t, \pi_q\right)} \end{pmatrix} \tag{5.18b}$$

5.4.3.2.5 Step 5: Determining the Critical Item

The goal is to identify a question or item in the upcoming test where the student is supposed to perform poorly. However, the $\mathbf{W}$ contains value that pertains to the topic scores of the student and not questions. Recall that a question may be associated with multiple topics. Therefore, to calculate for the forecasted scores of the student to each question $\mathbf{P}$, the cross product of the $\mathbf{Q}$ matrix and the $\mathbf{W}$ is computed. To simplify the process, $\mathbf{Q}$ is used instead of $\mathbf{Q}'$ as the results are identical. A similar approach was employed by Huang et al. (2020) in which the predicted students scores were the inner product of the student's proficiency vector and the question's knowledge vector. Thus, in this framework, the student's predicted scores for each question is computed as $\mathbf{Q} \bullet \mathbf{W}^{\mathsf{T}}$. Each value in $\mathbf{W}$ range from 0 to 1 to indicate the normalized score to a

question.

$$\mathbf{P}_{q\times 1} = \begin{pmatrix} \alpha_{1,1} & \alpha_{1,2} & \cdots & \alpha_{1,t} \\ \alpha_{2,1} & \alpha_{2,2} & \cdots & \alpha_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{q,1} & \alpha_{q,2} & \cdots & \alpha_{q,t} \end{pmatrix} \bullet \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_t \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_q \end{pmatrix}$$

The question with the lowest normalized score is marked as the *critical item*. It is the question which the system believes the student will have the lowest performance or the most misconceptions. Therefore, it is beneficial for the student be aware of this considering it is an item the student will gain the most when resolved. However, since the test is yet to be administered in class, this item cannot be directly shown to the student. For this reason, a proxy question from the library will be identified based on its relevance and will be provided to the student.

5.4.3.3 Determining Question Relevance

Given the $\mathbf{I}$ of a new question, the goal is to find a similar item from the library. The measurement of item similarity is, however, a challenging task. For the vast majority of domains, there does not seem to be a standard approach (Pelánek, 2020). Model-based and feature similarity-based approaches are the two most common approaches found in EDM literature. In this work, the latter approach is employed since the former often relies on latent attributes derived from student performance data (e.g., matrix factorization). Moreover, since questions have already been associated with topics in this context, it is reasonable to use them. In fact, one approach that can be employed is to measure similarity based on associated concepts as done by Hosseini and Brusilovsky (2017). Detailed steps as well as considerations of PRIME are outlined

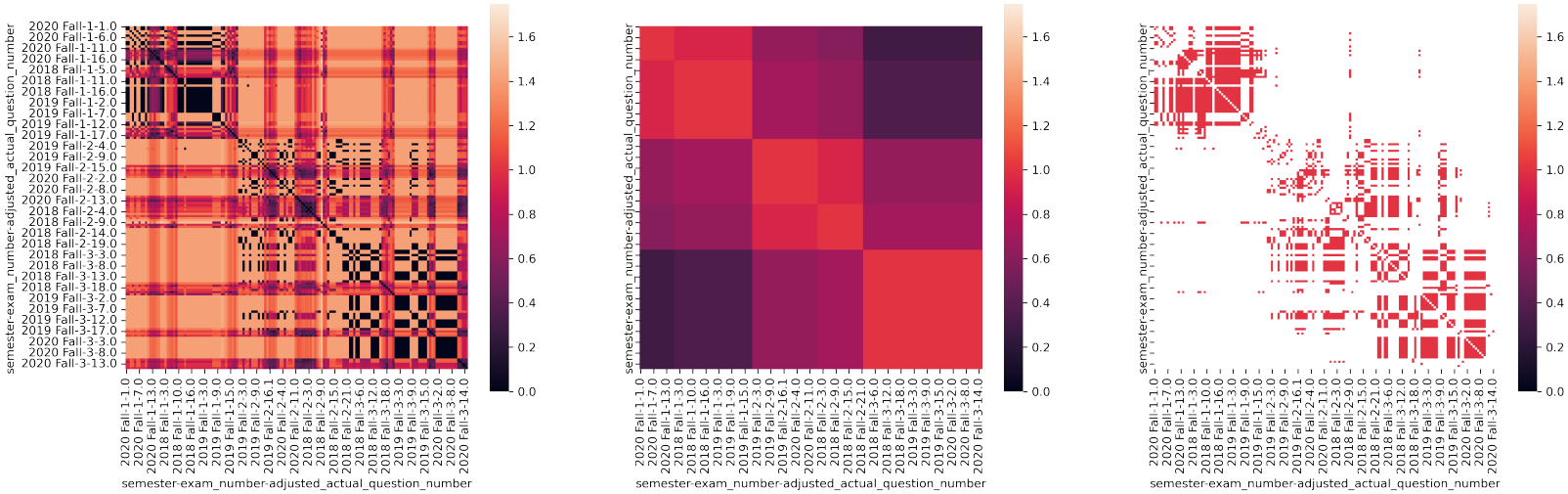


Figure 22. Heatmap Visualizing the Pairwise Application of the Three Metrics to Questions in the Library

Note: Questions are sorted according to their associated time point π, followed by the year.

in this section. For illustration purposes, the following discussion focuses on existing test questions in the library. However, it also applies to newly created questions.

5.4.3.3.1 Step 1: Computing the Euclidean Distance

A metric often employed in the data mining literature to measuring similarity between two vectors is the Euclidean distance. It is a common measure used for quantifying item similarity based on the item features (Pelánek, 2020). In accordance with the learning edge momentum hypothesis, items with closely related topics should be identified (Robins, 2010). Based on the dimension of $\mathbf{I}$, the distance ranges from 0 to $\sqrt{t+1}$ where a value of 0 indicates an absolute identity. Given two questions a and b, the distance can be computed using (5.19). The pairwise distance of all the questions in the library is visualized using a heatmap shown in Figure 22, left.

$$\mathrm{QD}\,(a,b) = \sqrt{\sum_{i=1}^{t+1} (b_i - a_i)^2} \tag{5.19}$$

5.4.3.3.2 Step 2: Computing Temporal Proximity

Given that the values in $\mathbf{I}$ are normalized along with the nature of the Euclidean distance, it is possible to find false similar questions from varying time points π if only the topic distribution were used. For example, a question administered from an earlier time point *might* be regarded as similar to a question from a later time point even if they are not. Additionally, because the schedule is defined in the syllabus, it was evident earlier in the growth curve of the ideal student ω that certain topics are often assessed together in a given time point (Figure 19, left). To address this, a metric that prioritizes questions that belong from a similar or nearer time point is

introduced. Intuitively, a question that is temporally nearer is given more importance over those that are further. Given two questions a and b, the temporal proximity can be computed using (5.20). The value ranges from 0 to 1 where a value of 0 indicates a low degree and a value of 1 indicates a high degree of temporal importance. The pairwise temporal proximity of all the questions in the library is visualized using a heatmap shown in Figure 22, center.

$$\mathrm{QT}\,(a, b) = 1 - |\, b_\pi - a_\pi \,| \tag{5.20}$$

5.4.3.3.3 Step 3: Computing Suitability

Recall that the objective is to identify proxy questions based on an upcoming test. These proxies will be provided to students to enable them to practice on the topics covered in the test. Thus, it is important to factor the topic coverage of the questions. For example, it is not beneficial for students to see questions on advanced topics that are not yet covered in class. This means that given the set of topics covered by two questions a and b, if b covers certain topics that are not present in a, b is deemed not suitable to be recommended. Suitability follows the idea of a superset in set theory and can be computed using (5.21).

$$\mathrm{QS}\,(a, b) = \begin{cases} 1, & \text{if } T(a) \supseteq T(b) \\ 0, & \text{otherwise} \end{cases} \tag{5.21}$$

where $T(\cdot)$ is the set of topics associated with the question. The value can either be 0 or 1. It should be noted that this value is not necessarily symmetric. The pairwise suitability of all the questions in the library is visualized using a heatmap shown in Figure 22, right.

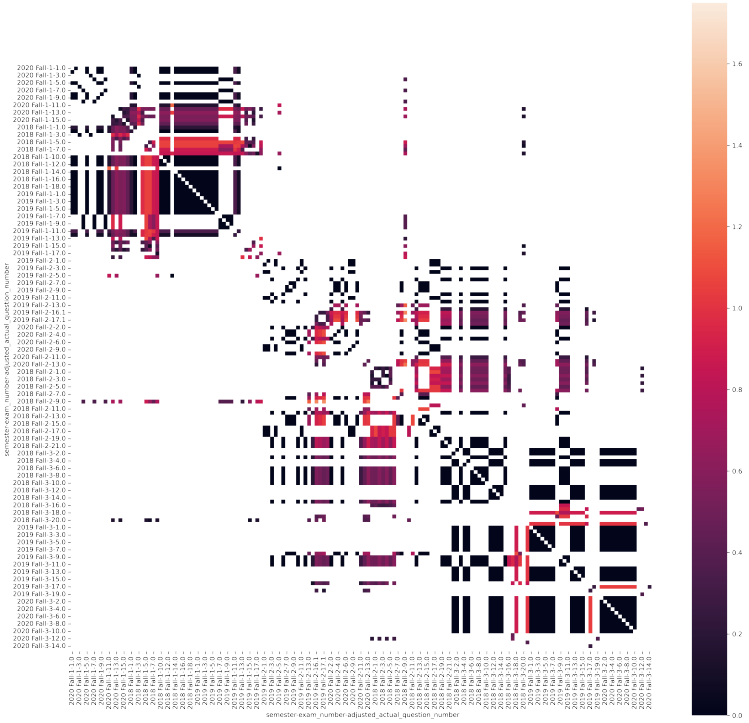


Figure 23. Heatmap Visualizing the Pairwise Relevance of Questions in the Library

Note: Questions are sorted according to their associated time point π, followed by the year.

5.4.3.3.4 Step 4: Computing Relevance

After considering the various aspects of a question, it is now possible to quantify relevance. Combining all the previous steps would yield the following (5.22).

$$\mathrm{QR}\,(a, b) = \mathrm{QD}\,(a, b) \cdot \mathrm{QT}\,(a, b) \cdot \mathrm{QS}\,(a, b) \tag{5.22}$$

Intuitively, it first computes for the Euclidean distance between a and b. Next, this distance value is adjusted based on their temporal proximity. Finally, the suitability acts as a masking function to determine whether b is ultimately a relevant question for a. As a consequence, the relevance value obtained is not necessarily symmetric. As this value is based on a distance measure, a value close to 0 is preferred. The pairwise

relevance of all the questions in the library is visualized using a heatmap shown in Figure 23. Figure 24 illustrates an example of the results obtained after using the metric on a single question.

5.4.3.4 Identifying an Appropriate Example to Recommend

Once a proxy question has been identified from the library, the final step entails identifying a single student answer from the past. Given that multiple students attempted the question, the objective is to find an appropriate one for the student γ. If the ultimate goal is to make apparent to students their misconceptions, it would be beneficial to identify one which closely resembles or imitates how they would answer. Also, how difficulty will be affected by the degree of error has to be taken into consideration (i.e., Goldilocks principle). Ideally, it should be within the student's zone of proximal development as to not discourage the student (Vygotsky, 1978). Students are expected to observe this example and learn from the consequences associated with it (Bandura, 1977). Therefore, the forecasted normalized score of student γ is used as a cut-off value to assign priority to existing answers from the database. Answers with scores lesser or equal to the cut-off value are given higher priority while the remaining are given low priority. Additionally, providing answers with higher scores gives rise to the possibility of simply giving out the *answer key* in which the student may fail to see their misconceptions or possible mistakes.

Priorities alone give rise to cases with multiple ties. Thus, for each priority group, the scores are arranged in descending order. The rationale here is to identify answers with scores that are as much as possible close to the γ's predicted score. The next step involves looking into the breakdown of the score based on topics. The

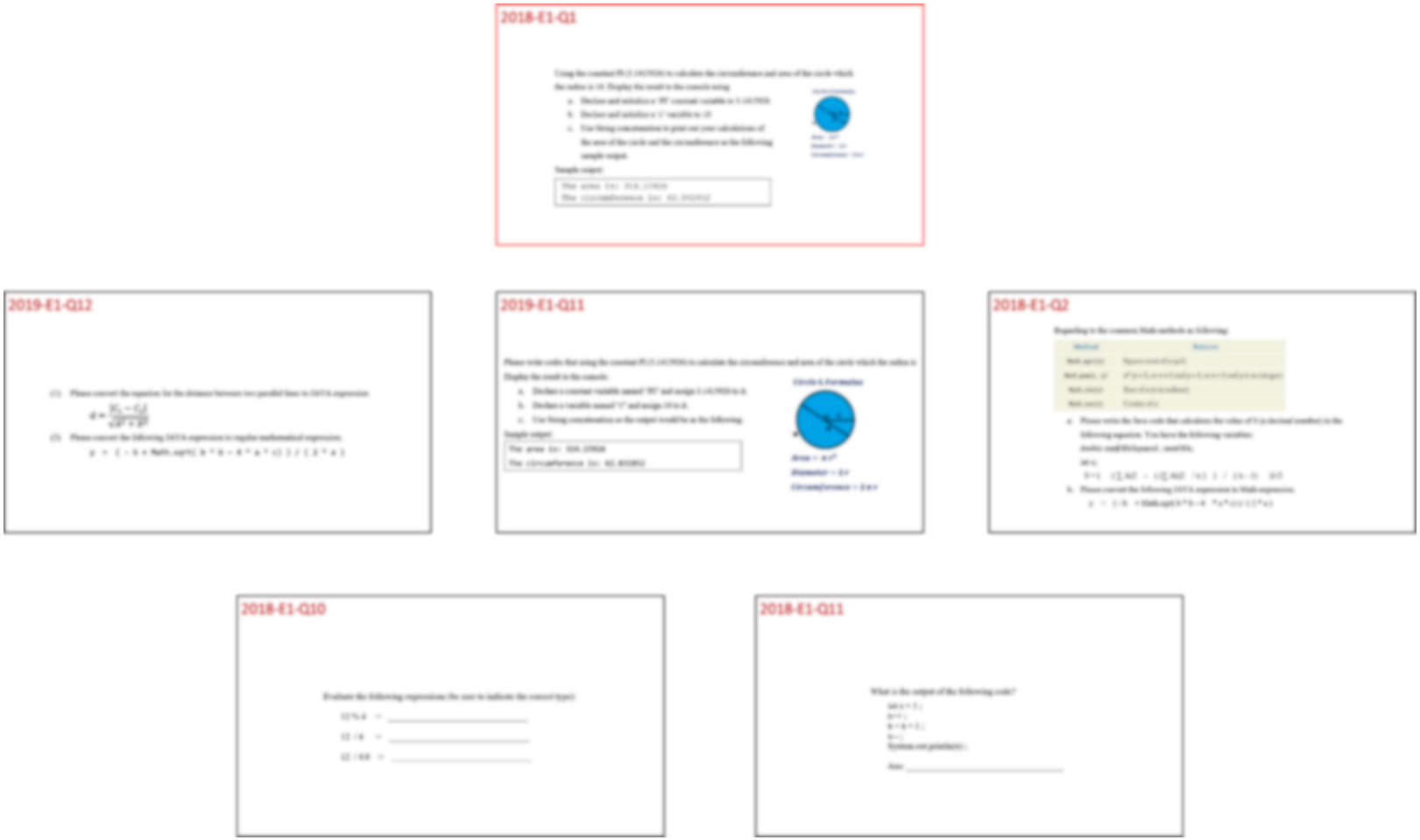


Figure 24. Top 5 Relevant Questions of an Example Test Item

Note: Question in red is the input while those in black are the candidate relevant questions arranged from top to bottom, left to right, as determined by PRIME. The test questions in the figure were intentionally blurred in this document to preserve their integrity.

topic performance weights $\mathbf{W}$ of γ will be used to identify an answer with a similar performance on the various topics. The distance can be calculated using Euclidean distance. Afterward, it will be arranged in ascending manner. Finally, there still is a possibility to see ties at this point. Therefore, the last step is to look into the student profile (i.e., topic mastery levels $\mathbf{S}_{\pi_q}$) of the owner of the answer at time point π_q (see Section 5.4.2.2.3 for the discussion). The distance between the profile of the answer's owner is and γ is computed using (5.8). The values are arranged in ascending order. At this point, based on the multi-criteria sorting employed, the top result will be marked as the appropriate answer to the question deemed to be relevant to the γ's critical item.

5.4.4 Underlying Assumptions

The following assumptions are made by this framework. First, that the course content is structured and consistent over the years. This means it follows a similar syllabus and deviations are minimal. Additionally, it assumes that the $\mathbf{Q}$ matrix is defined by an expert and that it is consistent with the syllabus. Finally, it is assumed that the granularity of concepts are consistent.

5.5 Methods

This section provides an overview of the experimental design along with the various evaluation used to validate the PRIME framework. The context in which is to be deployed is for the framework to support predictions online and in real-time with minimal to no additional effort on the teacher's part. As previously stated, the primary

goal of the development of this framework is to provide a principled approach to providing personalized items that are appropriate to the student's needs. This is the initial attempt in doing based on the context of summative assessments where students are assessed at predefined time points in the semester based on topics outlined in the class syllabus.

This framework aims to address some of the limitations of existing methods given the current context. First, questions are complex. Students are provided limited learning opportunities to demonstrate their mastery of a topic given that these involve exams. The student performance prediction in this framework aims to predict at a finer grain level, particularly the score that will be obtained on an item as opposed to successfully answering it correctly or not. Given that the framework proposes to address some of the limitations of existing modeling techniques, fitting the current dataset to such models would not be feasible as it would result in losing some contextual information such as partial credit. Therefore, instead of providing a comparison of the results of this framework to existing ones, this investigation solely focuses on providing a comprehensive exploration of the behavior of the framework and understanding the boundary conditions it can support. Additionally, The goal of this evaluation is not focused on measuring the learning effect of having the learning activity as it requires the existence of a valid approach in doing so. Thus, the initial step investigates the validity of the framework before proceeding to the next step which is to measure any learning effects.

5.5.1 Evaluation of Individual Components

It is important to note that evaluation in AI research is often thought of only in terms of how well the system performs. However, as Cohen and Howe (1988) argue, these are not typically confined only to performance measures. In fact, in the research community, describing how programs work and the various problems encountered faced along the way are also valuable. Cohen and Howe (1988) provided some experiment schemas and guidelines in conducting the evaluation of AI systems. Considering how the framework is composed of multiple components, it was deemed reasonable to have these components tested independently. Some evaluation approaches on AI systems include layered evaluation as done by Sosnovsky and Brusilovsky (2015). Another is through an ablation study which investigates how the system overall performs once certain components are removed (Newell, 1975, as cited in Cohen & Howe, 1988). Lastly, another is through limitations studies where the system is provided with unusual cases to identify the behavior and thus would uncover the boundary cases.

There are two main components of the PRIME framework. The two may appear to be interdependent, but they are in fact two independent components that just happen to be arranged in a certain order. The first component focuses on predicting the performance of a student on a future test. The second component focuses on recommending a relevant test item to a student. Both rely on historical data. The second component can be said to be informed by the first component. The second component can be viewed as a recommendation problem. The first component, although the goal is to make forecasting, the underlying idea is that of a recommendation problem. To be more specific, it follows the idea of a neighborhood-based forecasting problem. The evaluation of the entire framework was divided into two, one for each component.

Given the nature of the task being solved by each component, its corresponding evaluation was done. The first component was objectively evaluated while the second component was subjectively evaluated.

5.5.2 Longitudinal Approach

The experimentation was framed to follow a retrospective approach. Essentially, the study withholds the data gradually and proceeds as if data comes into the system. Given the longitudinal approach of the data, this investigation follows a walk-forward validation method to evaluate the framework. This involves doing the forecasting which involves redoing steps using historical data prior to the current time step which is a combination of a rolling window analysis and expanding window analysis for time series data often leveraged in the literature of financial trading. Afterward, the same observation will become part of the historical data and the process continues. This closely resembles the actual utilization of the system and the framework. Since the complete data have already been collected, this will be held out during the process and will only be used during the relevant time step and to compute for the error. RMSE will be used to measure the errors as well as the mean signed error.

There was three years' worth of data, the first data will form the preliminary content of the library. Afterward, the following year will be the first to implement and deploy the model. The final year will utilize the two prior years' data for the task. To be more clear about it, when populating the library, the library $\mathbf{L}$ constitutes all the student models in the database. It accumulates all information collected by the system over the years. Each student in the dataset is assigned an ID along with their year. The experiment was framed to begin in the second year since the first

year was simply used to populate the library. The process was applied and evaluated. Afterward, it became part of the library and the process was repeated for the year 2020. Thus, the library was composed of all the historical events that transpired in the years prior to it. Additionally, given that in a year there are only three time points that contain evidence, the experiment was framed such that the next test (at time point π_q) was the performance of the student that the model should predict. Thus, in a year, there were only two time points of interest: E2 and E3. E1 cannot be forecasted because it was considered a cold-start problem (i.e., no other information is known about the student).

5.5.3 Objective Evaluation

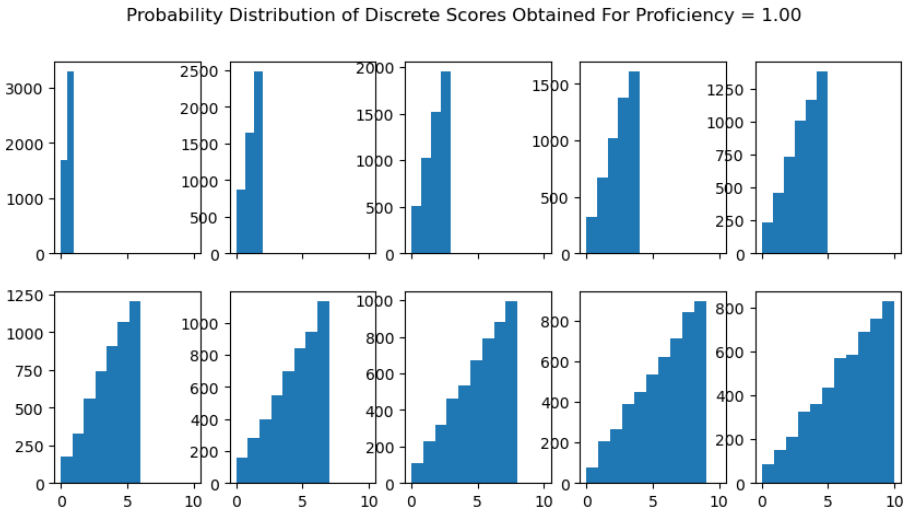
The objective evaluation pertains to those where ground truth is available. Particularly, in this context, the actual scores obtained by each student for each question on a test. It also includes the mastery levels derived from these performance data.

5.5.3.1 Dataset of Real and Artificial Students

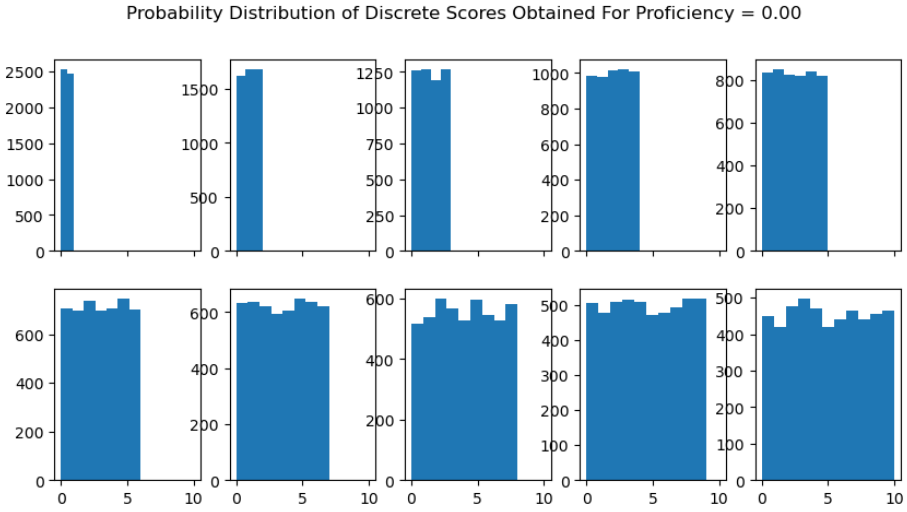
As previously mentioned, the data have been collected already from prior years. It was used retrospectively in this experiment. A detailed discussion is provided in Section 5.3. The first component focuses on understanding the predictive performance of the framework. Using the curated dataset, predictive accuracy can be identified. In addition to this, it is worth investigating how PRIME behaves when various students are encountered. However, with only a limited number of students and the difficulty of accurately capturing the real mastery levels of students, the evaluation also employed

simulated or artificial students, similar to that of VanLehn et al. (1998). Students were generated at the conceptual level of each test following the algorithm defined in Listing B.1. A student was instantiated primarily by randomly assigning them a level of proficiency sampled from a uniform distribution. Afterward, this proficiency level adjusts the bandwidth of the student's probability of obtaining a particular score, particularly in questions that require partial credits. In cases when the score has a fractional component, it is truncated to accommodate the algorithm. The idea behind the weighting follows that of a decreasing reciprocal association of weight. In essence, the higher the proficiency of the student, the higher the probability of obtaining a higher score as compared to that of a binary outcome. On the other hand, a lower proficiency reduces the determining of the score to an equally likely chance. For example, the distribution of getting a particular score of a student who has a complete proficiency (Figure 25a) is geared towards getting full credit while a student who has no proficiency (Figure 25b) has an equal chance of getting any possible score. This attempt to account for partial credit in simulation goes beyond the typical approach to simply simulating binary outcomes such as success. Unlike other simulations, as the goal is to investigate the robustness of the framework, all possibilities were accounted for instead of generating a set of simulated students drawn from a predetermined distribution (i.e., often the normal distribution). Finally, it is noted that there are indeed some correlations between certain concepts in a given test. Given that this investigation aims to serve as the baseline, incorporating this additional information in the simulation is reserved for future work[5].

[5]See https://sdv.dev/Copulas/ for more information on Copulas.



(a) High-Proficiency



(b) Low-Proficiency

Figure 25. Probability Distribution of Simulated Students Adjusted Based on Randomly Assigned Proficiency on Various Maximum Item Points

5.5.3.2 Evaluation of the Unified Student Model

The first aspect of the framework involves developing an individual student growth model that will encapsulate student performance data to demonstrate a student's mastery of the various topics. As a result, the unified student model (5.4) was developed and discussed in Section 5.4.2.1.4. A total of 371 student models were instantiated where several parameters associated with the student were obtained. Afterward, each model was used to predict a student's mastery level at every time point in which the student took a test. Thus, it was possible to obtain the difference between the predicted $\hat{\mathbf{M}}$ and the actual $\mathbf{M}$ mastery levels for each topic for each time point. Specifically, a student had three $\hat{\mathbf{M}}$-$\mathbf{M}$ pairs. $\mathbf{M}$ denotes the cumulative level for each t topics.

5.5.3.3 Student Performance Prediction Error

After identifying a candidate profile, the system proceeds to predict the performance of a future test provided the $\mathbf{Q}$ matrix is known. It should be noted that no predictive model was developed. Rather, predictions were made based on parameters transferred from a candidate profile, specifically based on the growth of mastery. As such, the accuracy is deemed to be influenced by the ability to choose the best candidate profile. Since the actual test results were available, these values served as the ground truth for measuring the errors. Unlike the typical workflow of measuring either the mean absolute error (MAE) or mean squared error (MSE), the errors are analyzed with their sign. Importance was given to understanding how the system does its forecasting over a variety of student proficiency levels, particularly whether it overestimates or

underestimates given a particular λ. Accordingly, this is referred to as mean signed deviation (MSD).

5.5.4 Subjective Evaluation

The subjective evaluation pertains to those where ground truth is not available. It also covers instances in which certain terms are operationalized under a particular assumption. Particularly, in this context, evaluating a recommender system involves defining what constitutes relevancy. Due to the retrospective nature of the evaluation, certain assumptions were made.

5.5.4.1 Assessing the Ability to Identify Relevant Items

As a result of predicting the performance of students on a variety of items, the system recommends a relevant item from the library for students to work on. This item is essentially similar to one expected to appear on an upcoming test. The system should therefore identify the item on which it believes the student will perform the poorest. As a result, the problem being addressed is a recommendation problem in which the system is supposed to provide students with recommendations that are relevant to them. Since actual users were not involved in evaluating the recommendations, relevance was operationalized as those items where the student had the lowest scores, which served as ground truth. This assumption was derived from the findings in Chapter 2, which indicate that high-performing students tend to review items on which they made mistakes. Furthermore, this aligns with the primary purpose of the framework, which is to assist students in identifying their misconceptions.

This problem can be formulated as a ranking problem provided by a defined utility function (e.g., the critical item defined in Section 5.4.3.2). Based on the top K recommendations, the system can be evaluated based on the coverage and the order of items recommended based on their relevance for the user. Specifically, in this context, items that require immediate attention from the student. There are several ways in which this can be assessed. It is possible to calculate rank correlation coefficients using Kendall Tau. It requires, however, that the lengths of the two lists be equal. If the library is large enough, recommendations that are irrelevant or at the lower end are still examined, which could lead to an unnecessary penalty. This evaluation takes into account only the top K items. Items below this are not considered. In this case, a commonly used information retrieval metric, the mean average precision at K (MAP@K), was used as similarly done by Thaker et al. (2020).

5.5.4.2 Defining a Baseline Recommender

In evaluating the recommenders, only the ranking was considered. The actual score became irrelevant. Therefore, all the scores were replaced with rankings using the `rank()` function of the pandas[6] library. However, an exploratory analysis of the dataset revealed that there were instances where the normalized performances of a student brought forth a tie. In terms of relevance, all those whose ranking was less than or equal to K were marked as relevant. On the other hand, when making the recommendation, the system followed the same approach by ranking the predicted scores of each test item. In the event of a tie, the system worked its way through from $1 \ldots K$. It followed the given algorithm in Listing B.2 in doing so. The algorithm

[6]https://pandas.pydata.org/

worked by randomly picking items from i-th rank level and adding them to a container until K items had been collected. Each new rank level begins with the exhaustion of all items from the i-th rank level.

5.5.5 Question Relevance Evaluation by Subject Matter Experts

The ground truth becomes increasingly subjective toward the end of the framework. The next part, which involves determining whether two items are similar, presents a challenge. As noted earlier, determining item similarity is challenging due to the lack of a standard approach (Pelánek, 2020). Similarly to the earlier discussion of candidate profiles, item similarity in this context can be viewed as a recommender problem, including one that involves ranking items according to a utility function (5.22). Contrary to the student profile, where ground truth was available, the only information available for questions was the distribution of topics within them. As a result, it requires expert evaluation. Nine teachers with experience teaching introductory computer programming courses in higher education were recruited to rate the quality of the items identified as relevant by PRIME. The average teaching experience was 11.89 years ($SD = 7.01$). 20 questions were randomly selected from the 158 questions to ensure that each exam was represented equally (15%). The four top candidates were presented to an expert who then assessed their relevance to the question on a five-point Likert scale. It was not disclosed to them that these candidates had been ranked in a particular order. The only instruction they received was to evaluate the relevance. The full instruction is provided in Appendix C.

5.6 Results and Discussion

To evaluate the feasibility and impact of PRIME, this section presents details of the evaluation findings. Specifically, the testing was conducted in the context of a practical and straightforward approach to providing students with tailored support in preparing for an upcoming test without additional work by the teacher. Each component of the framework was independently evaluated. There are three main parts to this section, each presenting the results of the evaluation of a component and answers to the corresponding research question. The section concludes with a general discussion.

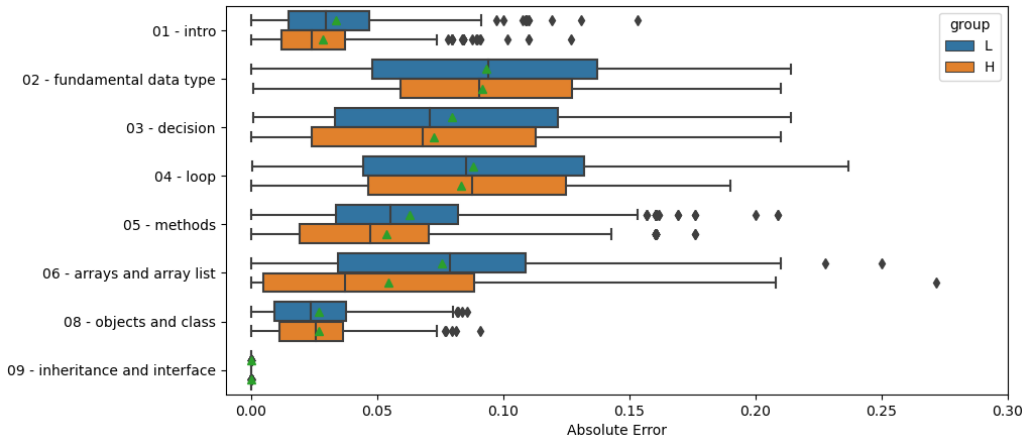
5.6.1 Growth on Student's Mastery Level Can be Estimated From Test Performance Data

One aspect of PRIME involves using student performance data, particularly test scores on complex items where multiple topics are assessed and typically allow for partial credit. A unified model was developed to encapsulate the student data to facilitate modeling a student's growth in mastery of the topics (see Section 5.4.2.1.4). To test its validity and to answer **RQ D.1**, all the student models from the real students were extracted from the library to test the ability of each model to estimate mastery of a topic at a given time point. Figure 26 provides an overview of the errors between the predicted and the actual mastery levels according to topics. The results were divided into student groups. Figure 26a illustrates the absolute errors. Unsurprisingly, Topic 9 had the lowest error as this topic was only assessed during the last examination, suggesting that it may be due to overfitting. Topics 2, 3, and 4

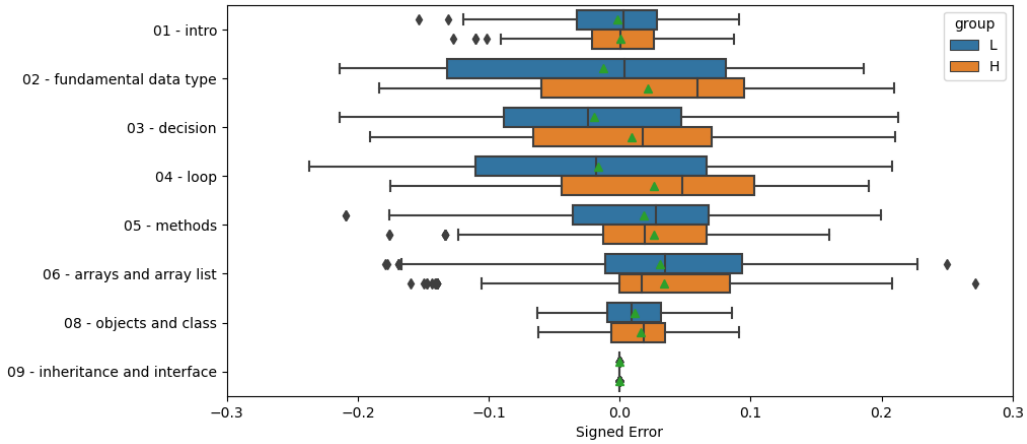
had relatively high errors. However, these three topics were constantly evaluated in each examination throughout the semester. On average, the absolute error for a topic mostly lies below 0.10 points. Thus, it suggests the potential of the unified growth model in summarizing all of a student's performance data into a set of parameters that could easily be used for forecasting and has the potential to work at scale. Note, however, that the model assumes that a student does not have any mastery at the beginning. It is acknowledged that certain students do already have prior knowledge early on in the semester. An investigation of the effect on the model when this is accounted for is reserved for future work.

5.6.1.1 More Evidence Leads to More Conservative Estimations of Mastery Levels for Low-Performers

In this analysis, special interest was given to the sign of the error as it provides an idea of whether the model underestimated or overestimated its prediction. Particularly it is worth looking into whether a difference exists between how it performs for a certain student group. As can be seen in Figure 26b, Topics 2, 3, and 4 had a clear distinction between which direction a particular student group lies. Clearly, the model has the tendency to overestimate a high-performing student's mastery of these three topics. On the other hand, a low-performing student's mastery is often underestimated. A series of Mann–Whitney U tests was performed on each topic and the results indicated statistically significant differences ($p < 0.05$). This finding suggests the conservative nature of the model in making predictions about low-performing students as more pieces of evidence become available. When such predictions are conveyed to this group, they do not become complacent about their preparation. Considering how this group



(a) Absolute Error



(b) Signed Error

Figure 26. Overview of Mean Absolute and Signed Errors of the Unified Model

often overestimates their own performance (i.e., Dunning–Kruger effect; Dunning, 2011; see Chapter 4), this behavior of the model can help counter such effects if the ultimate goal is to provide students with appropriate learning resources. In contrast, a further investigation should be undertaken on the effect of the overestimation of the prediction on the part of the high-performers. Particularly, how does this interact with certain phenomena often attributed to this group, such as *imposter syndrome*?

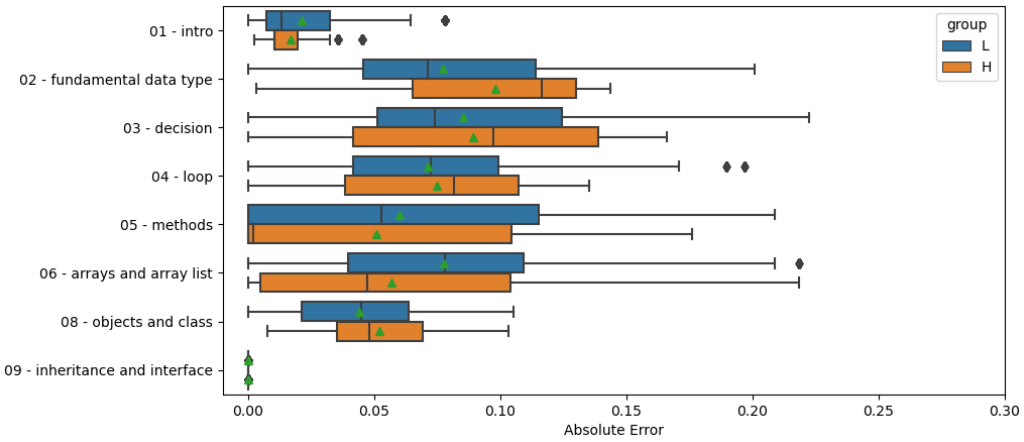
5.6.1.2 Modeling Approach Performs Consistently Regardless of Student Profile

To further explore the nature of the unified model, 5,000 random artificial students who were hypothetically enrolled in 2020 were leveraged. The algorithm used is described in Listing B.1. The goal is to generate as many students to cover as many profiles or stereotypes as possible. As a result, these students will have varying proficiencies which affect the scores they obtain for each KC in the questions. Also, these students will also have their corresponding overall performances λ.

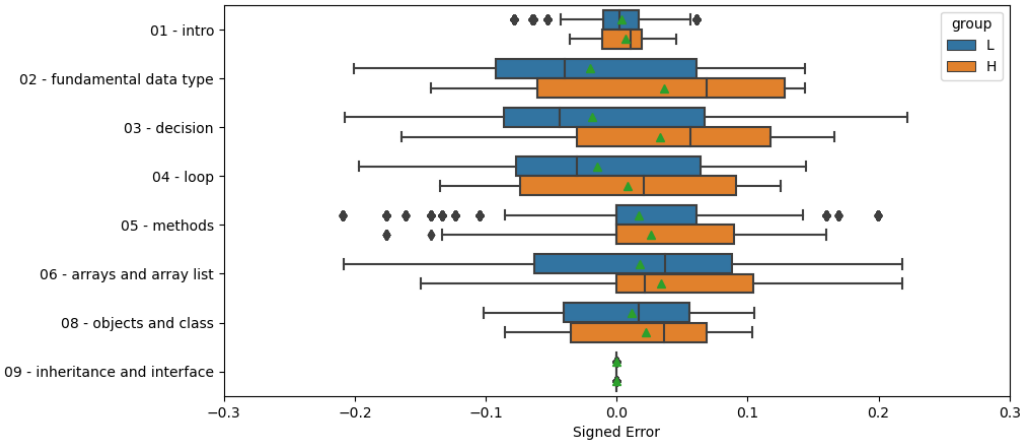
The nature of the errors of the unified models for these students was looked into. Figure 27 illustrates both the absolute and signed errors grouped according to topics. Interestingly, the trend exhibited for the artificial students was similar to that of the real students illustrated in Figure 26. This suggests that not only it was feasible to summarize the performance data into a unified model for real students, but it is also possible to do so for artificial students. Remember that these artificial students follow a different distribution than that of the real ones. Thus, it may be the case that the approach works for any student profile.

5.6.2 Student's Performance on Complex Test Items Can Be Predicted Using Parameters from a Candidate Model

The next component relies on the existence of these candidate models in the model library **L**. With a new cohort of students, these models will be used to forecast a new student's performance on a future test. Moreover, these predictions attempt to forecast the actual scores of the students instead of simply predicting success. To test this and to answer **RQ D.2**, a rolling and expanding window analysis was performed.



(a) Absolute Error



(b) Signed Error

Figure 27. Overview of Mean Absolute and Signed Errors of the Unified Model of Artificial Students

Specifically, the years 2019 and 2020 were used to determine the accuracy of the predictive approach. There are two steps in this process. First, a candidate profile needs to be determined. Afterward, prediction can take place.

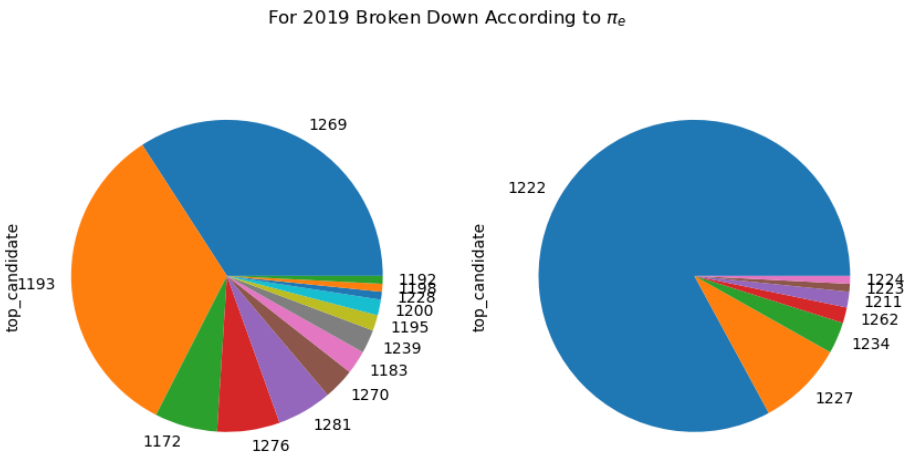
Table 13. Overview of Candidate Profiles

	2019	2020
Student Count	123	124
Library Model Count	124	247
Unique Candidates (Coverage)	20 (0.16)	35 (0.14)
E2 Forecast	13 (0.11)	26 (0.11)
E3 Forecast	7 (0.06)	9 (0.04)
Average Candidate λ (SD)		
E2 Forecast	0.89 (0.02)	0.86 (0.06)
E3 Forecast	0.54 (0.07)	0.58 (0.08)
Average Signed $\Delta\lambda$ (SD)		
E2 Forecast	0.11 (0.10)	0.08 (0.08)
E3 Forecast	-0.24 (0.10)	-0.20 (0.09)

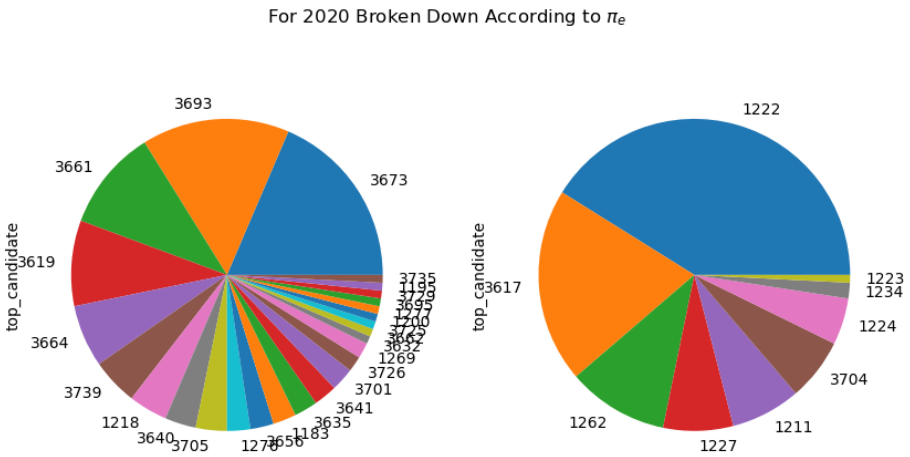
5.6.2.1 Relatively Few Profiles Were Recommended

Finding a candidate profile is framed as a recommender problem. One issue that is being faced by recommender systems is the coverage of the items from the entire list that is selected and eventually recommended to its users (Aggarwal, 2016). It is often attributed to data sparsity. In the current framework, a list of top candidates can be determined by sorting the weighted distance measure (5.9) against all profiles. For this analysis, only the first candidate is focused on. Essentially this is the model which has the lowest weighted distance from the student. The breakdown of the two years is summarized in Table 13.

For each year, the system was designed to do two forecastings (i.e., two time points π_q). The first is to predict the result of E2, given the performance information from E1. The second is to predict the result of E3 given the performance information from both E1 and E2. Figure 28 provides an illustration of the top candidates as well as the distribution of the year from which this candidate profile was from. Interestingly,



(a)



(b)

Figure 28. Top Candidates Associated to Students

Note: The chart on the left represents E2 while the right represents E3.

candidate 1222 has been popularly and consistently selected. Thus it is worth looking at this profile.

Figure 29 illustrates the topic growth of student profile 1222. This student has a $\lambda = 0.52$. Clearly, this student has had a below-average performance throughout the semester. In fact, when broken down by exams, it becomes clear that 1222 is more

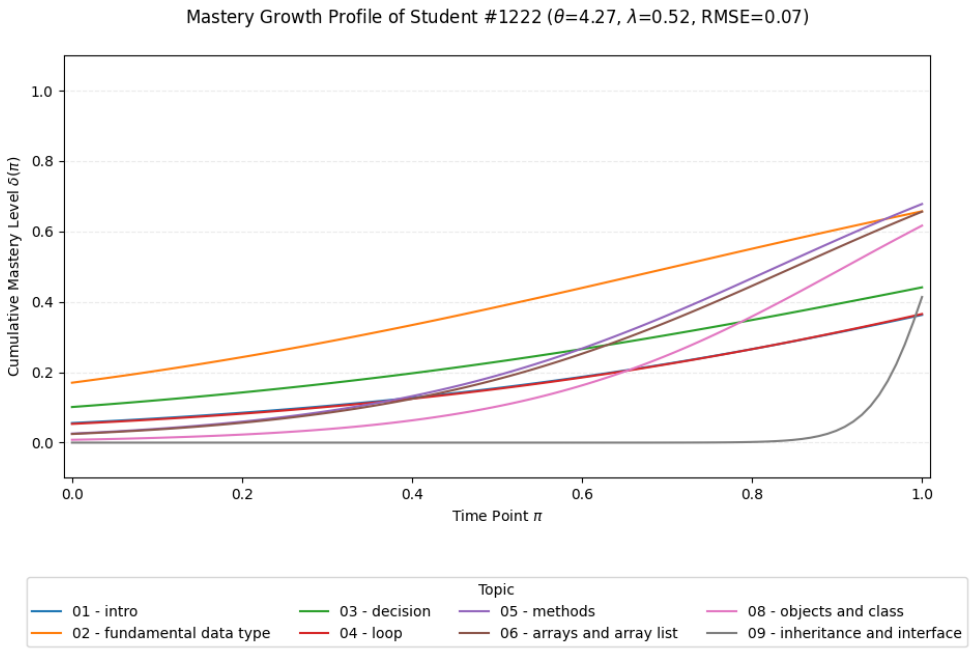


Figure 29. Growth Model of Top Candidate

prominent when forecasting the result of E3. Therefore, future students will certainly benefit from the mistakes done by this student. However, this objective is far along in the pipeline. For now, the main goal of the candidate selection process was to identify a *similar* profile.

Since these top candidates were former students who took the same class, it was possible to recover their overall performance λ. This would provide insights into the average performance of the profiles being recommended by the system. Table 13 summarizes the averages by tests and year. During E2, the profiles associated with the students tend to belong to the high-performing group. On the other hand, during E3, the profiles mostly belong to the low-performing group. Notably, the same trend persisted in the succeeding year. Two perspectives can be taken into account when analyzing this finding. First, the ability to find a similar profile may warrant further

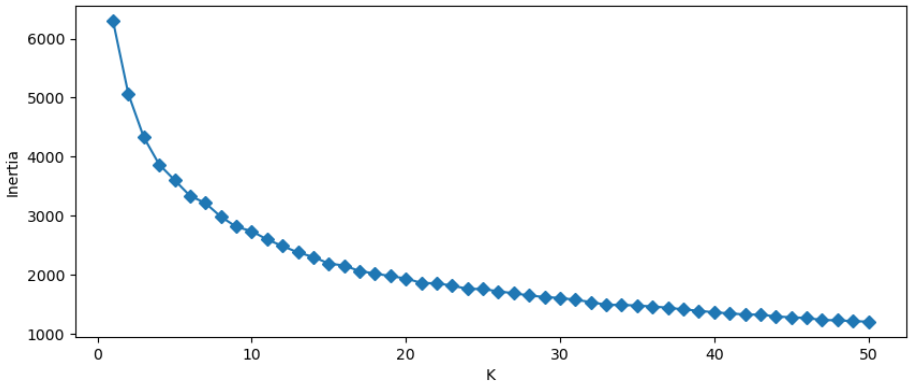
adjustments. But, on the other hand, such a trend could be utilized in informing the succeeding tasks. Therefore, these two perspectives are taken into account in subsequent sections.

5.6.2.2 Upper Limit on the Number of Unique Student Profiles is Unknown

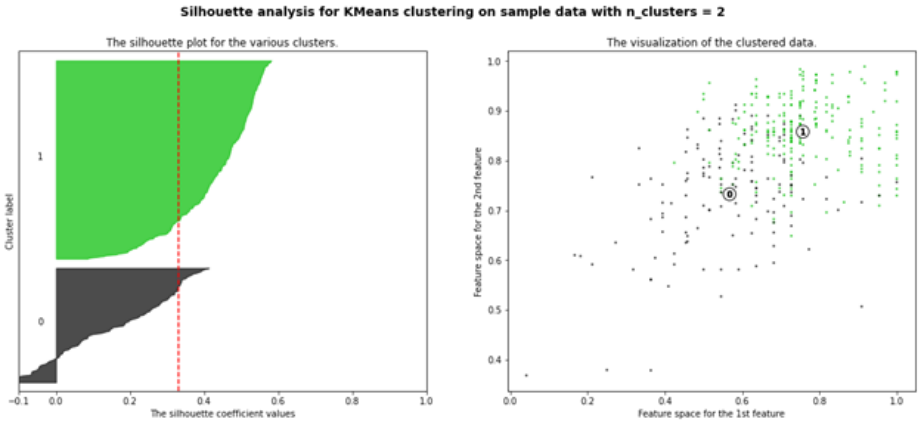
If the library is expected to expand every year as new students are added, it seems logical to expect a more diverse list of candidate profiles over the years. Based on the findings in the previous section, it is worth investigating why the system had limited unique profiles when it made the recommendations. Could this suggest that there are only so many profiles that exist in the course (i.e., stereotypes)?

To further understand the behavior of the system, a clustering algorithm, particularly K-Means was applied to the profiles of the students in the library. K-Means is an unsupervised approach and since the number of profiles (K) is unknown, it had to be looked into from the data itself. There are two approaches often employed. The first is the elbow method while the second is through the use of silhouette scores.

In this exploration, the final mastery levels $\mathbf{M}$ of all students were used as it was consistent with their overall performance. In this case, each student was represented with eight features denoting their final mastery level for each of the eight topics. No rescaling was done as these values fall within the same range of values from 0 to 1. The result of the Elbow Method where K-Means was run from 1 to 50 to identify the optimal number of clusters (K) is illustrated in Figure 30a. The figure shows the corresponding inertia, or the sum of the squared distance of each point to their corresponding centroid, based on a given K. It can be seen that it resulted in a somewhat smooth curve making it challenging to identify the optimal number of



(a) Elbow Method



(b) Silhouette Score Analysis

Figure 30. Identifying Optimal K Profiles for K-Means

clusters with confidence. However, it can be observed that the improvement slowed down past $K = 4$.

Another approach done was by comparing the various silhouette scores of the different K. Figure 30b illustrates that $K = 2$ had the highest silhouette coefficient. What this likely suggests is that it somewhat reiterates what has already been found in the literature of programming learning that indeed, there is a bimodal distribution of students (Robins, 2010). What this means is that knowing this upper-bound number of profiles not only ensures the framework works at scale, but could also provide

valuable insights for the teacher as they could provide tailored sets of instructions to these particular profile groups or stereotypes. However, for now, it is inconclusive whether there is an upper limit to this based on the current findings.

5.6.2.3 Candidate Profile Can Potentially Be Used to Inform the Difficulty of a Task to Recommend

A student's overall performance λ is unknown up until the end of the semester. However, in this offline evaluation, these are known. Therefore, it is worth looking into how similar the student is to the candidate model based on the difference in their λs. This difference will be referred to as $\Delta\lambda$.

$$\Delta\lambda = \lambda_{\text{candidate}} - \lambda_{\text{student}}$$

As with the earlier analysis, the sign or the direction is of special interest and was retained. A negative value indicates that the candidate profile was less proficient relative to the current student. Thus, the student's λ was underestimated. Similarly, a positive value indicates the opposite. Table 13 provides an overview of the average performance difference during the two time points of the two years.

Interestingly, a similar trend for the two years was observed when the system forecasted the students' performance for E2. The positive $\Delta\lambda$ suggests that the system had the tendency to pair a student with a candidate that was relatively more proficient. This means that eventually, the items that will be recommended will have fewer mistakes as these belong to better performers. On the other hand, when the system made a forecast for E3, it was the other way around. Students were mostly paired with those that were relatively less proficient. As a consequence, the system will end up recommending items containing more mistakes. Considering that the

overall performance accounts for all the exams, these numbers suggest that early on, the system overestimates the likely projection of the student's performance in an upcoming exam. However, as it collected more performance data, it became more conservative and underestimated its forecast.

In light of the purpose and timing of making the forecast, it is reasonable to provide students with easier items at the beginning of the process, while they are still learning the domain. Overwhelming them early on could affect their cognitive load. It is expected that by the end of the semester, students will have mastered the majority of the topics. Therefore, they would be capable of dealing with being exposed to more mistakes to determine whether they had truly mastered what was expected of them. As a result, this behavior of the system appears to be promising. Therefore, a deeper look at the relationship between the two λs was looked into. Two separate Pearson correlation tests were performed, one for each test. The results suggest that there is a significant correlation between the two variables for E2 ($r = 0.47, p < 0.05$) and E3 ($r = 0.47, p < 0.05$). This encouraging finding provides initial support for the behavior of the system in which it is able to provide a similar profile to a student. However, further investigation is needed.

5.6.2.4 Expanding the Library Led to an Improvement in Identifying Candidate Profiles

Another aspect that was looked into was whether there was a difference in the average $\Delta\lambda$ between the two years. It is expected that as more students are added to the library, the system's collection will expand and lead to providing a closer profile to that of the student. Therefore, a Mann–Whitney U test was performed to the

average of the two exams for the two years. Both suggest that there was a significant difference between the average difference for E2 in 2019 and E2 in 2020 ($p < 0.05$), and the same goes for E3 ($p < 0.05$). It can be seen that the difference lowered as more profiles were added to the library. This result is encouraging considering how similar this is to that of a human experience. As a person gains more experience, they become better at performing a task. Comparing the performance differences by years yields an overall of around -0.07 as summarized in Table 13.

It is important to note that performance differences are still referred to as errors given the objective is to identify a similar profile. By disregarding the signs, a similar trend can be found. As provided in Table 13, the errors significantly lowered in the succeeding year when more profiles became available in the library. These findings suggest that as more profiles were incorporated into the library, its ability to provide a better candidate model improved. It may be a consequence of introducing a new profile or stereotype in the library that was previously not available. However, an upper bound of the number of profiles in a particular class is yet to be discovered.

5.6.2.5 Student Ability Should be Taken into Consideration When Determining the Difficulty of a Recommended Task

Motivated by findings in the earlier sections where the model had varying tendencies depending on a student's proficiency, a closer look at the $\Delta\lambda$ was done. Particularly, how does it correlate to the student's λ? Figure 31 illustrates the relationship between the two variables and grouped according to tests. Essentially, the x-axis is the student's λ while the y-axis is the $\Delta\lambda$ which ranges from -1 to 1. Interestingly, a significant negative correlation ($r = -0.416, p < 0.05$) was found. What this suggests is that it is

the case that high-performing students were associated with profiles of which that are lower than theirs. As a consequence, these students tend to see more errors. Similarly, low-performing students were associated with profiles that were geared towards either those close to them or those who performed better. As a result, these students would see fewer errors.

This provides a better picture of how the system provides a candidate depending on the student's ability and the amount of data it has about the student. Perhaps there may be more appropriate methods of determining difficulty when recommending an item than simply maximizing gains. Pedagogically, this approach to identifying the level of difficulty exemplifies the ZPD where students are provided with problems that are within their ability and would not require additional help (Vygotsky, 1978). This helps lower-performing students' cognitive load to be manageable (Sweller, 2011). Additionally, this finding is in consonance with that of the fading worked example principle and prevents the expertise reversal effect (Kalyuga, 2007; Kalyuga et al., 2001; Renkl, 2002). If an item contains more mistakes, it translates into a problem-solving activity. On the other hand, if an item contains fewer mistakes, it translates into a worked example. Knowing the right amount of difficulty for a student would hopefully prevent students from disengaging from the learning activity.

Thus far, the findings obtained demonstrate the possibility of identifying a profile from which recommended items will be derived. The signed performance differences could be used to guide the process. Nevertheless, the same measure suggests that the component's goal of identifying a similar profile from the library may still need to be improved.

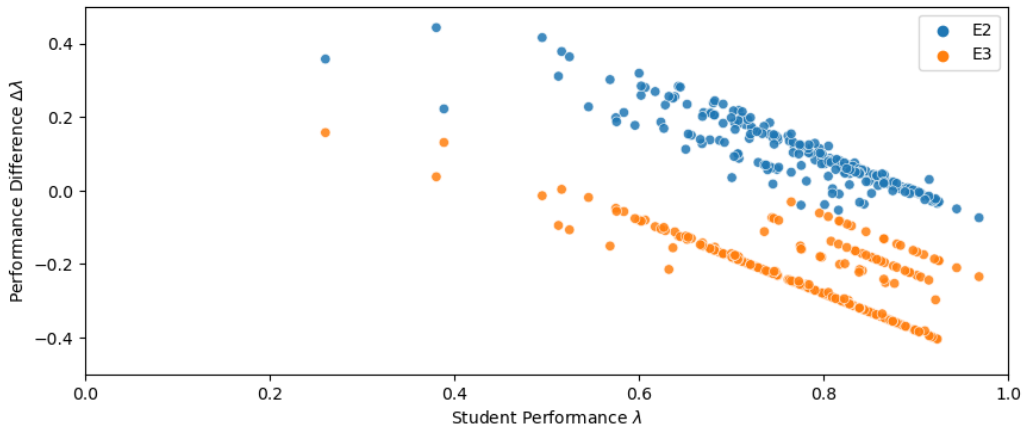


Figure 31. Correlation Between λ and $\Delta\lambda$

5.6.2.6 An Item's Grading Method Impacts the Accuracy of Predicting a Student's Score

Among the many goals of the system is the ability to predict student performance on multi-skill items that require partial credit. As opposed to simply determining the probability of answering an item correctly, it aims to determine the degree of correctness of a student's answer. The steps to accomplish this are outlined in Section 5.4.2.2. In essence, the normalized gain $\Delta\delta$ of the candidate model is transferred to the current student's level to forecast the student's mastery level, which is used to predict item-level performance on a test. Figure 32 illustrates the distribution of the signed errors on each item in the two tests from the two years. Both the mean (blue) and the median (red) are shown for reference. As previously stated in Table 12, certain items were graded on a binary basis. As a result, the student may either receive full marks or not. Item numbers for these questions are highlighted in red. This determination, however, was not made by the teacher. It was determined based on performance data. If the grader assigned no marks and only a full mark, it would be considered binary, otherwise, it would be considered non-binary.

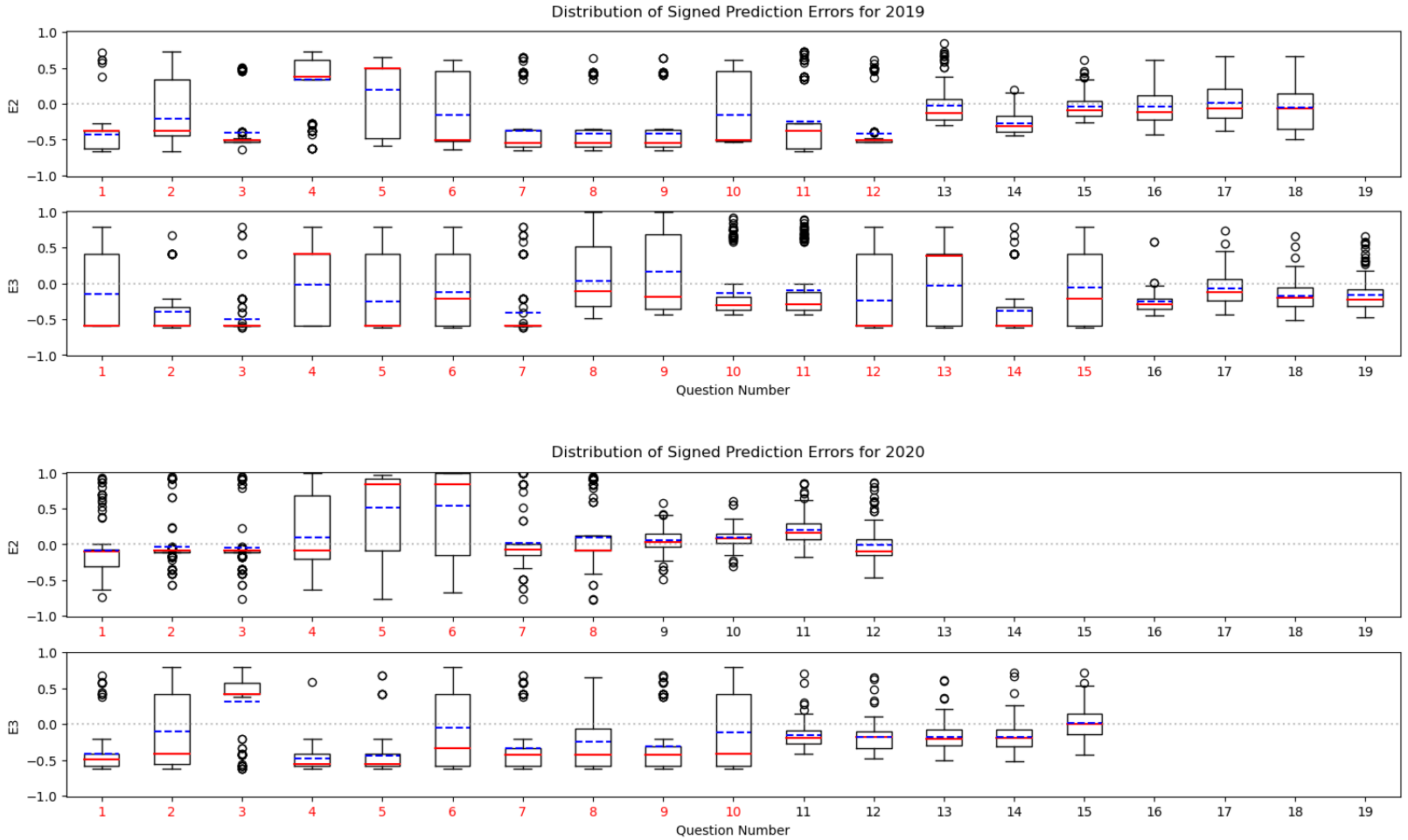


Figure 32. Distribution of Signed Predictions Errors For the Two Exams of the Two Years

Note: The x-axis denotes the question numbers while the y-axis denotes the signed errors. Questions in red indicate binary grading. The median is indicated by a red line and the mean by a blue line.

A preliminary visual inspection of the distributions in Figure 32 suggests the apparent distinction between binary questions (question numbers in red) and those with partial credits (question numbers in black). The errors were grouped accordingly to whether the question was binary or not. Both the average and midpoints of the two groups were compared. The result of a Mann–Whitney test suggests a significant difference ($p < 0.05$) in the prediction errors (or MSD) on non-binary questions ($M = -0.07, SD = 0.25, Mdn = -0.12$) and on binary ones ($M = -0.14, SD = 0.49, Mdn = -0.35$). As both values show negative signs, it appears that the system underestimates the scores of students when making forecasts in line with an earlier finding.

In addition to understanding the predictive accuracy of the system, it is also important to consider whether the nature of the question has any impact on the accuracy. Figure 33 visualizes the kernel density of the two question groups. There are two apparent distributions in the error for the binary group, both of which are centered at -0.5 and 0.5. This can be explained by the nature of the question itself. It follows a binomial distribution that has an expected value of p. In this particular case, $p = 0.5$. The sign is a consequence of not disregarding the direction of the error in the current analysis. Moreover, item difficulty was looked into to determine its impacts. However, no significant patterns or differences were found in terms of the errors and how they were distributed. This suggests that the prediction errors are not likely due to the item difficulty for binary questions.

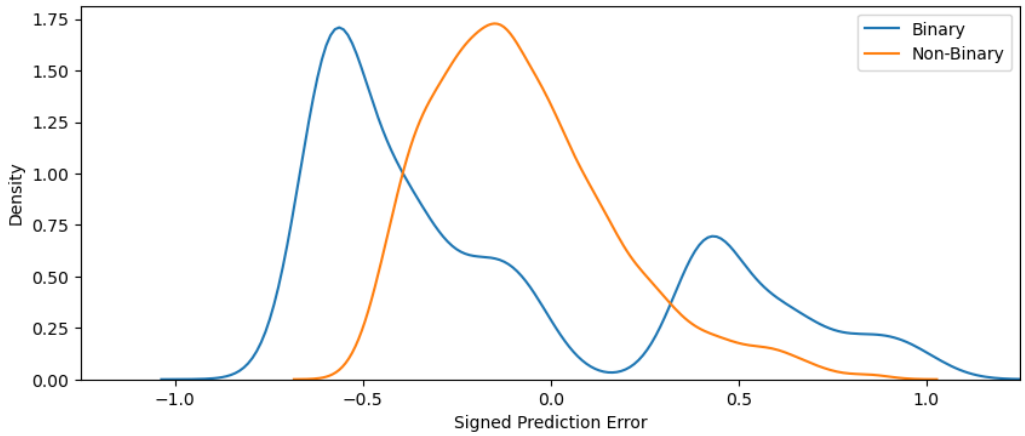


Figure 33. Distribution of Signed Predictions Errors Based on Type of Grading

5.6.2.7 Increasing the Number of Topics Assessed by an Item Improves the Accuracy of Score Prediction

The previous section described how an item is graded impacts the accuracy of the prediction. Most of the binary questions often had a single topic associated with them. On the other hand, non-binary ones often have multiple. To determine whether assessing multiple topics in a question impacts the accuracy of prediction, a Pearson correlation was performed. The results indicate a significant moderate positive correlation ($r = 0.49, p < 0.05$). Considering how error can either be positive or negative, this finding suggests that the system tends to overestimate its prediction of the student score as more topics were being associated.

A simple linear regression was conducted to determine whether the number of topics associated with an item significantly influenced the error rate on non-binary questions. The fitted regression model was:

$$\text{errors} = -0.235 + 0.0343 * \text{topic\_count}$$

The regression was statistically significant ($R^2 = 0.235, F(1,\ 15) = 4.615, p = 0.04$).

It was found that the number of topics significantly predicted error on non-binary questions ($\beta = 0.0343, p = 0.04$). Based on the fitted model, it appears that by default, the system underestimates the student's performance. As each question has a minimum of one topic, the model yields an underestimate of -0.20. As only eight topics are available in the course, the system would always produce an underestimation of student performance. As stated previously, if the main goal is to recommend additional resources, this should not be a significant concern. It is possible that students will be more motivated to improve their performance if they are provided with an underestimated performance.

In spite of the fact that the predictive accuracy was not as good as expected, pedagogically, it was better to assume that a student would underperform and thus motivate them to review rather than mislead them by overestimating their abilities. Due to the system's role as a recommendation system for supplementary learning materials, relatively low accuracy is not necessarily indicative of failure, since supplemental resources over and above what students are expected to utilize may be of value to them. As illustrated in Figure 32, complex questions requiring partial credit are more likely to be correctly predicted. Therefore, this component of the framework was able to achieve what it set out to accomplish. The accuracy of prediction for binary questions can be further improved by using this framework in conjunction with those typical knowledge tracing models since these are the types of questions those models cater to.

As a final point in evaluating the predictive accuracy, one should note that the prediction was not made taking into account the time horizon between two time points. Generally, the wider the gap between π_e and π_q, the higher the uncertainty, and thus it is suspected to lead to increased error in prediction. Due to a lack of data, this was

not evaluated. Also, the framework uses **W** to estimate the items' scores as discussed in Section 5.4.3.2.5. The points were equally distributed among all questions that related to the topic. In reality, it is possible that the distribution of points may not be uniform. There is a need to determine how topic points may be allocated to the questions which could be done through machine learning. This should be taken into account in future improvements. In doing so, the complexity of the alternative should be considered as well.

5.6.3 A Subjective Approach to Evaluating PRIME's Ability to Identify Relevant Items

Following the forecast of the likely performance of students on a future test, it is up to the consumer of the information to determine the best course of action. As discussed previously, PRIME consumes this information to provide a student with an appropriate worked example that contains errors. This example is intended to encourage the student to reflect on his or her own mastery as they prepare to take a test in the near future. To assess the relevance of such recommendations and to answer **RQ D.3**, the predicted and the actual scores of the students were compared. The performance of PRIME was compared against a baseline recommender system, which simply shuffles all items and returns the first K elements. The MAP@K scores of the two recommenders for varying K values were compared. The results are summarized in Figure 34, left.

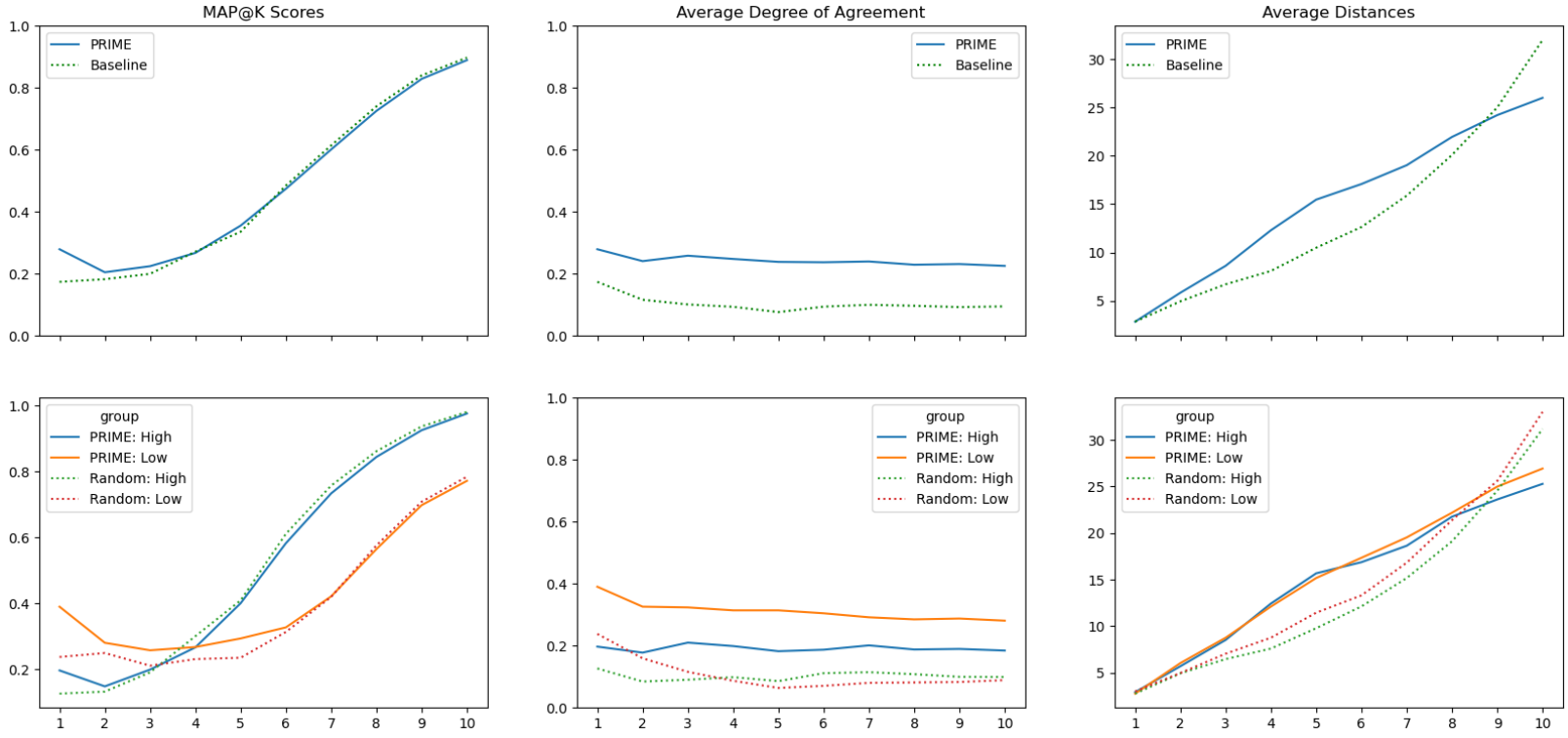


Figure 34. Performance Comparison of PRIME and the Baseline Recommender

Note: The x-axis denotes the K values or cut-off points while the y-axis denotes the corresponding metric score.

5.6.3.1 Low Performers are more likely to be Recommended Relevant Items by PRIME

It can be seen that by $K = 5$, the baseline MAP score began to match the PRIME score. Note, however, that the main purpose of the recommender is to provide students with appropriate items to master and not necessarily to cover all items. Due to this, this recommender system is typically used to select top items such as the top three. The graph clearly shows that PRIME performs similarly to a random recommender. Nonetheless, it can be seen that at earlier values of K, the system had a slightly better MAP score, albeit a relatively low one. Three points are worth mentioning. Considering that the ground truth used for the list of relevant questions captures more test items, the performance of both recommenders should converge as K increases. The operationalization of item relevance can be attributed to such a trend. Secondly, the overall low MAP scores are to be expected as a consequence of the use of a proxy and a strict definition of item relevance. Third, considering that the goal is simply to recommend a critical item (1 or 2), focusing only on those K values provides a slight improvement. Moreover, this suggests that on average, students receive 25% relevant results at K=1. In this instance, only assessment data were used. In the same way as other predictive models, if more information were provided to this model, it would perform better. Therefore, future work could examine the possibility of improving the system by incorporating additional information about the user.

As with earlier sections, the MAP scores were grouped by student performance to examine whether the system was capable of recommending relevant items to students of varying abilities. The results are illustrated in Figure 34, left. There is some interesting evidence suggesting that the recommender system performed better for low-

performing students. It is encouraging to see this trend because it is these students who are believed to benefit the most from the guidance provided. On the other hand, MAP scores were low in the high-performing group. The odds of the system identifying the relevant items for them are slimmer since they had already performed well. Those where many errors have been committed are prioritized by the system. As a result, if they are already high performers, it is understandable why the system would have difficulty providing them with the appropriate items. Considering that this is supplementary material, the recommender is primarily concerned with the first few items on the list.

5.6.3.2 Relevant for a Student: What Does It Mean?

A closer examination of the dataset where students were grouped according to their performance, year, and exams was done. It is supposed that increasing the number of students in the library should improve the system's ability to provide relevant items for review. As shown in Table 14, comparing the MAP scores between the years indicates that the ability to provide relevant information decreased for both student groups. This finding contradicts the earlier finding that expanding the library led to an improved ability to identify candidates. These findings, therefore, should not be construed as evidence of PRIME's poor recommendation capability. Instead, this result should be viewed as a recommendation to refine the definition of relevance. Human judgment would certainly provide more insight into this issue. Due to the inability to reach the original owners of the dataset, this finding should be regarded as an indicator of future research. Alternatively, this pattern may also serve as a potential diagnostic tool. If it is suspected that these recommendations are truly

Table 14. MAP@K Scores of PRIME By Student Groups

K	Group	2019		2020	
		E2	E3	E2	E3
1	High	0.24	0.35	0.06	0.14
	Low	0.42	0.46	0.28	0.40
2	High	0.17	0.20	0.11	0.12
	Low	0.25	0.30	0.25	0.32

ineffective and such patterns are exhibited by a student, it may serve as an early warning sign for intervention.

Several other metrics were explored to determine the system's capability to recommend relevant items based on the actual performance of the students. Although several K values were explored, the framework only encompasses the top 1 or 2 items. Thus, the remaining elements on the list become irrelevant. The degree of agreement between the actual items and the recommended items was evaluated. This computation of the degree of agreement was adapted from Huang et al. (2020). The degree of agreement was normalized and illustrated in Figure 34, center. As can be seen from the results, it is apparent that another method of determining relevance is required based on the relatively low normalized degree of agreement score. Thus, relevance may extend beyond simply focusing on students' actual scores. However, as consistently has been seen in the evaluation so far, PRIME performed better for low-performing students. It is these students who are intended to benefit most from the recommendations. Another metric explored was the average distance. It basically computes the sum of displacements between two sequences. Unfortunately, the results as illustrated in Figure 34, center, are inconclusive. However, this suggests the need

for formalizing a metric that could truly measure the quality of an ordered list of recommended learning resources for students.

5.6.4 Question Relevance Rating of PRIME Aligns With Subject Matter Experts

As the actual questions used for forecasting a student's performance has not yet been released, it was necessary to identify proxies from the question library (refer to Section 5.4.3.3 for details). To evaluate the approach, the most viable candidates for each of the 158 questions were identified (i.e., those with the lowest relevance score). In general, the relevance score of a candidate question was 0.07 ($SD = 0.12$), where a value close to zero is preferred. In other words, there typically exists a relevant question in the library that can serve as a proxy. Upon closer examination of these candidates, it can be found that in some cases, multiple candidates are identified due to a tie in relevance scores. Therefore, variation can be incorporated into the process of making recommendations. As a matter of fact, on average, 0.08 ($SD = 0.07$) of the library's materials were considered viable candidates for a given item. This further suggests the possibility of variation. In this situation, it is up to the system to decide which ties to recommend to the student. A closer examination of the effects of randomization on the system can be conducted in future research. For the present study, only the first item on the list will be returned.

Due to the lack of ground truth regarding whether such a question is indeed relevant to the other questions, it was necessary to rely on experts to perform the validation. The following were determined: Each of the nine teachers independently rated the relevance of the top four results that will be returned by PRIME. Interestingly, on average, most of the results were assessed to be relevant as indicated in Figure 35.

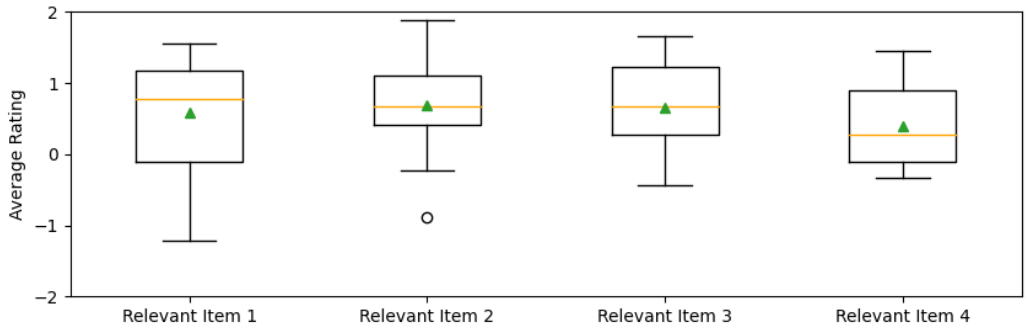


Figure 35. Average Rating of Subject Matter Experts on Top Four Relevant Questions Provided by PRIME

A Kruskal–Wallis test was performed to determine whether there was a significant difference between ratings of the relevant items. The result suggests that there was no significant difference ($p = 0.39$) in terms of the rating provided by the teacher. It is important to note that when this is deployed on WebPGA, only one item from a list of relevant items will be chosen (e.g., Relevant Item 1). However, it was unclear what relevance truly meant for experts. In the same survey, they provided their approach or basis on how they identified whether a question was relevant or not.

Teachers were also asked about their thought processes on how to assess relevance. Some participants found the tasks challenging and at times confusing. One participant indicated that the topics they believed were associated with the question were one way. Another participant indicated that the prerequisite was another way to determine relevance. If the output was a prerequisite or topic of the input, then it was deemed relevant as well because you cannot really proceed further if not. Interestingly, this reinforces the idea of the utility function of PRIME as discussed in Section 5.4.3.3. Recall that it is composed of three parts. First, it focuses on the similarity of distribution. Followed by the temporal aspect and the appropriateness of the question.

The third one deals with the prerequisite aspect of the case. Based on this information from a participant, it is encouraging how these align with PRIME.

5.6.5 General Discussion

It appears that the results of the objective evaluations conducted earlier point to the potential of PRIME. It is worth noting that two different perspectives can be considered when analyzing the results. Taking a broader view, the results indicate that it was possible to identify tasks that are appropriate for the needs of a student based on the candidate profiles identified for a student. It is possible to determine the difficulty level by evaluating the underestimation or overestimation of the overall performance λ, or in other words, the performance difference $\Delta\lambda$ of the student and the candidate. Meanwhile, taken from a narrow perspective (i.e., the goal of the individual component), it appears that PRIME requires further improvement to provide a more accurate prediction of an individual's overall performance. Understanding these contradictory scenarios could assist in clarifying PRIME's role in the future. Findings related to the ability to provide relevant items refer to an examination of the ability to do so in the absence of ground truth that can only be derived from a human perspective. As a result, it is necessary to investigate what considerations are pertinent to consider when determining whether an educational item is relevant to a student. As a final point, the same framework can also be applied to a similar problem. A common problem teachers face is the need to conduct a pilot test to help improve the quality of a new test. Using the same infrastructure, teachers can use the library to assist them in conducting a pilot test to assist them in improving the quality of the test.

5.7 Conclusion

In this chapter, the PRIME framework was presented as a means of addressing the limitations of an earlier work as well as the limitations of existing knowledge-tracing methods. A principled and practical approach was developed to model student mastery growth using performance data collected from semester-long classes. Particularly, scores on test items that are complex and may require partial credit were taken into account and used to obtain parameters that represent a student's growth rate in learning domain topics. Using the same model and a neighborhood-based forecasting approach, it was possible to provide a more detailed prediction of performance instead of a binary indicator of success, particularly by leveraging parameters from an identified candidate in the model library. Additionally, these predictions may help recommender systems provide relevant resources to students as they prepare for examinations. In this case, a worked example for students to evaluate. The workflow was intentionally designed to facilitate real-time recommendations. As new data is received, the framework can make updated recommendations, which simplifies the process without requiring additional effort from the teacher or additional offline training.

Every component of the framework was evaluated in order to gain a better understanding of its behavior and to assess its accuracy. As the results indicate, the framework holds particular promise for encapsulating student performance data into a unified growth model. In turn, these models can be easily instantiated and used by a new cohort of students to forecast their own performance. A closer examination of the framework's performance revealed that it was more effective with low-performing students, albeit marginally. Therefore, it achieved the purpose for which it was initially

intended. Certain findings, however, raise some concerns, including the various factors affecting the accuracy of the predictions. Moreover, it challenged some assumptions about what constitutes relevancy for students that had been adopted in earlier chapters of this dissertation. The framework should therefore be tested in its entirety in the future after it has been put together to determine whether it contributes to student learning.

The content of this chapter provides a framework for understanding how WebPGA can analyze existing performance data to intelligently provide students with personalized learning resources as they prepare for a test. Nonetheless, some of its limitations should be considered to guide future research. First, item difficulty was not considered in this study either in estimating mastery or during score prediction as it might have been in item response theory. It is posited that additional information about the items would result in improvement. Second, it was assumed that a well-defined syllabus would be followed over time, but it is worthwhile to consider how the framework will adapt to any major deviations. Third, for simplicity, certain processes were performed manually, such as relabeling KCs and matching them with topics. Several studies have already explored the possibility of automating some of these processes (e.g., ExamParser; Hsiao & Awasthi, 2015) that could be applied to certain steps of the framework. To minimize the possibility of introducing confounding factors, automation was minimized during the development. Fourth, the dataset examined only one aspect of the student. Additionally, students were assumed to have no prior knowledge at the beginning of the class. To provide a holistic picture of the student's mastery, many other systems could be included. This can be incorporated into the process, for example, if a standardized data collection method is used (e.g., xAPI; Paredes et al., 2020). Finally, it is essential to emphasize the importance of privacy.

Considering how this framework deals with sensitive student performance data, future research can examine how to possibly reconstruct student answers. It will ensure that the responses students receive are synthetic but based on real answers. Exploring this avenue would be a worthwhile endeavor.

Chapter 6

SUMMARY

Throughout this dissertation, several studies were designed and conducted on WebPGA to leverage learning analytics on educational assessment data. In addition to bridging the gap between the physical and digital worlds of educational assessment, WebPGA paved the way for empirical research into how students review their graded summative assessments. The various endeavors to better understand the students were hoped to facilitate the development of informed interventions so that students can become more aware of their misconceptions. In this work, assessment data was viewed as a source of human experience from which it was possible to gain knowledge. Moreover, it was demonstrated that experience could come either from oneself or from others.

In Chapter 2, it was hoped that students would reflect on their own experiences and learn from their mistakes by reviewing their own graded tests. Since this dissertation is focused on a specific learning environment and scenario of interest, no prior data was readily available. To devise interventions to assist students in becoming better learners, several factors need to be understood first. As a result, **RQ A.1**, **RQ A.2**, and **RQ A.3** were posed with the goal of determining whether behavioral differences exist among students. This revealed that better students tend to review their previous performance and learn from their mistakes, while the other group tended to do the opposite. This information was utilized to improve the system to provide guidance to low-performing students so that their misconceptions could be more clearly highlighted. Consequently, **RQ A.4** was posed to determine whether students would benefit from

tailored recommendations based on their performance. Guidance of this type was found to be beneficial to students, particularly low-performers. The belief that importance (or relevance as in Chapter 5) should be operationalized in this manner emerged as a result. By simply pointing out the areas where students made the most mistakes, it is assumed that they will be able to maximize their learning as they strive to close this gap.

In Chapter 3, another approach was used to understand students' behavior. Transitions or sequences were specifically considered, which had been omitted in the previous analysis. Following this, **RQ B.1** and **RQ B.2** were posed to further investigate the best practices that could be identified so that low-performing students could see and emulate these and improve. This chapter echoes findings previously reported but provides a more detailed description of how students distributed or attended their graded tests throughout the semester. Once again, it was evident how low-performing students do not recognize their misconceptions as they fail to make an effort to uncover them and learn from them. Specifically, these students rely on superficial, high-level feedback such as overall scores and do not take advantage of the feedback they are given. Additionally, the same students missed out on various other learning resources that were not required but provided additional knowledge.

In Chapter 4, an innovative approach to reviewing was presented. The students were asked to evaluate incorrect answers to test items of a hypothetical student. As a matter of fact, these test items are similar to items on the test they will soon take. It is therefore somewhat similar to reviewing, but with a twist. In addition, it is intended to illustrate the process of learning from the experiences of others. **RQ C.1** was posed in an attempt to determine whether students would benefit from the learning experience. The results of the study, however, were inconclusive. As a result, **RQ C.2**,

RQ C.3, and **RQ C.4** were posed to determine whether students attempted to make sense of the experiences of others, specifically whether they sought feedback to confirm their understanding. It was determined that students benefited from validating their understanding based on their improved performance on their midterms. Additionally, the same students improved their calibration or ability to assign the correct scores.

Finally, Chapter 5 takes a different approach. Incorporating the lessons learned in earlier chapters led to the development of a framework that addressed both the limitations of the proposed learning activity and those of existing knowledge-tracing techniques. This framework hopes to address several things based on **RQ D.1**, **RQ D.2**, and **RQ D.3**. First, encapsulate student performance data in a simple manner that allows for growth in mastery levels to be estimated. Second, make predictions that extend beyond knowing whether the student will succeed in answering a question. Third, identify relevant items for each student to facilitate the process of learning from others. All of this is for the purpose of streamlining and making intelligent the proposed learning activity. While the results of the evaluation appear promising, more questions remain. Identifying a relevant resource for a student and understanding how the characteristics of the student affect that resource are two key questions that must be addressed.

Overall, the system has provided an infrastructure for blended learning environments that streamline the various processes of learning analytics to enhance the student's learning experience. Due to its modular design, future improvements can be easily incorporated and tested, increasing the system's usefulness.

6.1 Educational Implications

This dissertation supports the shift towards the notion of *assessment as learning*. Assessments are already being viewed differently and students are expected to take an active role in understanding their assessment results. It is critical that students have the opportunity to reflect upon their experiences as part of developing the fundamental skills that enable them to become independent learners. The findings regarding the behaviors of successful students can serve as a reminder for teachers to emphasize the importance of reflecting on one's own performance to their students. In addition, this dissertation has highlighted the approach of learning from erroneous examples and how it can potentially be employed in instruction, particularly in the domain of computer programming. By also showcasing mistakes, people can widen their understanding of the domain and avoid repeating common mistakes. In the era of rapid technological advancements, teachers must continue to explore ways of incorporating innovation into their teaching strategies to meet the ever-changing needs of modern students. Developing technologies will be facilitated by combining artificial intelligence and machine learning. The success of these depends, however, on their ability to harness both human and artificial intelligence.

The current system can be utilized by teachers to assist them in making sense of the numerous questions that they have created over the years. The collected data may be used to improve the items (e.g., to adjust the points or question type). As a matter of fact, although the PRIME framework is intended to ultimately facilitate and guide students, its original purpose was to provide teachers with a data-driven approach for improving the quality of the tests that they construct (Paredes & Hsiao, 2022a). Students from the same library can be used to simulate a classroom so the teacher

can see the likely outcome of a newly constructed test. In this way, teachers will be able to revise their tests as necessary. It is possible that these revision behaviors of teachers could contribute to the identification of best practices that can be shared with fellow teachers early in their careers. Additionally, an understanding of these behaviors could contribute to closing the loop on automated educational assessment.

A recurring theme in this dissertation is the exploration of potential uses of performance data and assessment that some teachers may be reluctant to consider. The days when it was impossible to obtain a copy of an old test are long gone. For example, if a course is common, such as an introductory course, a determined individual may be able to locate test banks that are similar to those that will be used by the teacher. As a compromise, it is believed that recognizing such problems and exploiting them to prepare students would be a good approach. To the best of my knowledge, the context of utilizing existing student performance data to serve as a potential exercise for future students has not been investigated. By removing any personal information or reproducing past students' answers, future students can benefit from these vicarious experiences. Further research is necessary to fully understand how it impacts students' mastery and their learning.

6.2 Limitations and Future Work

All of the explorations have been focused on computer programming. Whether or not these findings are also applicable to other domains, including those that are ill-defined, remains unclear. In addition, all the datasets were derived from digitized paper tests. Digital tests are increasingly being considered by more and more classes.

Therefore, it would be interesting to learn if the same findings hold for WebPGA's digital counterpart.

The digital artifacts (i.e., scanned paper) were not tapped due to their sensitive nature. However, it is believed that there may be something useful that can be learned from them. In fact, there is growing interest in exploring how to make sense of these markings. It could potentially pave the way for reconstructing a student's answer such that the actual is not presented elsewhere, thus addressing the privacy issue. In addition to student performance and behavior data, WebPGA collected behavioral data about teachers and graders as they graded student answers. It is possible to examine the coherence and consistency of graders' evaluations in the context of evaluating answers to the same question. While data have already been collected, this issue has not yet been investigated. Another aspect that has not yet been explored is the use of performance data in conjunction with behavioral data. The behavioral component could help shed light on determining what constitutes relevance for a specific individual. For example, if an item has already been reviewed by the student, it should not be recommended further. With data constantly arriving at the system, these ever-changing dimensions are likely to serve as implicit feedback to assist in improving performance. A deeper understanding of the behavioral aspects would also be beneficial in guiding how artificial students behave in a simulation environment. Lastly, there has been much research on integrating human aspects into educational systems, such as the use of pedagogical agents (Biswas et al., 2005; Matsuda et al., 2013; Schroeder et al., 2013). Integrating lessons learned from other systems would be a worthwhile endeavor.

6.3 Contributions

Among the contributions of this dissertation is the design and development of a research platform that facilitates the application of learning analytics to educational assessment systems. In the past six years, it has served over 6,000 students from two universities. The system helped streamline and support the grading process for over 40 classes (including Mathematics) handled by more than 15 faculty members, thereby eliminating the need for physically moving papers. It resulted in at least a 40% reduction in the time it takes students to receive feedback! Furthermore, it provided empirical evidence on how students review their graded answers, thus paving the way for the introduction of various interventions that could assist students in reviewing their results. Further, this study contributes to the literature on student modeling by developing the PRIME framework, a novel method for predicting student performance on complex questions that allow partial credit. A prototype of the framework itself, deployed on WebPGA, is the final contribution of this dissertation, as it provides a guideline for students to identify appropriate items to review. The findings of the evaluation can serve as baseline values for future developments.

REFERENCES

Aggarwal, C. C. (2016). *Recommender systems*. Springer.

Ambrose, S. A., Bridges, M. W., DiPietro, M., Lovett, M. C., & Norman, M. K. (2010). *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons.

An, S., Kim, J., Kim, M., & Park, J. (2022). No task left behind: Multi-task learning of knowledge tracing and option tracing for better student assessment. *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, 4424–4431.

Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, *70*(2), 181–214.

Bandura, A. (1977). *Social learning theory*. Prentice Hall.

Bandura, A. (1997). *Self-efficacy: The exercise of control*. W.H. Freeman and Company.

Barnes, T. (2011). Novel derivation and application of skill matrices: The q-matrix method. *Handbook on educational data mining* (pp. 159–172).

Bauman, K., & Tuzhilin, A. (2018). Recommending remedial learning materials to students by filling their knowledge gaps. *MIS Quarterly*, *42*(1), 313–332.

Biswas, G., Leelawong, K., Schwartz, D., Vye, N., & The Teachable Agents Group at Vanderbilt. (2005). Learning by teaching: A new agent paradigm for educational software. *Applied Artificial Intelligence*, *19*(3-4), 363–392.

Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J. (2013). Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. *Learning and Instruction*, *25*, 24–34.

Boud, D., Keogh, R., & Walker, D. (1985). What is reflection in learning? *Reflection: Turning experience into learning* (pp. 7–17). Routledge.

Brahim, G. B. (2022). Predicting student performance from online engagement activities using novel statistical features. *Arabian Journal for Science and Engineering*.

Brewer, P. W. (1996). *Methods for concept mapping in computer based education* (Master's thesis). North Carolina State University.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-adapted Interaction*, *6*(2-3), 87–129.

Brusilovsky, P. (1998). Methods and techniques of adaptive hypermedia. *Adaptive hypertext and hypermedia* (pp. 1–43).

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, *11*, 87–110.

Brusilovsky, P. (2003). Developing adaptive educational hypermedia systems: From design models to authoring tools. *Authoring tools for advanced technology learning environments* (pp. 377–409). Springer.

Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis–a general method for cognitive model evaluation and improvement. *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, 164–175.

Chen, J., & de la Torre, J. (2018). Introducing the general polytomous diagnosis modeling framework. *Frontiers in Psychology*, *9*, Article 1474.

Chen, Y., Liu, Q., Huang, Z., Wu, L., Chen, E., Wu, R., Su, Y., & Hu, G. (2017). Tracking knowledge proficiency of students with educational priors. *Proceedings of the 26th ACM International Conference on Information and Knowledge Management*, 989–998.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*, 145–182.

Chi, M. T. H., & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, *49*(4), 219–243.

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

Cohen, P. R., & Howe, A. E. (1988). How evaluation guides AI research: The message still counts more than the medium. *AI Magazine*, *9*(4), 35–43.

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

Craig, S. D., Chi, M. T. H., & VanLehn, K. (2009). Improving classroom learning by collaboratively observing human tutoring videos while problem solving. *Journal of Educational Psychology*, *101*(4), 779–789.

Daly, C., & Waldron, J. (2004). Assessing the assessment of programming ability. *ACM SIGCSE Bulletin*, *36*(1), 210–213.

de la Torre, J. (2011). The generalized DINA model framework. *Psychometrika*, *76*, 179–199.

Doignon, J.-P., & Falmagne, J.-C. (1999). *Knowledge spaces*. Springer.

Dunning, D. (2011). The Dunning–Kruger effect: On being ignorant of one's own ignorance. *Advances in experimental social psychology* (pp. 247–296).

Earl, L. M. (2013). *Assessment as learning: Using classroom assessment to maximize student learning*. Corwin Press.

Falmagne, J.-C., Koppen, M., Villano, M., Doignon, J.-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, *97*(2), 201–224.

Folsom-Kovarik, J., Chen, D.-W., Mostafavi, B., & Freed, M. (2019). Personalization. In J. J. Walcutt & S. Schatz (Eds.), *Modernizing learning: Building the future learning ecosystem* (pp. 181–198). Government Publishing Office.

Gervet, T., Koedinger, K., Schneider, J., & Mitchell, T. (2020). When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, *12*(3), 31–54.

Ghosh, A., Raspat, J., & Lan, A. (2021). Option tracing: Beyond correctness analysis in knowledge tracing. *Proceedings of the 22nd International Conference on Artificial Intelligence in Education*, 137–149.

Gong, Y., Beck, J. E., & Heffernan, N. T. (2010). Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. *Proceedings of the International Conference on Intelligent Tutoring Systems*, 35–44.

Große, C. S., & Renkl, A. (2007). Finding and fixing errors in worked examples: Can this foster learning outcomes? *Learning and Instruction*, *17*(6), 612–634.

Guo, L., Bao, Y., Wang, Z., & Bian, Y. (2014). Cognitive diagnostic assessment with different weight for attribute: Based on the DINA model. *Psychological Reports*, *114*(3), 802–822.

Hanna, G. S., & Dettmer, P. (2004). *Assessment for effective teaching: Using context-adaptive planning*. Allyn & Bacon.

Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, *77*(1), 81–112.

Hellas, A., Ihantola, P., Petersen, A., Ajanovski, V. V., Gutica, M., Hynninen, T., Knutas, A., Leinonen, J., Messom, C., & Liao, S. N. (2018). Predicting academic performance: A systematic literature review. *Companion Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 175–199.

Horstmann, C. S. (2013). *Java for everyone: Late objects*. John Wiley & Sons.

Hosseini, R., & Brusilovsky, P. (2017). A study of concept-based similarity approaches for recommending program examples. *New Review of Hypermedia and Multimedia*, *23*(3), 161–188.

Hsiao, I.-H., & Awasthi, P. (2015). Topic facet modeling: Semantic visual analytics for online discussion forums. *Proceedings of the 5th International Conference on Learning Analytics and Knowledge*, 231–235.

Huang, Z., Liu, Q., Chen, Y., Wu, L., Xiao, K., Chen, E., Ma, H., & Hu, G. (2020). Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems*, *38*(2), Article 19.

Isotani, S., Adams, D., Mayer, R. E., Durkin, K., Rittle-Johnson, B., & McLaren, B. M. (2011). Can erroneous examples help middle-school students learn decimals? *Proceedings of the 6th European Conference on Technology Enhanced Learning*, 181–195.

Kalyuga, S. (2007). Expertise reversal effect and its implications for learner-tailored instruction. *Educational Psychology Review*, *19*(4), 509–539.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Educational Psychology*, *93*(3), 579–588.

Keith, G. (2021). Transforming scores into probability. https://www.cantorsparadise.com/transforming-scores-into-probability-fb4d4be7deab

Khan, A., & Ghosh, S. K. (2021). Student performance analysis and prediction in classroom learning: A review of educational data mining studies. *Education and Information technologies*, *26*(1), 205–240.

Knight, J. K., Weaver, D. C., Peffer, M. E., & Hazlett, Z. S. (2022). Relationships between prediction accuracy, metacognitive reflection, and performance in introductory genetics students. *CBE—Life Sciences Education*, *21*(3), Article 45.

Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, *36*(5), 757–798.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37.

Linn, M. C., & Dalbey, J. (1985). Cognitive consequences of programming instruction: Instruction, access, and ability. *Educational Psychologist*, *20*(4), 191–206.

Liu, N., Wang, Z., Baraniuk, R. G., & Lan, A. (2022). Open-ended knowledge tracing. https://doi.org/10.48550/arXiv.2203.03716

Liu, Q., Shen, S., Huang, Z., Chen, E., & Zheng, Y. (2021). A survey of knowledge tracing. https://doi.org/10.48550/arXiv.2105.15106

Matsuda, N., Yarzebinski, E., Keiser, V., Raizada, R., Cohen, W. W., Stylianides, G. J., & Koedinger, K. R. (2013). Cognitive anatomy of tutor learning: Lessons learned with SimStudent. *Journal of Educational Psychology*, *105*(4), 1152–1163.

McConlogue, T. (2015). Making judgements: Investigating the process of composing and receiving peer feedback. *Studies in Higher Education*, *40*(9), 1495–1506.

McLaren, B. M., & Isotani, S. (2011). When is it best to learn with all worked examples? *Proceedings of the 15th International Conference on Artificial Intelligence in Education*, 222–229.

Mehrens, W. A., & Lehmann, J., Irvin. (1991). *Measurement and evaluation in education and psychology*. Holt, Rinehart; Winston, Inc.

Melis, E. (2005). Design of erroneous examples for ActiveMath. *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 451–458.

Nicol, D. (2021). The power of internal feedback: Exploiting natural comparison processes. *Assessment & Evaluation in Higher Education*, *46*(5), 756–778.

Nicol, D., Thomson, A., & Breslin, C. (2014). Rethinking feedback practices in higher education: A peer review perspective. *Assessment & Evaluation in Higher Education*, *39*(1), 102–122.

Norman, D. A., & Draper, S. W. (1986). *User centered system design: New perspectives on human-computer interaction.*

Ohlsson, S. (1996). Learning from error and the design of task environments. *International Journal of Educational Research*, *25*(5), 419–448.

Paredes, Y. V., Azcona, D., Hsiao, I.-H., & Smeaton, A. F. (2018). Predictive modelling of student reviewing behaviors in an introductory programming course. *Proceedings of the 1st Educational Data Mining in Computer Science Education Workshop.*

Paredes, Y. V., & Hsiao, I.-H. (2021a). Can students learn from grading erroneous computer programs? *Proceedings of the 2021 International Conference on Advanced Learning Technologies*, 211–215.

Paredes, Y. V., & Hsiao, I.-H. (2021b). WebPGA: An educational technology that supports learning by reviewing paper-based programming assessments. *Information*, *12*(11), Article 450.

Paredes, Y. V., & Hsiao, I.-H. (2022a). Combining data and human intelligence through predictive visual analytics to improve educational assessments. *Proceedings of the 30th International Conference on Computers in Education Volume I*, 317–319.

Paredes, Y. V., & Hsiao, I.-H. (2022b). Modeling students' ability to recognize and review graded answers that require immediate attention. *Proceedings of the 30th International Conference on Computers in Education Volume II*, 85–90.

Paredes, Y. V., Siegle, R. F., Hsiao, I.-H., & Craig, S. D. (2020). Educational data mining and learning analytics for improving online learning environments. *Proceedings of the 2020 Human Factors and Ergonomics Society 64th International Annual Meeting*, 500–504.

Park, J. Y., Dedja, K., Pliakos, K., Kim, J., Joo, S., Cornillie, F., Vens, C., & Van den Noortgate, W. (2022). Comparing the prediction performance of item response theory and machine learning methods on item responses for educational assessments. *Behavior Research Methods*. https://doi.org/10.3758/s13428-022-01910-8

Pavlik, P. I., Jr, Cen, H., & Koedinger, K. R. (2009). Performance factors analysis–a new alternative to knowledge tracing. *Proceedings of the 14th International Conference on Artificial Intelligence in Education*.

Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, *27*, 313–350.

Pelánek, R. (2020). Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies*, *13*(2), 354–366.

Piaget, J., & Inhelder, B. (1969). *The psychology of the child*. Basic Books.

Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein, J. (2015). Deep knowledge tracing. *Advances in Neural Information Processing Systems*, *28*.

Pillay, N. (2003). Developing intelligent programming tutors for novice programmers. *ACM SIGCSE Bulletin*, *35*(2), 78–82.

Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction*, *12*(5), 529–556.

Renkl, A. (2014). The worked examples principle in multimedia learning. *The Cambridge handbook of multimedia learning* (pp. 391–412). Cambridge University Press.

Robins, A. (2010). Learning edge momentum: A new account of outcomes in CS1. *Computer Science Education*, *20*(1), 37–71.

Robins, A. (2019). Novice programmers and introductory programming. In S. Fincher & A. Robins (Eds.), *The Cambridge handbook of computing education research* (pp. 327–376). Cambridge University Press.

Romero, C., & Ventura, S. (2020). Educational data mining and learning analytics: An updated survey. *WIREs: Data Mining Knowledge Discovery*, *10*(3), Article e1355.

Sadler, D. R. (2010). Beyond feedback: Developing student capability in complex appraisal. *Assessment & Evaluation in Higher Education*, *35*(5), 535–550.

Schroeder, N. L., Adesope, O. O., & Gilbert, R. B. (2013). How effective are pedagogical agents for learning? a meta-analytic review. *Journal of Educational Computing Research*, *49*(1), 1–39.

Sharp, L. A. (2012). Stealth learning: Unexpected learning opportunities through games. *Journal of Instructional Research*, *1*, 42–48.

Sheard, J., Carbone, A., D'Souza, D., & Hamilton, M. (2013). Assessment of programming: Pedagogical foundations of exams. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 141–146.

Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. In P. J. Durlach & A. M. Lesgold (Eds.), *Adaptive technologies for training and education* (pp. 1–35).

Sosnovsky, S., & Brusilovsky, P. (2015). Evaluation of topic-based adaptation and student modeling in QuizGuide. *User Modeling and User-Adapted Interaction*, *25*, 371–424.

Sweller, J. (2011). Cognitive load theory. *Psychology of learning and motivation* (pp. 37–76).

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, *2*(1), 59–89.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 345–354.

Tew, A. E., & Guzdial, M. (2010). Developing a validated assessment of fundamental CS1 concepts. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 97–101.

Thai-Nghe, N., Drumond, L., Horváth, T., & Schmidt-Thieme, L. (2011). Multi-relational factorization models for predicting student performance. *Proceedings of the KDD 2011 Workshop on Knowledge Discovery in Educational Data.*

Thai-Nghe, N., Drumond, L., Horváth, T., Krohn-Grimberghe, A., Nanopoulos, A., & Schmidt-Thieme, L. (2012). Factorization techniques for predicting student

performance. *Educational recommender systems and technologies: Practices and challenges* (pp. 129–153). IGI Global.

Thai-Nghe, N., & Schmidt-Thieme, L. (2015). Multi-relational factorization models for student modeling in intelligent tutoring systems. *Proceedings of the 7th International Conference on Knowledge and Systems Engineering*, 61–66.

Thaker, K., Zhang, L., He, D., & Brusilovsky, P. (2020). Recommending remedial readings using student knowledge state. *Proceedings of the 13th International Conference on Educational Data Mining*, 233–244.

Tsovaltzi, D., Melis, E., McLaren, B. M., Meyer, A. K., Dietrich, M., & Goguadze, G. (2010). Learning from erroneous examples: When and how do students benefit from them? *Proceedings of the 5th European Conference on Technology Enhanced Learning*, 357–373.

van de Watering, G., & van der Rijt, J. (2006). Teachers' and students' perceptions of assessments: A review and a study into the ability and accuracy of estimating the difficulty levels of assessment items. *Educational Research Review*, *1*(2), 133–147.

VanLehn, K. (1996). Cognitive skill acquisition. *Annual Review of Psychology*, *47*, 513–539.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, *16*(3), 227–265.

VanLehn, K., Niu, Z., Siler, S., & Gertner, A. S. (1998). Student modeling from conventional test data: A bayesian approach without priors. *Proceedings of the International Conference on Intelligent Tutoring Systems*, 434–443.

Verhulst, P. F. (1845). Resherches mathematiques sur la loi d'accroissement de la population. *Nouveaux Memoires de l'academie Royale des Sciences*, *18*, 1–41.

Vygotsky, L. S. (1978). *Mind in society: Development of higher psychological processes*. Harvard University Press.

Wang, S., He, F., & Andersen, E. (2017). A unified framework for knowledge assessment and progression analysis and design. *Proceedings of the 2017 Conference on Human Factors in Computing Systems*, 937–948.

Wigfield, A., & Eccles, J. S. (2000). Expectancy–value theory of achievement motivation. *Contemporary Educational Psychology*, *25*(1), 68–81.

Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *ACM SIGCSE Bulletin*, *28*(3), 17–22.

Zapata-Rivera, D., Lehman, B., & Sparks, J. R. (2020). Learner modeling in the context of caring assessments. *Proceedings of the International Conference on Human-Computer Interaction*, 422–431.

Zimmerman, B. J. (1998). Developing self-fulfilling cycles of academic regulation: An analysis of exemplary instructional models. *Self-regulated learning: From teaching to self-reflective practice* (pp. 1–19). Guilford Publications.

APPENDIX A

KOLMOGOROV-SMIRNOV TEST RESULTS

E1	S1	S2	S3	Cum% S1	Cum% S2	Diff	Cum% S1	Cum% S3	Diff	Cum% S2	Cum% S3	Diff
01 - intro	0.06	0.02	0.02	0.06	0.02	0.04	0.06	0.02	0.04	0.02	0.02	-
02 - fundamental data type	0.56	0.51	0.59	0.62	0.53	0.09	0.62	0.61	0.01	0.51	0.59	0.08
03 - decision	0.20	0.29	0.16	0.82	0.82	-	0.82	0.77	0.05	0.29	0.16	0.13
04 - loop	0.18	0.18	0.23	1.00	1.00	-	1.00	1.00	-	0.18	0.23	0.05
05 - methods	-	-	-	1.00	1.00	-	1.00	1.00	-	-	-	-
06 - arrays and array list	-	-	-	1.00	1.00	-	1.00	1.00	-	-	-	-
08 - objects and class	-	-	-	1.00	1.00	-	1.00	1.00	-	-	-	-
09 - inheritance and interface	-	-	-	1.00	1.00	-	1.00	1.00	-	-	-	-
					D-stat	0.09		D-stat	0.05		D-stat	0.13
					D-crit	0.19		D-crit	0.19		D-crit	0.19
						SAME			SAME			SAME

E2	S1	S2	S3	Cum% S1	Cum% S2	Diff	Cum% S1	Cum% S3	Diff	Cum% S2	Cum% S3	Diff
01 - intro	0.03	0.08	0.06	0.03	0.08	0.05	0.03	0.06	0.03	0.08	0.06	0.02
02 - fundamental data type	0.08	0.12	0.19	0.11	0.20	0.09	0.11	0.25	0.14	0.12	0.19	0.07
03 - decision	0.11	0.04	0.07	0.22	0.25	0.03	0.22	0.31	0.09	0.04	0.07	0.02
04 - loop	0.07	0.04	0.07	0.29	0.29	0.01	0.29	0.38	0.10	0.04	0.07	0.02
05 - methods	0.22	0.13	0.08	0.51	0.42	0.09	0.51	0.46	0.04	0.13	0.08	0.05
06 - arrays and array list	0.22	0.19	0.17	0.72	0.61	0.11	0.72	0.63	0.09	0.19	0.17	0.02
08 - objects and class	0.28	0.39	0.37	1.00	1.00	-	1.00	1.00	-	0.39	0.37	0.02
09 - inheritance and interface	-	-	-	1.00	1.00	-	1.00	1.00	-	-	-	-
					D-stat	0.11		D-stat	0.14		D-stat	0.07
					D-crit	0.19		D-crit	0.19		D-crit	0.19
						SAME			SAME			SAME

E3	S1	S2	S3	Cum% S1	Cum% S2	Diff	Cum% S1	Cum% S3	Diff	Cum% S2	Cum% S3	Diff
01 - intro	0.02	0.07	0.05	0.02	0.07	0.05	0.02	0.05	0.03	0.07	0.05	0.02
02 - fundamental data type	0.10	0.06	0.15	0.11	0.13	0.02	0.11	0.20	0.09	0.06	0.15	0.09
03 - decision	0.08	0.02	0.04	0.19	0.15	0.04	0.19	0.24	0.05	0.02	0.04	0.02
04 - loop	0.04	0.03	0.02	0.23	0.18	0.05	0.23	0.26	0.03	0.03	0.02	0.01
05 - methods	0.10	0.03	0.07	0.32	0.21	0.11	0.32	0.33	0.01	0.03	0.07	0.04
06 - arrays and array list	0.07	0.06	0.02	0.39	0.27	0.12	0.39	0.35	0.04	0.06	0.02	0.04
08 - objects and class	0.33	0.38	0.25	0.72	0.65	0.07	0.72	0.60	0.12	0.38	0.25	0.13
09 - inheritance and interface	0.28	0.35	0.40	1.00	1.00	0.00	1.00	1.00	0.00	0.35	0.40	0.05
					D-stat	0.12		D-stat	0.12		D-stat	0.13
					D-crit	0.19		D-crit	0.19		D-crit	0.19
						SAME			SAME			SAME

alpha	0.05

APPENDIX B

SIMULATION CODES USED FOR THE EXPERIMENT

```python
import random
import numpy as np

def get_adjusted_band_width(max_score, adj):
    # this widens the success directly prop to adj
    # adjusted for success
    choices = max_score + 1
    adj_r = 1-adj # had to flip if we want to be directly prop

    # this is just weighted means incremental
    a = np.arange(choices)+1
    b = np.flip( np.arange(choices) )
    c = b * adj_r
    d = a+c

    # recomputed prob
    pr_orig = np.array( [1/choices]*choices )
    pr_mult = pr_orig * d
    pr_chance = pr_mult / (d.sum()/choices)

    # get new cut offs
    return pr_chance.cumsum()


def get_adjusted_score(max_score, adj):
    # current limitation is that we truncate floats
    max_score = int( max_score )

    # get the adjusted bands
    cut_offs = get_adjusted_band_width(max_score, adj)

    # flip a coin
    coin = random.uniform(0, 1)

    # determine the position, if tie, position it to the right
    return cut_offs.searchsorted(coin, side='right')
```

Listing B.1. Code defined to identify partial credit scores of simulated students given proficiency

```
def get_prime_predicted_at_k(row, K):
    # rank and remove nans
    row = row.dropna().rank(method='min')
    tmp_rr = row[row <= K].sort_values().reset_index()
    prop_set = []

    actual_elem_counter = 0
    for ir, dr in tmp_rr.groupby(tmp_rr.columns[-1]): # sorted guaranteed per documentation
        needed_elems = K - actual_elem_counter
        # how many elems here
        potential_elems = dr['question_number'].tolist()
        potential_elems_count = len( potential_elems )

        # if still below capacity
        if potential_elems_count <= needed_elems:
            # add all
            prop_set.append( set( potential_elems ) )
        else:
            # just shuffle and choose the remaining
            prop_set.append( set( random.sample(potential_elems, k=needed_elems) ) )

    return prop_set

def get_actual_at_k(row, K):
    # rank and remove nans
    row = row.dropna().rank(method='min')
    tmp_rr = row[row <= K].sort_values().reset_index()
    prop_set = []
    for ir, dr in tmp_rr.groupby(tmp_rr.columns[-1]): # sorted guaranteed per documentation
        potential_elems = dr['question_number'].tolist()
        prop_set.append( set( potential_elems ) )
    return prop_set

def get_baseline_predicted_at_k(row, K):
    # literal assume random with singleton only
    return [ {el} for el in random.sample( row.dropna().index.tolist(), k=K ) ]

def get_actual_relevant_questions(row, K):
    row = row.dropna().rank(method='min')

    return row[row <= K].index.to_numpy()
```

Listing B.2. Code defined to obtain the recommendation of both PRIME and the Baseline recommender

APPENDIX C

DATA COLLECTION SYSTEM USED FOR EXPERT EVALUATION OF QUESTION RELEVANCE

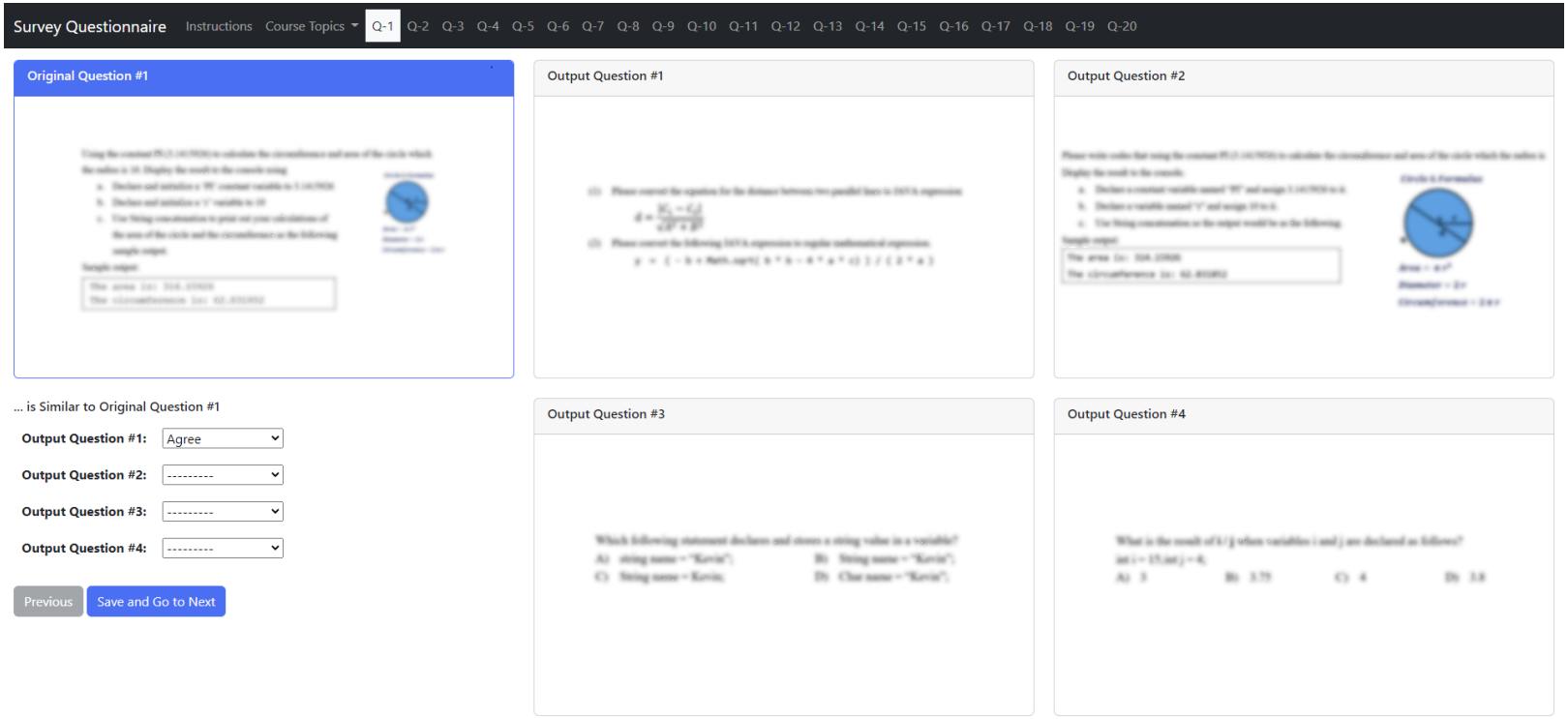


Figure 36. Sample Screenshot of System Used to Solicit Expert Rating on Question Relevance

Note: The tests questions in this figure were intentionally blurred in this document to preserve their integrity.

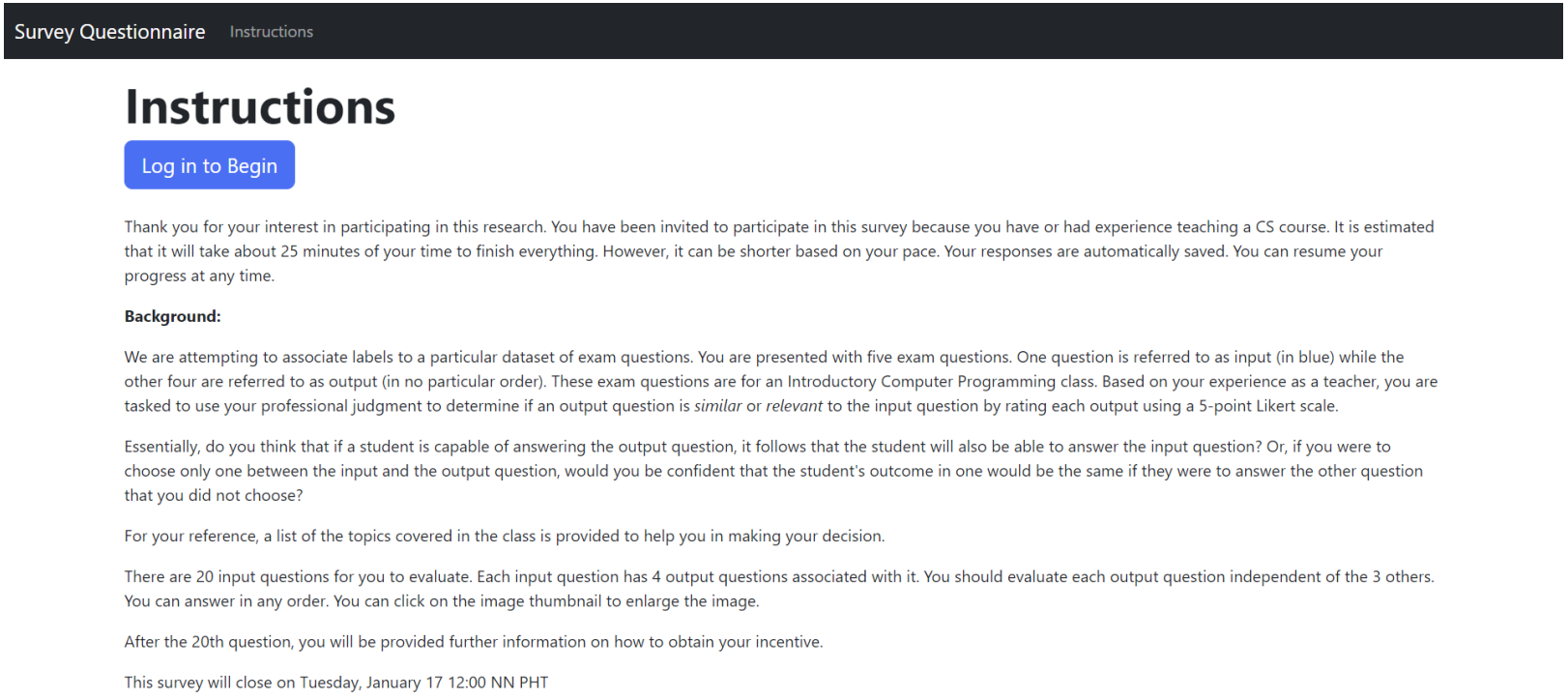


Figure 37. Experiment Instructions for Experts

APPENDIX D

WORKED EXAMPLE OF IDENTIFYING CRITICAL ITEM

Raw Points

	T1	T2	T3	Total
Q1	2.00	-	5.00	7.00
Q2	4.00	2.00	1.00	7.00
Q3	9.00	3.00	7.00	19.00
Total	15.00	5.00	13.00	**33.00**
Beta	0.70	0.60	0.40	

Normalized Q-Matrix

	T1	T2	T3	Total
Q1	0.29	-	0.71	1.00
Q2	0.57	0.29	0.14	1.00
Q3	0.47	0.16	0.37	1.00

Manually Computed

	T1	T2	T3	Total	Normalized
Q1	1.40	-	2.00	3.40	0.49
Q2	2.80	1.20	0.40	4.40	0.63
Q3	6.30	1.80	2.80	10.90	0.57

Directly Computed

	T1	T2	T3	Total
Q1	0.20	-	0.29	0.49
Q2	0.40	0.17	0.06	0.63
Q3	0.33	0.09	0.15	0.57

APPENDIX E

IRB APPROVAL

APPROVAL: EXPEDITED REVIEW

Scotty Craig
IAFSE-PS: Human Systems Engineering (HSE)
480/727-1006
Scotty.Craig@asu.edu

Dear Scotty Craig:

On 5/11/2022 the ASU IRB reviewed the following protocol:

Type of Review:	Initial Study
Title:	Development of a Web-based Educational Technology: Incorporating Artificial Intelligence in the Educational Assessment Process
Investigator:	Scotty Craig
IRB ID:	STUDY00015953
Category of review:	
Funding:	Name: ASU: Graduate and Professional Student Association (GPSA)
Grant Title:	
Grant ID:	
Documents Reviewed:	• Consent Form Student, Category: Consent Form; • Consent Form Teacher, Category: Consent Form; • IRB_Application_11-05-2022.docx, Category: IRB Protocol; • Recruitment_Methods_Email_11-05-2022.pdf, Category: Recruitment Materials; • Research Grant AppID388553.PDF, Category: Sponsor Attachment; • Supporting_Documents_11-05-2022.pdf, Category: Measures (Survey questions/Interview questions /interview guides/focus group questions);

The IRB approved the protocol from 5/11/2022 to 5/10/2027 inclusive. Three weeks before 5/10/2027 you are to submit a completed Continuing Review application and required attachments to request continuing approval or closure.

If continuing review approval is not granted before the expiration date of 5/10/2027 approval of this protocol expires on that date. When consent is appropriate, you must use final, watermarked versions available under the "Documents" tab in ERA-IRB.

In conducting this protocol you are required to follow the requirements listed in the INVESTIGATOR MANUAL (HRP-103).

REMINDER - - Effective January 12, 2022, in-person interactions with human subjects require adherence to all current policies for ASU faculty, staff, students and visitors. Up-to-date information regarding ASU's COVID-19 Management Strategy can be found here. IRB approval is related to the research activity involving human subjects, all other protocols related to COVID-19 management including face coverings, health checks, facility access, etc. are governed by current ASU policy.

Sincerely,

IRB Administrator

cc: Yancy Vance Paredes

APPENDIX F

PERMISSION FROM CO-AUTHOR OF PREVIOUSLY PUBLISHED ARTICLES

CERTIFICATION

There are chapters in this dissertation that have been previously published. I was the first author of all previous publications. The following publications were incorporated into this dissertation and adapted:

Chapter 2

> Paredes, Y. V., & Hsiao, I.-H. (2021b). WebPGA: An educational technology that supports learning by reviewing paper-based programming assessments. *Information*, *12*(11), Article 450.

Chapter 3

> Paredes, Y. V., & Hsiao, I.-H. (2022b). Modeling students' ability to recognize and review graded answers that require immediate attention. *Proceedings of the 30th International Conference on Computers in Education Volume II*, 85–90.

Chapter 4

> Paredes, Y. V., & Hsiao, I.-H. (2021a). Can students learn from grading erroneous computer programs? *Proceedings of the 2021 International Conference on Advanced Learning Technologies*, 211–215. **(best paper nominee)**

The purpose of this page is to certify that I have obtained the approval of my co-author for the inclusion of the above-mentioned publications in this dissertation.

Yancy Vance Paredes

January 6, 2023