Exploratory Process Data Analysis in the Mixed-format Assessment: Using Reservoir Computing and Topic Modeling

by

Jiawei Xiong

(Under the Direction of Allan S. Cohen)

Abstract

Computer-based assessments have become prevalent in many educational assessments. These assessments are usually in mixed formats. That is, they contain different item formats such as multiple-choice (MC) and constructed-response (CR) items. These items, regardless of the item formats, are developed to measure the examinee's skill or ability related to the construct of interest, e.g., problem-solving or critical thinking. Item response models are frequently used to calibrate the examinee's latent trait based on their response score patterns. One concern with the scores examinees receive is the scores alone may not completely convey sufficient information to help understand the targeted latent trait. For example, scores may not necessarily provide information about specific thinking or reasoning examinees used in their responses. In this study, we focus on extracting additional information from examinees' responses that may provide this kind of additional construct relevant information. In this regard, we explore the information contained in examinees' sequential actions as recorded in the log file of a computer-based

assessment. This information is referred to as response process data, or more simply process data. The process data generated by examinees in their responses to a computer-based assessment have been shown to be related to information in their responses given to both MC and CR items.

This dissertation consists of two studies. The first study examines a novel exploratory methodology for extracting process data from the log files of a computer-based assessment. This methodology is called reservoir computing and is implemented with an optimization algorithm. This first study is used for analyzing and extracting process data in the log file from an administration of MC items. This method will be studied for its use in extracting features of the response data with an eye to help interpret the latent information in the response processes associated with the measurement of the latent construct. The second study examines the use of a natural language method using a probabilistic topic model to extract the latent features in the textual responses to CR items. In this second study, the utility of the unsupervised and supervised topic models will be studied for the analysis of textual responses with an eye to extracting construct-relevant information from the process data that can be used to help interpret examinees' status on the latent construct. The combination of the two studies is intended to help provide a way for extracting and studying the combination of item response scores and item response process data to improve interpretations of examinees' latent proficiencies.

INDEX WORDS: Mixed-format assessment, Process data, Reservoir computing, Topic model, Latent variable modeling, Feature extraction

Exploratory Process Data Analysis in the Mixed-format Assessment: Using Reservoir Computing and Topic Modeling

by

Jiawei Xiong

B.E., Chongqing University, China, 2017

M.S. University of Georgia, 2020

A Dissertation Submitted to the Graduate Faculty of the

University of Georgia in Partial Fulfillment of the Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2022

Exploratory Process Data Analysis in the Mixed-format Assessment: Using Reservoir Computing and Topic Modeling

by

Jiawei Xiong

Major Professor: Allan S. Cohen

Committee: George Engelhard Jr.

Seock-Ho Kim

Shiyu Wang

Sheng Li

Electronic Version Approved:

Ron Walcott

Vice Provost for Graduate Education and Dean of the Graduate School

The University of Georgia

August 2022

Dedication

This is in memory of my paternal grandfather,

Yuanchun Xiong

and my maternal grandmother,

Fengjiao Ding

Your love spanned my childhood, adolescent, and my early 20s. I love you.

I would like to dedicate my work to my maternal gradfather, Jianguo Wu, my paternal gradmother, Wenqi Zhou, my parents, Xinlong Xiong and Zhaohui Wu. I can not make it without their love and care.

Appreciation to my aunts, Maggie Xiong, Yan Wu, Yun Wu, and Xinling Xiong, my uncles-in-law, Guoxing Sheng and Wenzhou Wang. I will always appreciate all they have done, especially for helping me develop my confidence and giving me family love. A special feeling of gratitude to my loving cousins, Rufei Sheng, Zheyuan Wang, and Junjie Song whose words of encouragement and push for tenacity ring in my ears, and who have forever loved me unconditionally in the journey of seeking who I am.

I also dedicate this work and give special thanks to Dr. Qidi Liu and Yuezong Zhao for being there throughout my entire doctorate program. Both of you have been my best cheerleaders and spiritual props.

I love you all and miss you all beyond words.

Acknowledgments

Dear Dr. Cohen, my advisor and chair of my committee, I know that words cannot express my gratitude to you for your invaluable patience and feedback on both my doctoral study and dissertation writing. I would say thanks so much for going above and beyond and bringing out the best to me. My life won't be the same without an advisor like you. I appreciate the time and effort that you put into the teaching and mentoring profession to me throughout my five years at UGA. I was in an uncertain period when I decided to change my major to educational psychology. It was you accepted me as your student and taught me and helped me with your funding for these five years. You encouraged me to learn statistics, to learn measurement, and to learn writing and speaking English. Dr. Cohen, you helps me build up my confidence. These five years mean a lot to me and the knowledge you have shared with me is priceless, and I will remember your valuable lessons for the rest of my life. Thank you, Dr. Cohen.

I still remember the writing of my first draft of manuscript. I know I did it really bad, but Dr. Cohen carefully read and made comments on every of the sentences. I still keep that version because I can see how much patience you have made to me and my writing. Thank you, Dr. Cohen.

Although there is still a long way for me to go further on academic writing, what Dr. Cohen taught to me is precious and my solid fundamental in my future career. I am very moved and I greatly appreciate what you have done for me. And, thank you, Dr. Cohen.

I also could not have undertaken this journey without my defense committee, Prof. George Engelhard, Prof. Seock-Ho Kim, Dr. Shiyu Wang, and Dr. Sheng Li, who generously provided knowledge and expertise. I am very so proud of being your student. My life has been greatly encouraged by all of you and you taught me to be a good research scientist. Regardless of how far away my career could go to, you are and you will be my forever mentors in my life.

Great appreciation to my forever friends, Cheng Tang, Guang Wang, Zeshen Tang, Xinyu Wang, Rui Zhang, Jie Kong, and Zhifeng Hu, and I love all of you. Without your tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my Ph.D. degree.

I am also grateful to my classmates and cohort members, Jordan M. Wheeler, Madeline Schellman, Constanza Andrea Mardones Segovia, especially my office mate, Yanyan Tan, and senior and junior classmates, Dr. Feiming Li, Dr. Kang Xue, Dr. Yu Bao, Dr. Juwe Wang, Dr. Jiajun Xu, Yawei Shen, Ye Yuayn, and Jing Li, for their moral support. We are always and forever QM DAWGS.

Thanks should also go to my best friends, Zhine Wang, Yutong Yang, Bowen Wang, and Fei Wen, who impacted, inspired, and had numerous parties with me during my doctoral study.

Your belief in me has kept my spirits and motivation high during this exciting process.

Table of Contents

Acknowledgement v

List of Figures ix

List of Tables xii

1 Introduction 1

1.1 The mixed-format assessment 1

1.2 Statement of the Research Problem 4

1.3 Purpose of the Study 8

1.4 Dissertation Overview 9

2 Literature Review 10

2.1 Theoretical Framework 10

2.2 Multiple-choice Item Process Data Analysis: Challenges and Progress 12

2.3 Recent Advances in Constructed-response Item Research 18

3 Sequence Reservoir Model: a New Perspective for Enhancing Multiple-choice Item Process Data **22**

3.1 Computer Log File Containing Non-uniform Action Sequences 22

3.2 Methodology 23

3.3 Design of Experiment 35

3.4 Results of Simulation Studies 43

3.5 An Exploratory Study Using both Process Data and Response Data 50

3.6 Application of Sequence Reservoir Model (SRM) to the Empirical Data 56

3.7 Summary and Discussion 75

4 Topic model: an interpretable algorithm for analyzing constructed-response item process data **78**

4.1 Unsupervised and Supervised Latent Dirichlet Allocation 79

4.2 Comparison of Unsupervised and Supervised Model Analyses of Constructed-response Answers on Two Social Study Assessments: An Empirical Example 84

4.3 Developing an Automated CR Answer Scoring Engine Using SLDA and A Generalized Logit Model 96

4.4 A Hybrid Framework Using Rasch Measurement Models and Topic Model 106

4.5 Summary and Discussion 125

5 Conclusion **128**

Bibliography **133**

Appendices **150**

A Code **150**

A.1 R code 150

A.2 Matlab 194

List of Figures

1.1 An example of a mixed-format ELA assessment for Grades 1 and 2 (taken from the website https://www.lennections.com/assesslets-ela) 5

1.2 A problem-solving item interface from the PISA (https://www.oecd.org/pisa/test-2012/testquestions/question4/) 7

2.1 An illustration of the MC item's time-wrapped process data 13

2.2 The structure of a neural network 15

2.3 An illustration of principal component analysis showing an orthogonal projection $\boldsymbol{\pi}_{\boldsymbol{x}}$ of the original vector $\boldsymbol{x}$ on a lower space U spanned by $\boldsymbol{e_1}, \boldsymbol{e_2}$ 16

2.4 A description of the general process of a topic model 21

3.1 The architecture of a traditional RNN 25

3.2 A reservoir computing framework with three components 28

3.3 Principal component analysis of the features extracted by SRM from Simulation Study I with $l = 3000$ and $n_z = 10$ 44

3.4 Average group classification accuracy with standard deviation yielded by an increasing reservoir for each combination of Number of Examinees and Number of Features . . . 46

3.5 Principal component analysis to the features extracted by SRM from Simulation II with $l = 3000$ and $n_z = 10$ 47

3.6 Average latent trait RMSEs with standard deviations yielded by an increasing reservoir for each combination of two factors 50

3.7 The model fit index values of the *Rsp+ProcData* model versus the *RspData* model . . . 55

3.8 Data sets and assessment block structure 59

3.9 Action frequency in each of the subsets of process data 61

3.10 SVM transformation process 62

3.11 SVM decision boundary and margins 63

3.12 Adjusted AUC value changes against the addition of new item's features in the SVM . 67

3.13 The first ten important features from subset 1 used in the final SVM model 71

3.14 The first ten important features from subset 2 portion 3 used in the final SVM model . 74

4.1 A graphical model representation of the LDA model* 81

4.2 A graphical model representation of the sLDA model* 83

4.3 Plots of DIC for topic models estimated for the U.S. History and Economics items . . 88

4.4 The left graph is a plot of the difference between the observed scores and predicted scores; the Right graph is a histogram of topic proportions for each observed score category . . 93

4.5 The left graph is a plot of the difference between the observed scores and predicted scores; the Right graph is a histogram of topic proportions for each observed score category . . 95

4.6 Classification accuracy yielded by the number of topics under each n-gram condition . 101

4.7 Different model classification accuracy and $QW - \kappa$ scores under optimal topic numbers 102

4.8 A structure illustration of the bi-factor model 108

4.9 Parallel analysis of the ELA data 113

4.10 DIC index on different number of topics 120

4.11 Two topic distributions in each ability category 123

List of Tables

1.1 Some types of items used in educational assessments 2

1.2 Two additional types of assessments 3

1.3 Overview of the chapters in this dissertation 9

3.1 The process of a simple ESN applied on the MC item process data 31

3.2 The process of a simple ESN applied on the MC item process data 36

3.3 Factors and levels that are used for generating action sequences in the simulation 37

3.4 Confusion matrix for classification with three categories. 41

3.5 Average best group classification accuracy with the standard deviations by the extracted features and baseline features using the multinomial logistic regression 45

3.6 Average best latent trait RMSEs with standard deviations by extracted features and baseline features using the LASSO regression 49

3.7 RMSEs of the latent ability recovery given by the Rsp+ProcData model and the RspData model 54

3.8 Variables of the NAEP process data 57

3.9 Descriptive statistics for the NAEP math assessment Subset 1 and Subset 2 process data 60

3.10 Item sequence length and number of features 66

3.11 Number of unique actions provided by each item and their corresponding contribution to the model classification AUC_{adj} values 68

3.12 Interpretation to the first ten important features 70

3.13 Feature number and adjust AUC value for each portion in Subset 2 72

3.14 Interpretation to the first ten important features 73

4.1 Stop words for the U.S. History assessment and the Economics assessment 86

4.2 Number of responses, number of words, and average response length before and after data cleaning 86

4.3 Top 15 highest probability words for the 4-topic LDA model for U.S. History item . . . 90

4.4 Top 15 highest probability words for the 3-topic LDA model for the Economics item . . 91

4.5 Top 15 highest probability words of the 4-topic sLDA model for U.S. History item . . . 92

4.6 Top 15 highest probability words of the 4-topic sLDA model for the Economics item . 94

4.7 Number of scores in each score category of the extended CR item 98

4.8 Descriptive statistics before and after the data clean 99

4.9 The average of classification accuracy and quadratic weighted kappa for different n-grams 103

4.10 The confusion matrix yielded by the unigram 3-topic sLDA model 104

4.11 Unigram model topic structures 105

4.12 Description of items on ELA assessment 111

4.13 Factor loadings for two EFA models 112

4.14 Item parameter estimates from the PCM 114

4.15 Summary statistics from PCM 114

4.16 Item coefficients from the bi-factor and constrained bi-factor model 116

4.17 Model-fit statistics for ELA measurement models 116

4.18 Single-factor loadings from PCM for ELA data 117

4.19 Factor loadings from two bi-factor models for ELA data 118

4.20 Ability measure summary from three measurement models 119

4.21 LDA topic structure with top 20 words for the ELA CR answer 121

Chapter 1

Introduction

This Chapter introduces the mixed-format assessment, research questions, study purpose, and dissertation layout.

1.1 The mixed-format assessment

Item response theory (IRT) models are a well-known family of psychometric models that are used to infer an examinee's status on latent traits based on their responses to assessment items. These models are often employed in large-scale assessment programs. This is true in statewide assessment programs (e.g., the Florida Comprehensive Assessment Tests) as well as in computer-based assessment programs like the Graduate Record Examination and the Graduate Management Admission Test. The most common usage in large-scale computer-based assessment programs has been to employ some type of selected-response item such as a multiple-choice (MC) item. Analysis of this kind of question focuses primarily on the correctness of the choice the examinees make. An important benefit of this kind of question is that it can

be scored rapidly and with high reliability. Open-ended or constructed-response (CR) items, on the other hand, are used less often, mainly because they require hand grading, which is relatively more costly and takes much longer than scoring MC items. MC items are usually scored dichotomously, that is, as either correct or incorrect. CR items are often scored polytomously, and their appeal is that they can sometimes be useful for assessing higher-order cognitive skills (Nickerson, 1989). Table 1.1 lists some commonly seen formats of both item types with detailed descriptions. For example, there are single–select response and multi-select response MC items, with different correct answer numbers. In addition, there are short-answer and extended-response CR items, with different constraints on examinees' textual responses.

Table 1.1: Some types of items used in educational assessments

Format	Description
Single-select	Selected-response items have answer choices usually with one correct key and distractors. Distractors typically represent common misconceptions or common errors that examinees might make.
Multi-select	Multi-select items have multiple possible answer choices with multiple correct responses. The incorrect choices usually reflect common errors or common misconceptions.
Short-answer response	Short answer items are CR items that ask examinees to generate a short response to a prompt (e.g., a task or question). They are usually scored using a rubric.
Extended-response	Extended-response items are CR items that may ask examinees to provide a longer response to a task or question than a short-answer item. Examples are items requiring narratives, informative explanatory, opinion, or argumentative responses to a prompt. They are usually scored using a rubric.

There are also some other types of items, despite differences in the item formats, which are still within the scope of MC items, and CR items. For example, Table 1.2 lists two additional types of assessments, performance assessments (Eisner, 1999), and game-based assessments (Mislevy et al., 2016). The performance assessment is a type of open-ended learning activity or assessment that asks examinees to perform or create a product to demonstrate their knowledge and skills. Game-based assessments consist of learning activities embedded in assessments that are designed to assess examinees' knowledge and skills in the context of

an activity, such as a learning activity. Examinees may respond to either an MC or CR item through interacting with a game or learning activity involved in the item. The response score patterns from these two types of items are usually either dichotomous or polytomous and can be modeled by appropriate psychometric models such as by IRT models (Uto and Ueno, 2018) or by Bayesian statistical networks (Cui et al., 2019).

Table 1.2: Two additional types of assessments

Format	Description
Performance assessments	Performance assessments are usually open-ended assessments embedded in learning activities that ask examinees to perform or to create a product to demonstrate their knowledge and skills. Performance tasks typically provide a scenario/situation so examinees can apply their knowledge and skills in authentic contexts. Performance tasks can be multi-faceted and can provide opportunities to assess knowledge and skills across content areas.
Games-based assessments	Game-based assessments are usually assessments embedded in game-based learning activities that can assess examinees' knowledge and skills through an engaging platform. Game-based assessments provide an alternative to traditional assessments in that they can often integrate seamlessly into the learning activity, such as used as formative items to monitor learning and to assess examinees' ability to transfer the knowledge and skills learned during instruction. Game-based assessments can provide a scenario in which examinees can apply their knowledge or skills, such as in a realistic game-like context.

More generally, MC items are considered to be useful for measuring static knowledge (Tatsuoka, 1991), while CR items are considered appropriate for assessing higher cognitive performance (Nickerson, 1989). Many researchers believe that the use of mixed-format items can sometimes capitalize on the benefits of both item types and increase overall measurement accuracy because these two item formats complement each other. For example, Ercikan et al. (1998) suggested that CR items can provide information about extremely low- or extremely high-ability examinees that may be otherwise poorly assessed by MC items. The mixed use of both MC items and CR items has been reported in large-scale assessments (Hendrickson

et al., 2010; Y. Kim, 2009; Kuechler and Simkin, 2010) based on the requirement that both MC and CR item formats measure the same underlying trait (Swygert et al., 2001).

Computer-based assessments consisting of both MC and CR items are becoming increasingly common due, in part, to improvements in the scoring of CR items (e.g., Shermis, 2014), and to different and complementary ways they provide for measuring different aspects of a given domain or set of domains (Hendrickson et al., 2010). Assessments of English and language arts (ELA), for example, are often composed of both MC items, to assess mechanics of writing, and CR items, to assess higher-order types of reasoning (Choi et al., 2017). Figure 1.1 shows an illustration of a mixed-format ELA assessment containing both two types of questions for Grades 1-2 (https://www.lennections.com/assesslets-ela). In the assessment, examinees are asked to read a prompt such as a story or passage, then to select an answer for the MC item, and finally, to construct a textual answer for the CR items.

1.2 Statement of the Research Problem

One concern with the usual use of IRT for these assessments is that they only model the response score patterns. This works well for the calibration of MC and CR response scores but, once the items have been calibrated and the latent ability scores have been estimated, little attention is further given to any additional information contained in the assessment process. For instance, for the two additional types of assessments listed in Table 1.2, IRT models are useful for modeling item response patterns but may not be that useful for understanding information contained in the item response process data.

Behrens et al. (2019) suggested applying recent advances in measurement models, analytic techniques, and digital tools and environments such as machine learning and artificial intelligence techniques to enrich

Mary Frances Early College of Education
K-12 Assessment Solutions
UNIVERSITY OF GEORGIA

Lennections
Learning Connections

Selected Response (SR) Sample Items Grades 1 and 2

English Language Arts

Sample 1

What lesson did Sam and Tessa learn in the story?

A. They learned to listen to other's advice.

B. They learned to work together as a team.

C. They learned to pay attention to the weather.

Answer Key	
Answer Choice	**Rationale**
A.	The student may not understand that there is no advice given by one character and ignored by the other. Sam and Tessa discuss whether it will rain, but this is not a focus of the story.
B.	The student may not be reading carefully. There is no evidence that getting home required them to learn new teamwork skills or cooperate more than usual.
C.	*Correct Answer*; The student understands that Sam and Tessa's narrow escape from an unexpected storm is the main event of the story. At the end, Sam resolves to check the weather before going out in a boat again.

Constructed Response (CR) Sample Item Grades 1 and 2

English Language Arts

Sample 1

How would you describe the setting at the beginning of the story?

Use a detail from the passage or the picture to support your answer.

Figure 1.1: An example of a mixed-format ELA assessment for Grades 1 and 2 (taken from the website https://www.lennections.com/assesslets-ela)

assessment, since these advances may provide additional construct-related information. Currently, some computer-based assessments have a user-friendly interface making it easy to capture the additional aspects of the responding process during the assessment including examinees' interactions in assessments that involve communication with the computer, detailed traces of actions as examinees navigate the assessment's environment, possibly to solve problems, and to store all these activities in the computer log file. A thesis of this dissertation is that these activities have the potential to provide a rich source of information about the reasoning process during the assessment (Ercikan et al., 2020). Therefore, an important challenge is making sense of these kinds of process data.

For the MC items, the response actions during the assessment process typically consist of sequences of actions with corresponding time points. These sequential time-stamped data are usually referred to as the process data (El Aouifi et al., 2021; Liu and Israel, 2022; Nebel and Ninaus, 2019). Research has been reported on the analysis of these process data either from machine learning algorithms (Auer et al., 2022) or from Bayesian networks (Cui et al., 2019), based on the assumption that these process data occur during and as a result of examinees answering assessment questions, and therefore, are related to examinees' reasoning process. For example, Figure 1.2 shows a problem-solving item from the Programme for International Student Assessment (PISA) which asks the examinee to calculate the best selection based on the item description. There are three selections on the item interface, city subway, country trains, and cancel. Once the examinee selected one of the options, proceeding sets of options will be given, until the completion of this item. All the sequential steps by the examinee, including any reversals to previous steps in making the response, are recorded in the process data. These process data have been shown to provide additional information along with the response score analysis (X. Tang et al., 2020). Therefore, the combination of steps during the response may have the potential to indicate the examinee's reasoning

and thinking process on this item. Examining these data will help to extract and understand the utility of this information.

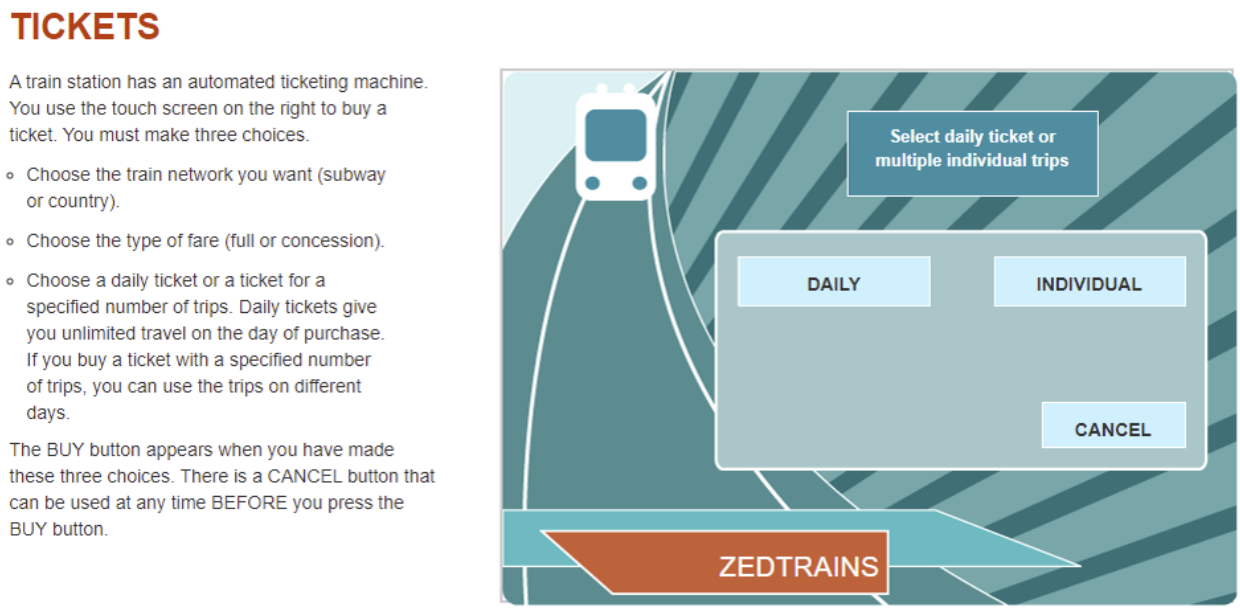

Figure 1.2: A problem-solving item interface from the PISA (https://www.oecd.org/pisa/test-2012/testquestions/question4/)

In this same regard, there is also abundant process information in the examinees' CR item responding process. The textual responses are also considered process data created by each examinee to the CR items. This textual information has been found to contain useful information in addition to the IRT analysis. For instance, recent research has shown that CR items, however, do offer the possibility of providing evidence beyond just the correctness of the answer (S. Kim et al., 2017). They can provide potentially useful information that is often ignored in the scoring process, but which can provide useful additional information about the thinking and reasoning of examinees not typically accounted for in the scoring (Xiong et al., 2019). As a result, potentially useful but ignored information could enhance what can be learned about what examinees know and what they do not know. The developments in machine learning

may also be able to improve the speed of scoring and also the kinds of additional information that can be extracted from these CR items (Cardozo-Gaibisso et al., 2020). Similarly, research on process data has also revealed abundant information that can help to understand examinee response behaviors (Meyer, 2010) and suggest item selection in adaptive testing (van der Linden, 2008). Models have also been proposed to incorporate both item responses and response time to provide additional information in cognitive assessments (Zhan et al., 2018) and to calibrate personal latent ability (T. Wang and Hanson, 2005). These studies suggest that incorporating process data in the modeling can provide useful information when combined with the usual item response modeling.

1.3 Purpose of the Study

In this dissertation, an important objective is to provide exploratory analyses to process data. In this way, we suggest it may be possible to improve what we can learn about examinees' thinking and reasoning by the use of process data. That is, we propose to analyze the additional information in the response process that can then be added to the information we already can obtain from the response score patterns for mixed-format assessments. This dissertation will explore the process data from mixed-format assessments for the following purposes:

1. How might we analyze the process data and interpret the results?
2. What are some applications of process data analysis?
3. In what aspects and to what extent might the use of process data contribute to the latent trait measurement?

1.4 Dissertation Overview

This chapter introduced the background of mixed-format assessments, the types of data that will be analyzed, and the kinds of research questions to be addressed. The remainder of this dissertation will consist of the following. The second chapter reviews related references and introduces the challenges and progress in dealing with process data. Chapter 3 focuses on the analysis of MC item process data, as contained in the log file, with a description of the methodology, simulation designs for studying this methodology, and an illustration of the methodology using empirical data. Chapter 4 analyzes the CR item process data consisting of the textual responses, using both supervised and unsupervised algorithms. Empirical data sets will be utilized in this chapter to illustrate the algorithms, with a focus on automated scoring, writing profile analysis, and a combination of both. Finally, Chapter 5 summarizes the findings and conclusions and points out possible future directions. References and appendices are also attached after Chapter 5. A general overview of summarizing all chapters is given in Table 1.3.

Table 1.3: Overview of the chapters in this dissertation

Chapter	Content description
Chapter 1	Introduction to the mixed-format assessments, data types, and research purposes of this dissertation.
Chapter 2	Review of literature and possible challenges that will be addressed in this dissertation.
Chapter 3	The analysis of MC process data, or computer log file data, with simulation studies and empirical example illustration.
Chapter 4	The analysis of CR process data, or textual responses, with both supervised and unsupervised learning algorithms. The emphases of this chapter are about automated scoring, writing profile analysis, and a combination of both.
Chapter 5	Summary of the findings and conclusions generated in this dissertation with a possible future direction related to this dissertation.
Reference	References that were used in this dissertation.
Appendix	Some codes used in this dissertation.

Chapter 2

Literature Review

2.1 Theoretical Framework

A major objective of assessments is to provide information about the status of examinees on the construct of interest. Current computer-based assessments can measure latent traits with complex collaborative problem-solving items with a virtual interactive interface (Kozma, 2009; Co-operation and Development, 2012). For example, Figure 1.1 in Chapter 1 shows the PISA item employs a virtual interactive interface to track examinees' problem-solving processes. These computer-based assessments can use the dynamic virtual environment to record both the final responses and examinees' real-time response-related activities realized during the responding process (Mislevy, 2019). These kinds of information are associated with examinees' cognitive activities and may be useful for understanding the latent traits that are the focus of the assessment (Han and Wilson, 2022; von Davier, 2017; Wilson et al., 2017; Yuan et al., 2019). A conjecture in this study is that the use of process data, such as the log file data and the textual responses, may add useful information to what we can obtain from the rubric-based scores alone. By analyzing

these process data in addition to the item response data, we can potentially make use of extra information that would reveal patterns that could otherwise be missed. Recent research on response process data has focused on this general issue (Leighton, 2017; Oranje et al., 2017). The assumption in that research is that the process data can be used to augment the information in the scores by attending to potential additional information in the responses that may reflect the process of thinking and reasoning (Clifton Jr et al., 2016; Ercikan and Pellegrino, 2017; van der Linden, 2011; S. Wang et al., 2021).

Mislevy (2019) suggested that two basic types of analytic procedures can be used to contribute to analyzing and modeling these behavior data. The first type of procedure is to characterize the evidence in the process data. In other words, we may extract features from the process data using techniques such as data mining, knowledge engineering, and computational linguistics (Bejar et al., 2016). This type of procedure is akin to the processes applied by human raters when they evaluate the performance items using the scoring rubic with human judgement instead of computing scores using mathematical equations with features (Mislevy, 2019). The second type of analytic procedure uses measurement models to operationalize variables related to the constructs of interest. That is, by tracking, accumulating, and synthesizing evidence across examinees' ongoing behaviors, it is possible to construct operationalized variables related to the targeted constructs. Then, the latent traits can be modeled probabilistically, depending on the constructed variables, through measurement models. Current methodologies and models developed to learn the process data are primarily within these two basic types of analytic procedures.

2.2 Multiple-choice Item Process Data Analysis: Challenges and Progress

The process data for MC items, obtained from examinees interacting with a computer-based assessment item, are recorded in the computer log files. These data often come in the form of sequences of events with time stamps. For example, an examinee may select response "A" at the time t, and switch to response "C" at the time $(t + 1)$. In this way, the record of each item response comes with a sequence of ordered and time-stamped actions. These are also sometimes referred to as sequential data. Figure 2.1 illustrates this type of data, from which we can see the varying-length and diverse set of activities contained in log files produced by the responses of different examinees. These types of data are more readily available in responses obtained with the use of computer-based assessments. One direction for applying these process data is the use of response time (Oranje et al., 2017). Differences in response times, for example, could be important indicators of differences in assessment-taking strategy, differences in the difficulty of questions, assessment speededness, or possible lack of motivation for the assessment (Guo et al., 2016; van der Linden, 2011). Information from response times might also include the effects of cognitive demands in the assessment item to understand item content, item format, and context (Wise and DeMars, 2006; Wise and Kong, 2005). Differences in response time, in other words, can potentially be used to support inferences on score meaning, the interpretation of the validity of assessment results, and performance differences of different examinee groups (Ercikan et al., 2020; Wise and DeMars, 2006).

In addition to the response time, the associated actions that are not captured by final solutions and responses may be important indicators to identify examinees' final performance level as the features in

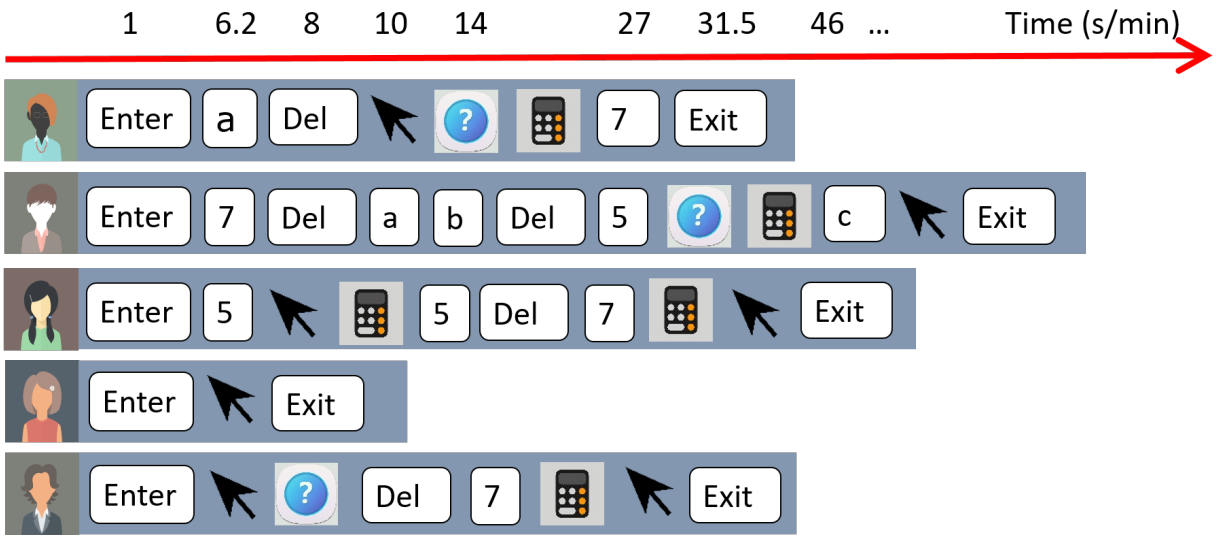

Figure 2.1: An illustration of the MC item's time-wrapped process data

the process data may provide information on how examinees from various backgrounds engage with the assessment (X. Tang et al., 2020). In this way, process data provide supporting evidence about individual performance in addition to scores reflecting the correctness of answers. Yuan et al. (2019) pointed out that the feature extraction methods can be divided into two types, the theory-driven method, and the data-driven method. The theory-driven method, for example, derives some indicator variables from the process data based on a scoring rubric. The use of the theory-driven method explicitly connects the construct being measured by the assessment with the process performance of examinees. In this way, it can be analyzed with appropriate measurement models to obtain information about the examinees' status on the latent trait(s). One concern with this approach, however, is that it may also introduce a high cost in terms of human effort and can only focus on the rule-based actions while ignoring information in other actions (Han and Wilson, 2022).

On the other hand, the data-driven method may overcome some of the disadvantages of the theory-driven method by employing statistical and data science methods to extract features. A challenge in handling the complex and diverse process data using the data-driven method, as illustrated in Figure 2.1, is that unlike the response data analyzed with IRT, the length of the sequence for each examinee on each item is not fixed but depends on the number of actions the examinee takes. To handle this type of data, Xu et al. (2020), for example, proposed a latent topic model with a Markov transition process to model process data that consist of time-stamped events. For each examinee, the sequence of latent topics can be viewed as the examinee's latent state. In the model, each topic can be viewed as a group of event types sharing similar meanings. By using topic transition probabilities along with response times, Xu et al. found that the behavior patterns in the data captured examinees' learning strategies.

Neural networks (Rojas, 2013) are another commonly used approach in the data-driven method. Neural networks are computer algorithms based on a collection of connected units or nodes called artificial neurons. The structure of a neural network is diagrammed in Figure 2.2. This is intended to provide a generative diagram about how the way the human brain receives and analyzes information from the signal input to the results output. Neural networks constitute core information processing technology in the artificial intelligence and machine learning fields and can be described as using forward or backward propagation (Cui et al., 2016; Knierim, 2014). From the input data to the output layer, this process is called forward propagation. The neural networks computes the loss after each iteration and then restarts the process from the input layer to update the estimation. This process is called backward propagation. The technology is designed to recognize patterns and store them in vectors into which real-world data, such as text or time series, can be translated.

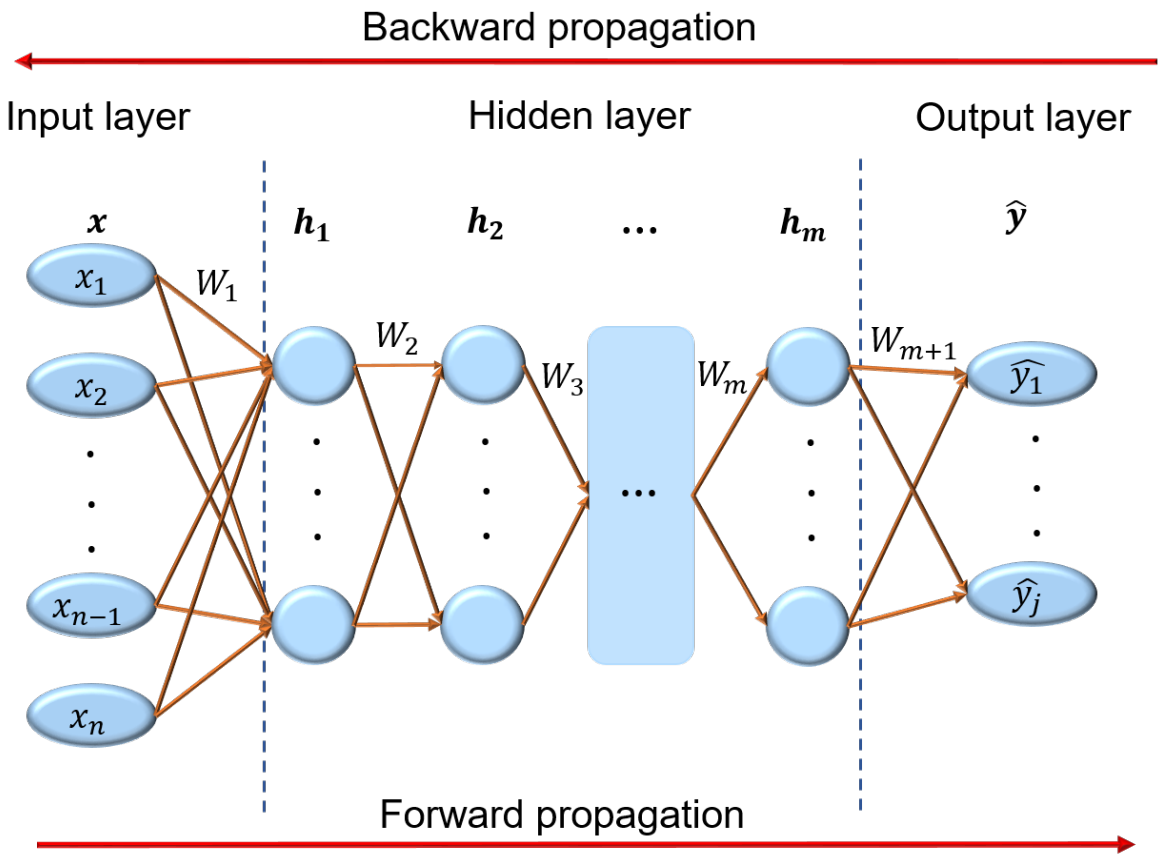

Figure 2.2: The structure of a neural network

Recurrent Neural Networks (RNNs; Medsker and Jain, 2001) are generalizations of feedback neural networks with the addition of internal memory. RNNs are a class of neural networks that allow the previous outputs to be used as inputs to the next state while having hidden states, which are used for the processing of sequential data (Rumelhart et al., 1986). RNNs consider the current input and the knowledge it has learned from the past state to make a decision. RNNs have important advantages over other statistical methods in that they do not require the explicit encoding of domain knowledge, do not depend on the test item's prior knowledge, and can capture more complex representations of examinees' response

actions (Goodfellow et al., 2016). X. Tang et al. (2021) suggested a sequence-to-sequence autoencoder that takes an action sequence and produces a reconstructed action sequence which can then be used to compress response processes into standard numerical vectors. The sequence-to-sequence autoencoder uses RNNs that allow previous outputs to be used as inputs while having hidden states as components of the encoder and decoder. The method automatically extracts numerical features from a computer log file and does not require manual feature engineering using domain knowledge.

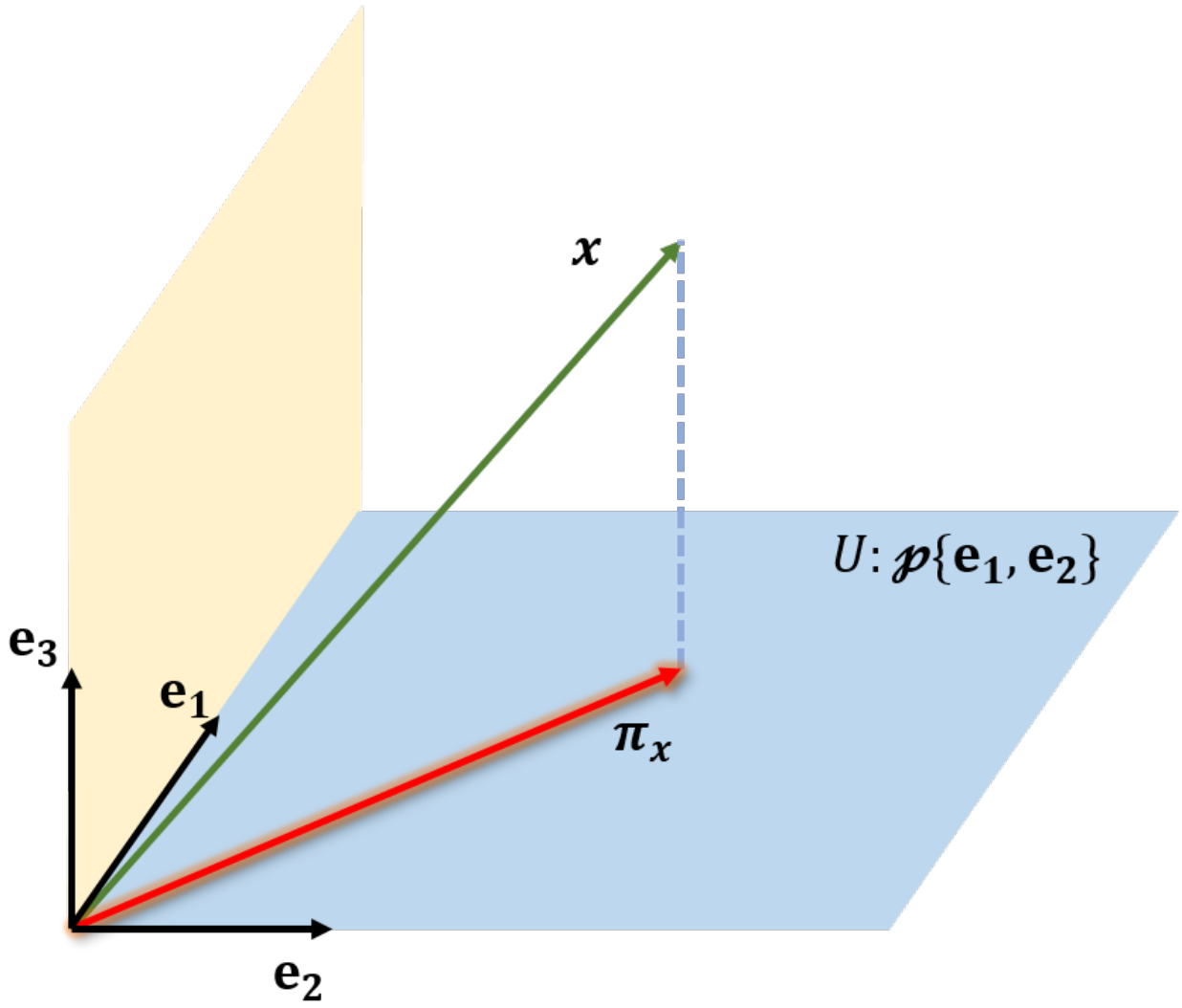

Figure 2.3: An illustration of principal component analysis showing an orthogonal projection π_x of the original vector x on a lower space U spanned by e_1, e_2

One commonly adapted way to interpret the extracted features is to use principal component analysis (PCA; X. Tang et al., 2020; X. Tang et al., 2021). PCA reduces the dimension of the original data and finds a lower-dimensional representation in a subspace. Figure 2.3 shows the illustration of how PCA reduces the data dimension. For a vector $\boldsymbol{x} \in \mathbb{R}^3$, PCA finds an orthogonal projection of $\boldsymbol{\pi_x}$ of $\boldsymbol{x}$ in a two-dimensional subspace $U : \mathcal{P}\{\boldsymbol{e_1}, \boldsymbol{e_2}\}$. It can be seen that the vector $\boldsymbol{\pi_x}$ is a representation of $\boldsymbol{x}$ in a lower dimension, since it is the closet vector to $\boldsymbol{x}$ in the space U, where the minimum distance is equal to $||\boldsymbol{\pi_x} - \boldsymbol{x}||$.

Although it is applicable to a wide range of process data, training RNNs is both challenging and computationally expensive, due to more complex signal movements (Pascanu et al., 2013). This is because RNNs can have signals traveling both forward and backward, and may also contain complex iteration loops. Therefore, some methods that could utilize the advantage of an RNN and at the same time overcome its computational disadvantages could be a possible direction to learn features from the process data. Reservoir computing (Verstraeten et al., 2007) is a computational method derived from RNNs theory that learns data representations through the dynamics of a fixed, non-linear system called a reservoir. It is a kind of machine learning algorithm for processing information generated by dynamical systems with time-dependent data (Gauthier et al., 2021). The detailed structure of the reservoir computing will be introduced in Chapter 3. It has been shown that reservoir computing only requires very small training data sets and may be used with linear optimization, thereby requiring lower computing resources, but performing as well as other machine learning methods, such as regular RNNs and deep learning (Bompas et al., 2020; Vlachas et al., 2020). The use of reservoir computing instead of traditional RNNs has been reported in applications such as text classification (Schaetti, 2019), feature extraction for images (Tong and

Tanaka, 2018), and time-series representation and prediction (Bianchi et al., 2020; Wyffels and Schrauwen, 2010).

The use of reservoir computing does not yet appear to have been reported in modeling process data in educational assessments. Since neural networks have been shown to be useful in extracting process data using regular RNNs, exploring the use of reservoir computing on educational process data could be a useful technique since it provides the capability of solving both the data processing challenge and the costs of computing of a complex RNN. The use of the reservoir computing method on educational process data, in other words, may also facilitate investigating examinees' actions, understanding where important user interactions are, and aligning both with whatever data are captured and available in the system.

2.3 Recent Advances in Constructed-response Item Research

The response process data for the CR items are different from those for the MC items. As mentioned previously, we suggest that there is information in the text of the responses in addition to that which is modeled by IRT in the scores of the item. The focus on the analysis of the text of examinees' responses is made possible given recent advances in the development of natural language processing (NLP) techniques (Chapelle and Chung, 2010; R. M. Kaplan, 1992) such as topic models (S. Kim et al., 2017) and neural networks (Nam et al., 2014; S. Tang et al., 2016). The original goal of NLP was to develop algorithms to allow computers to extract information from the natural language to perform some tasks such as speech processing (Kamath et al., 2019), text summarization (Lai et al., 2015), and text classification (Merchant

and Pande, 2018). The use of NLP on educational texts has been reported on objects such as automated scoring (Flor and Hao, 2021) and response time prediction (Baldwin et al., 2021).

NLP methods may be able to learn and select features from texts based on neural networks with less human effort (Cai, 2019; Liang et al., 2018). The extracted features from neural networks, however, are not necessarily easy to interpret (Montavon et al., 2018). Among the many NLP models, probabilistic topic models include supervised and unsupervised techniques designed to detect the latent topics which occur in a collection of textual documents. These topics are the latent themes (referred to as latent topics, or more simply as topics) that occur in a collection of textual documents. Topic models have been used to determine the latent topics in examinees' answers to CR items by detecting patterns and recurring vocabularies (e.g., S. Kim et al., 2017; Xiong et al., 2019). Each word in a corpus of documents has a probability of occurring in each topic in the model. The topic model is characterized by the number of latent topics in the model, by the probability of use of each topic in each document, and by the probability of each word in the corpus in each topic in the model. In this way, analyzing CR answers with topic models do not require pre-defining features but can provide interpretable topics for use in understanding examinees' thinking and reasoning as exhibited in their responses (Choi et al., 2021). The results from the topic model have been found to provide useful and important information to reflect examinees' thinking and reasoning in their CR responses (Cardozo-Gaibisso et al., 2020). The topic model has been used in exploring topics in CR response data. For example, in an assessment designed to measure the effects of an instructional intervention teaching middle grades examinees the process of science inquiry, S. Kim et al. (2017) found no differences in performance between the instructional treatment group and a business-as-usual group in their scores on the CR items. Significant pre-assessment to post-assessment differences

were detected, however, in the use of the latent themes detected by topic models in the CR responses by examinees receiving the instructional intervention.

Figure 2.4 describes the general process of estimating the topic model. First, the major results of the topic model are given as the word distributions of the topics (γ) and the topic distributions of the documents (η). The word distributions of the topics γ give an overview of the corpus. This allows us to detect the latent themes in the text. The word distributions of the topics γ are a set of multinomial distributions. Next, the words in the examinees' answers are grouped into topics based on the probabilities of co-occurrence. Each topic may have different response proportions for each examinee. The distributions of words in each of the topics indicate the different proportions of an examinee's use of each topic in the model.

Topic models based on latent Dirichlet allocation (LDA; Blei et al., 2003) were originally developed to evaluate the text of large corpora, such as abstracts of scientific journals, over an extended period of time. LDA was applied to the CR answers of university teachers' self-assessment surveys for the sake of understanding strategies used by teachers to improve student retention (Buenaño-Fernandez et al., 2020). LDA also has been used to understand examinees' CR responses. For example, LDA was used to detect changes in topic use in students' writing (Southavilay et al., 2013). Results indicated that the evolution of topics based on LDA analysis clearly showed the changes and improvements in students' writing. Ramesh et al. (2014) explored the use of topic models to analyze the latent themes students use in discussion forums in massive open online courses (MOOCs) as indicators of student retention in the course. Results indicated that features detected by the topic model helped predict student retention in the courses. Chen et al. (2016) applied LDA to explore themes in pre-service teachers' journals about their teaching experiences. Results of the topic model analysis were found to help predict course grades. Xing et

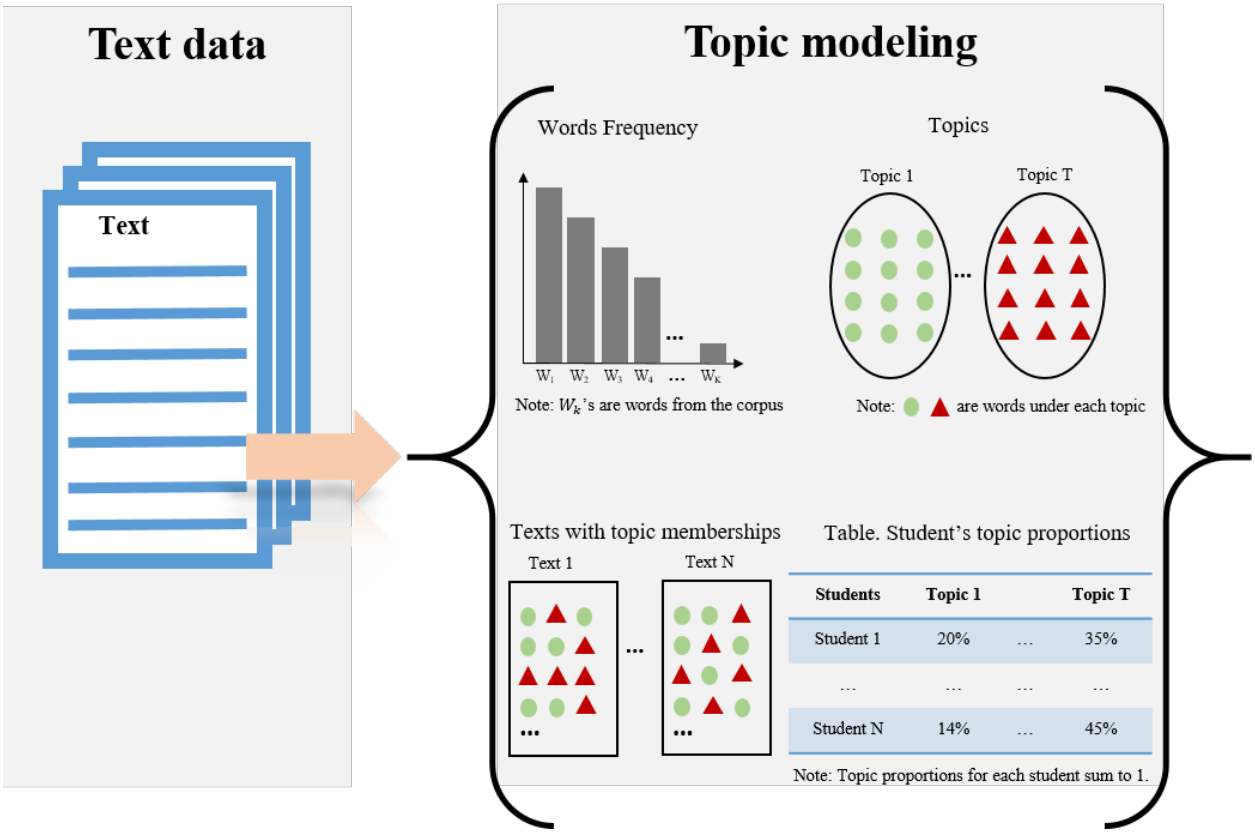

Figure 2.4: A description of the general process of a topic model

al. (2020) also used the LDA as a content analysis tool to identify underlying patterns in students' scientific argumentation. Their results indicated that LDA could discover semantic patterns from students' writing within particular domains and help teachers to improve students' writing.

Both the unsupervised LDA and supervised LDA can be used to help interpret and expand the information obtained from examinees' responses to the CR items. We suggest, therefore, that the results from the LDA analysis can provide useful and important information that can augment the information provided by the scores on examinees' CR responses, thus amplifying what is learned about the measured construct.

Chapter 3

Sequence Reservoir Model: a New Perspective for Enhancing Multiple-choice Item Process Data

This chapter discusses the exploratory MC item's process data analysis.

3.1 Computer Log File Containing Non-uniform Action Sequences

In this chapter, we present an investigation of the use of reservoir computing for extracting features or learning representations from the process data of MC items. As mentioned above, the examinee's sequence of actions for each MC item is recorded in the computer log file along with the timestamps of each action.

These data are thought to provide a detailed picture of steps each examinee has made in responding to an item. This includes such information as the order of actions, the revision of answers, and the amount of time spent on that item. The sequence of actions by each examinee on each item is likely to be different within a single test and between examinees in the population. Thus, it is the case that these process data usually will not be in a uniform fixed format that can be directly analyzed by a traditional statistical model.

In this chapter, therefore, a sequence reservoir model is proposed to extract the standard format features for each examinee. Two challenges are addressed in this chapter. First, each action is a categorical variable and it is mapped into a numerical representation. Second, the non-uniform sequences are transformed into fixed-length features that can be used to infer information about each examinee. A simulation study is proposed to examine the utility of this model under practical testing conditions, and an illustration using empirical data is also presented.

3.2 Methodology

3.2.1 Action embedding

Categorical actions can not be always be easily processed directly, so actions are transformed into numerical representations, referred to as action embedding. For the zth item, suppose there exist a total of n_z possible unique actions which are denoted as $\boldsymbol{S} = (s_1^z, \ldots, s_{n_z}^z)$. The lth examinee may use one of the n_z actions at each time point t when responding to the item. Suppose the lth examinee employs a time of T_l to finish that item, and at each time point t, the examinee's action is d_t, where $d_t \in \boldsymbol{S}(t = 1, \ldots, T_l)$, then in the computer log file, the final record data has a data structure as shown in Equation 3.1

$$
\begin{aligned}
D &= (\boldsymbol{d^1}, \cdots, \boldsymbol{d^l})^T \\
&= \begin{bmatrix} d_1^1 & \cdots & d_{T_1}^1 \\ \vdots & \ddots & \vdots \\ d_1^l & \cdots & d_{T_l}^l \end{bmatrix}
\end{aligned} \tag{3.1}
$$

where the lth row represents a sequence of actions $\boldsymbol{d^l}$ given by the lth examinee, and each row has a unique length $\boldsymbol{d^l} = (d_1^l, \ldots, d_{T_l}^l)$ because T_l varies among different examinees.

The objective of embedding is to first map each of the n_z unique categorical actions to a N_u dimensional vector $\boldsymbol{e}$, where N_u is called embedding size which was learned by our model. So for all n_z unique actions, there are n_z unique N_u dimensional embedding vectors and each corresponds to an action s_n^z. The embedding matrix is a $n_z \times N_u$ dimensional matrix consisting of all embedding vectors such that $E = (\boldsymbol{e_1}, \ldots, \boldsymbol{e_{n_z}})^T$.

One-hot matrices were employed to map each action. A one-hot matrix X^l has dimension $T_l \times n_z$ in which each row has a "1" representing the action taken by examinee l at time point t, and the remaining $n_z - 1$ elements are all "0". For example, if the second action s_2^z is employed by lth examinee at time point t, then the tth row of the one-hot matrix X^l can be represented as $(0, 1, \ldots, 0)$ in which the "1" appears at the second element of this vector. Therefore, each embedded action sequence $\boldsymbol{d}^l$ can be represented as a matrix $A^l \in \mathbb{R}^{T_l \times N_u}$, which is the multiplication of the one-hot matrix X^l and the embedding matrix E. In this matrix, each row represents an embedded action. Mathematically, this can be denoted by Equation 3.2

$$
A^l = X^l E \tag{3.2}
$$

Denote each row vector of A^l as $\boldsymbol{a}^l$. This can be viewed as a result of selecting and reordering each row of the embedding matrix E based on each examinee's action order.

3.2.2 Recurrent Neural Networks (RNNs)

RNNs can be used for processing sequential data by embedding the time-dependent actions of the inputs into their recurrent structure. RNNs are capable of representing a dynamical system driven by sequential inputs, due in part to their feedback relationships with other inputs in the sequence (i.e., the previous outputs from the RNN that can be used as inputs for the next step).

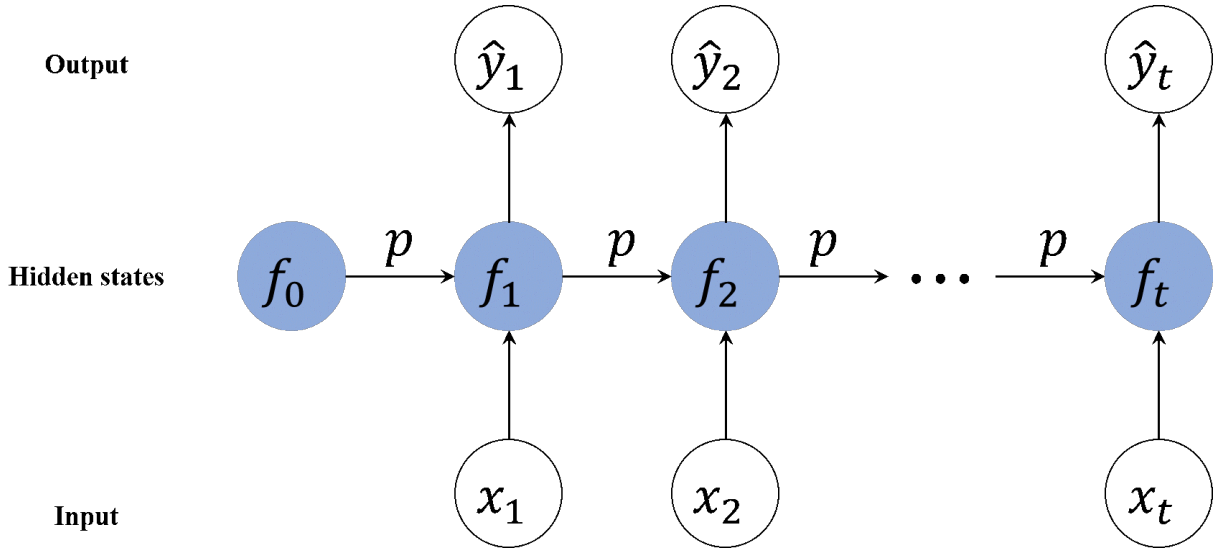

Figure 3.1: The architecture of a traditional RNN

Figure 3.1 shows a common type of RNN called a Many-to-Many (Luong et al., 2015). This structure basically has three components: input, hidden states, and output. Each input node stands for the action at a time point. The RNN makes use of the current input $\boldsymbol{x}_t$ and a summary of previous states $\boldsymbol{f}_(t-1)$ by activation function $p(\cdot)$ to produce updated state $\boldsymbol{f}_t$ as shown in Equation 3.3. This, in turn, produces an output $\hat{\boldsymbol{y}}_t$ at each time step t using another function $g(\cdot)$ and Equation 3.4. The functions $p(\cdot)$ and

$g(\cdot)$ are estimated from the training data. In this learning process, the inputs $\boldsymbol{x}_t$'s are M dimensional vectors, and the hidden states $\boldsymbol{f}_t$'s are N_f dimensional and served as the neural memory that helps transit the input information sequentially. In summary, the sequence of actions is processed consecutively in the input layer, and a hidden state is computed from both the input action and the previous hidden state. The final outputs are yielded as a function of the hidden states $\boldsymbol{f}_t$ at each time point t.

$$\boldsymbol{f}_t = p(\boldsymbol{x}_t, \boldsymbol{f}_{t-1}) \tag{3.3}$$

$$\hat{\boldsymbol{y}}_t = g(\boldsymbol{f}_t) \tag{3.4}$$

Although RNNs are capable of capturing complex nonlinear information from sequential data, they are not easy to train, because they are different from the feed-forward networks. In feed-forward neural networks, signals only travel one way, from an input layer to one or more hidden layers, finally moving forward to the output layer. Signals in RNNs, however, can travel both forward and backward and may contain various "loops" in the network, where numbers or values are fed back into the network. In addition, information at a distant point in the network is sometimes lost, because it is considered lower in information than that which is close by.

Addressing that problem has led to research on reservoir computing methods, such as Echo State Networks (ESNs; Jaeger, 2001) and Liquid State Machines (LSMs; Maass et al., 2002). These methods required less learning time to converge and achieved good model accuracy (Chouikhi et al., 2019). The work of reservoir computing with sequential data has been reported in other areas such as time series prediction (Bianchi et al., 2020), human brain signal processing (Sun et al., 2019), and document classification (Schaetti, 2019). There does not yet appear to be research reported, however, on the performance

of reservoir computing for educational process data, although the MC items' process data is also a type of time-dependent data. These data have been successfully analyzed by RNNs as discussed in Chapter 2 literature reviews. Therefore, this research was designed to employ reservoir computing with an optimization algorithm to handle the log file for MC items.

3.2.3 Echo State Network (ESN)

The ESN contains an RNN-based computational framework, making it suitable for processing sequential data (Jaeger, 2001). The ESN has been reported with high classification and prediction accuracy for time series problems (Ma et al., 2016) and has successfully been used to extract useful features from time-series data (Sun et al., 2019). The ESN has been shown to achieve comparably good performance and significantly lower training times with respect to RNN processing of sequential data (Jirak et al., 2020). This is because the ESN reduces the training-related challenges by fixing the dynamics of the reservoir and only training the linear output layer. As shown in Figure 3.2, a general ESN system consists of three components: an input layer for sequential data, a random sparse recurrent hidden layer called a reservoir layer, which consists of an untrained RNN that functions as a temporal kernel by mapping the input into a high-dimensional feature space, and an output layer for training the high-dimensional features resulting from the reservoir. The reservoir in ESN is the internal structure of the system. The dynamic neurons interconnected within the reservoir are activated by a nonlinear function such as a hyperbolic tangent. Each of the dark bubbles in Figure 3.2 represents a neuron within the reservoir. The major roles of the reservoir in reservoir computing are to first nonlinearly transform the sequential inputs to a high-dimensional space and then store information by use of recurrent neuron loops.

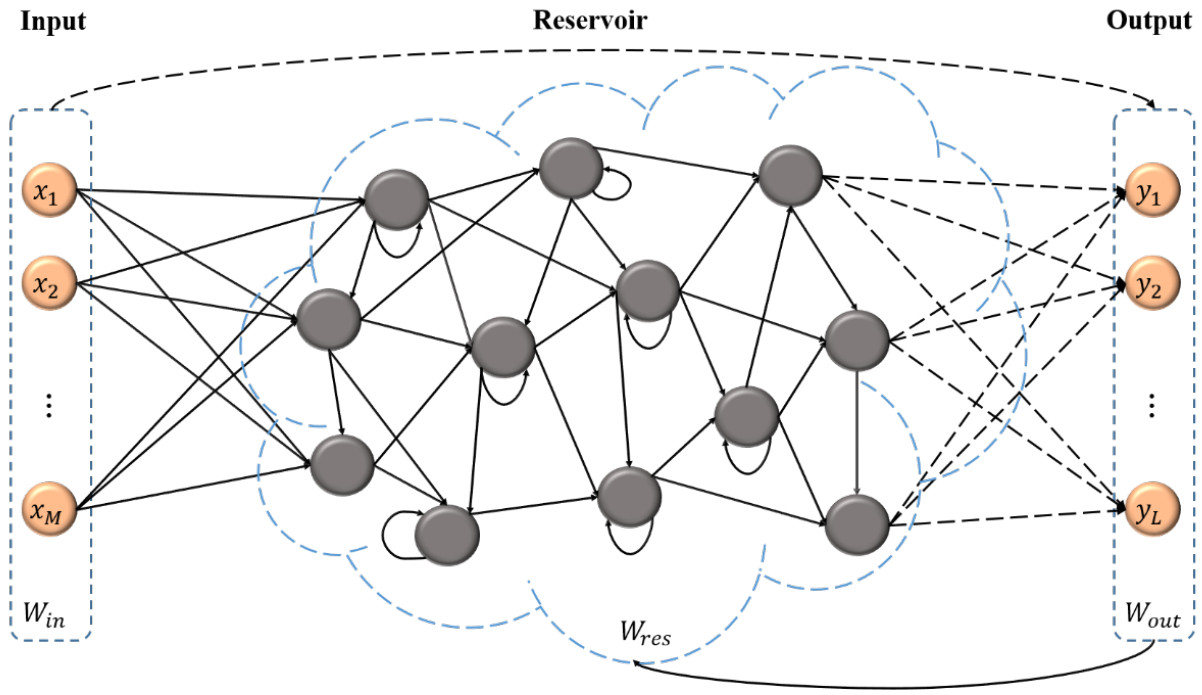

Figure 3.2: A reservoir computing framework with three components

The reservoir computing model differs from conventional RNNs in that the input weights, W_{in}, and the weights of the recurrent connections within the reservoir, W_{res}, are not trained, whereas the output weights, W_{out}, are trained with a simple learning algorithm such as regression or classification. The output is a layer that performs a simple linear transformation on the output of the reservoir. The solid lines in the figure represent fixed connections while the dashed lines define connections that need to be learned by the reservoir computing system. In this way, by using the embedded information from short-time memory (i.e., close-by neurons), reservoir computing can achieve the same performance as provided by utilizing the information from the complete RNN (Gallicchio, 2018; Takens, 1981). The training of only the output instead of all the weights in the network indicates that, as long as an RNN possesses the property of using the short-time memory, i.e., close-by embedded information, supervised adaption of all interconnection

weights is not necessary, and only training a supervised output that has no memory is sufficient to obtain accurate performance. The supervised output means that the extracted features from the reservoir are guided or supervised by a label, such as the actions themselves, examinees' responses, or teachers' grades. This relatively simple and fast training process is an advantage of reservoir computing and makes it possible to reduce the cost involved in computing all the weights in a conventional RNN.

Suppose we have a total of L examinees for a test, and for the lth examinee, the ESN model updates the hidden state of each input with Equation 3.5 and 3.6

$$\tilde{x}^l(t+1) = f\left(W_{in}\left(\boldsymbol{a}^l(t+1)\right)^T + W_{res}x^l(t)\right) \tag{3.5}$$

$$x^l(t+1) = (1-\alpha)x^l(t) + \alpha\tilde{x}^l(t+1) \tag{3.6}$$

where $\boldsymbol{a}^l(t) \in \mathbb{R}^{N_u}$ is the input embedded vector at time point t from the matrix Equation 3.2, $x^l(t) \in \mathbb{R}^{N_x \times N_u}$ is a matrix of reservoir neuron activation and $\tilde{x}^l(t+1) \in \mathbb{R}^{N_x \times N_u}$ is its update, $f(\cdot)$ is the activation function which is usually a hyperbolic function $tanh(\cdot)$, $W_{in} \in \mathbb{R}^{N_x \times 1}$, $W_{res} \in R^{N_x \times N_x}$ are the input and reservoir weight matrices respectively, and $\alpha \in (0, 1]$ is the leaking rate. In order to use Equation 3.5 and 3.6 to update the model parameters, one additional thing that should be noted is that the maximal absolute eigenvalue of the reservoir matrix W_{res} should be less than 1. The maximal absolute eigenvalue of the reservoir matrix W_{res} is a central parameter of an ESN, which is also named as the spectral radius ρ of the reservoir weight matrix. When the ESN maintains $\rho < 1$, this is also called the echo state property, the ESN can properly work and use Equations 3.5 and 3.6 to update mode parameters.

Denote the vertical concatenation of state vector and input vector at each time point as in Equation 3.7

$$H^l(t) = \left[x^l(t) : \left(\boldsymbol{\alpha}^l(t)\right)^T\right] \in \mathbb{R}^{(1+N_x)\times N_u} \tag{3.7}$$

then keep the last output matrix as a summary of information, then the output layer is defined as in Equation 3.8

$$\left(\boldsymbol{y}^l(t)\right)^T = W_{out} H^l(t) \tag{3.8}$$

where $y^l(t) \in \mathbb{R}_u^N$ is the target vector, the $W_{out} \in \mathbb{R}^{1(1+N_x)}$ is an output weight matrix, which can be learned by a regression or classification model.

Given the structure of ESN, the following hyper-parameters are first initialized and not trained in ESN. These are the ESN reservoir size N_x, the leaking rate α, spectral radius ρ, the reservoir weight matrix W_{res} and input weight matrix W_{in}. The input of the model is the embedded action sequence matrix A^l for each examinee. Calculate the last output summary of information $H^l(t)$ as from Equation 3.5 to 3.7, and calculate the average of each column to get a vector $\boldsymbol{h}^l \in \mathbb{R}^{N_u}$, which is the raw representation vector for the lth examinee. Conduct a principal component analysis (PCA) of the raw representation matrix consisting of all l examinees, yields a resulting matrix which contains final representation vectors for all l examinees. This whole process given by ESN can be described as in Table 3.1.

3.2.4 Particle swarm optimization and singular value decomposition

In ESN, because the input weight matrix W_{in}, reservoir size N_u and reservoir weight matrix W_{res} are randomly generated but not trained, the initialization of these parameters influences the ESN model performance. Specifically, research has shown that the singular value spectrum of the reservoir weight matrix

Table 3.1: The process of a simple ESN applied on the MC item process data

Process	Step
1	Initialize the ESN by constructing the input weight matrix, reservoir weight matrix, and leaking rate.
2	Calculate the representation matrix $H^l(t)$ consisting of the vertical concatenation of state vector and input vector for the lth examinee by Equations 3.5 to 3.7.
3	Calculate the last output summary of information $H^l(t)$ and calculate the average of each column to get a horizontal vector $h^l \in \mathbb{R}^{N_u}$, which is the extracted feature representation vector for the lth examinee.
4	Denote the row combination of each horizontal vector $\boldsymbol{h}^l$ as a representation matrix $\mathcal{H}$, in which each row is a representation vector to the lth examinee.

closely affects the ESN performance (F.-J. Li and Li, 2017; Strauss et al., 2012). Therefore, as optimizing the parameters of the reservoir weight matrix may provide a better ESN structure, an optimization algorithm called particle swarm optimization (PSO; Kennedy and Eberhart, 1995) with singular value decomposition (SVD; Wall et al., 2003) is introduced and utilized in this chapter with the ESN.

PSO is an evolutionary learning method for optimizing parameters. PSO has demonstrated its superiority with the artificial neural network on nonlinear pattern classification (Garro and Vázquez, 2015) and recurrent neural network on the sequence prediction (Juang, 2004). It has been further compared with the commonly used backpropagation method for the neural network training, and results indicated that the neural network parameters would converge faster with the PSO than with the backpropagation (Gudise and Venayagamoorthy, 2003). Chouikhi et al. (2017) used PSO in ESN to pre-train the fixed reservoir matrix weights and the trained reservoir matrix weight was then applied in ESN to process time series. Their results suggested that using PSO can enhance the learning results of ESN for time series forecasting.

This system searches for optimal solutions by iteratively updating values. The system of PSO is first initialized with a group of random particles. In this system, each particle is a single solution (a set of singular values) that will move through the search space to a global optimum. Each particle r can be marked by a pair of values called position and velocity, denoted as (p_r, v_r). Some terms will be introduced such as the personal best $p_{r,best}$, which is the best solution the rth particle has achieved so far, and the global best g_{best}, which is the best value obtained by any particle in the whole system. If the algorithm stops running, the global best value would be taken as the optimal value. For the obtained best value at each step, some constants serve as controlling coefficients and determine their importance for calculating the movement of the particle, as described in Equations 3.9 and 3.10. The particle's velocity and the position of the rth particle at the kth time point in a search space can be updated with the following Equations 3.9 and 3.10:

$$v_{r,k+1} = W_{tia} v_{r,k} + C_1 \times rand[0,1] \times (p_{r,best} - p_{r,k}) + C_2 \times rand[0,1] \times (g_{best} - p_{r,k}) \quad (3.9)$$

$$p_{r,k+1} = p_{r,k} + v_[r, k+1] \quad (3.10)$$

where the W_{tia} represents the inertia weight that is applied to control the search, C_1 and C_2 are constants controlling the displacements of particles toward the local or the global optima, and the $rand[0,1]$ gives a random value in the range of $[0,1]$. The updated solutions are evaluated by a fitness function, which was defined to indicate the quality and convergence of optimization.

SVD is a statistical method for matrix factorization. It generalizes the eigenvalue decomposition of a square normal matrix with an orthonormal eigen-basis to any dimensional matrix. It has been used for

training neural networks such as deep neural networks (Qasem and Mohammadzadeh, 2021) and growing ESN (Y. Li and Li, 2019) due to its significantly better prediction accuracy and higher estimation performance with less tunable parameters and less time. It was further suggested that using SVD for training neural networks can reduce the high dimensionality and efficiently improve the network performance (C. H. Li and Park, 2009). The equation for SVD of an $m \times n$ matrix X with rank r can be written as Equation 3.11,

$$X = USV^T \tag{3.11}$$

where U is an $m \times n$ matrix, S is an $n \times n$ diagonal matrix, and V^T is also an $n \times n$ matrix. The columns of U, $\boldsymbol{u}_k$, are left singular vectors, and $\boldsymbol{u}_i\boldsymbol{u}_j = 1$ for $i = j$ and $\boldsymbol{u}_i\boldsymbol{u}_j = 0$ for $i \neq j$. The rows of V^T, $\boldsymbol{v}_k$, are right singular vectors, and $\boldsymbol{v}_i'\boldsymbol{v}_j' = 1$ for $i = j$ and $\boldsymbol{v}_i'\boldsymbol{v}_j' = 0$ for $i \neq j$. Each element of matrix S is only nonzero on the diagonal. These are named the singular values. Therefore, $S = diag(s_1, \cdots, s_n)$ and $s_k > 0$ for $1 \leq k \leq r$, and $s_k = 0$ for $(r + 1) \leq k \leq n$. The ordering of the singular vectors is usually determined by high-to-low sorting of the singular values, with the highest singular value in the upper left index of the S matrix.

3.2.5 Sequential reservoir model: ESN with SVD based PSO for the process data

The Sequential reservoir model (SRM) in this chapter is constructed by employing the ESN with SVD based PSO to create the reservoir weight matrix with optimized singular values. That is, the PSO optimizes the singular values first, and then the singular values are used to construct a reservoir weight matrix by SVD. Once the reservoir has been constructed, the ESN can train the data, as it does in the simple ESN. The

common feedback RNN models, as introduced above, use the internal recurrence to iteratively process data from one layer to another in order to obtain the optimal parameters. So, SRM results in lower computation costs than repeated training of the output layer, as is required by the common RNN models. Given the SVD in Equation 3.11, the reservoir weight matrix $W_{res} \in \mathbb{R}^{N_x \times N_x}$, can be decomposed as in Equation 3.12

$$W_{res} = USV^T \tag{3.12}$$

where $U^TU = V^TV = I$, I is an identity matrix, and U and V are two orthogonal matrices. $S = diag(\sigma_1, \ldots, \sigma_{N_x})$, where σ's are singular values and are optimized by PSO. According to PSO, W_{res} and S have the same singular values. Therefore, once the σ's are optimized, the reservoir weight matrix W_{res} constructed by Equation 3.12 yields the optimized reservoir. Recall that max(σ)<1 can maintain the echo state property.

In ESN, for the lth examinee, we obtain the state matrix $H^l(t) = \left[x^l(t) : \left(\boldsymbol{\alpha}^l(t)\right)^T\right] \in \mathbb{R}^{(1+N_x)N_u}$ as a vertical concatenation of state vector and input vector, then keep the last output summary of information $H^l(t)$ and calculate the average of each column to get a vector $\boldsymbol{h}^l \in \mathbb{R}^(N_u)$, which is the representation vector for the lth examinee. Given the output $y^l(t) = \boldsymbol{\alpha}^l(t) \in \mathbb{R}^(N_u)$ (because the output is set to be equal to the input in order to realize vector recovery), then the output space as $\mathcal{L}^l = span(\boldsymbol{\alpha}^l(1), \cdots, \boldsymbol{\alpha}^l(t))$, so the Gram-Schmidt process can be applied to yield Equation 3.13

$$\begin{cases} \boldsymbol{\zeta}_i^l = \boldsymbol{\alpha}^l(i) - \sum_{j=1}^{i-1} < \boldsymbol{\alpha}^l(i), \boldsymbol{\xi}_i^l > \boldsymbol{\xi}_i^l, i \in [1, t] \\ \boldsymbol{\xi}_i^l = \frac{\boldsymbol{\zeta}_i^l}{|\boldsymbol{\zeta}|}, i \in [1, t] \end{cases} \tag{3.13}$$

where the $< \boldsymbol{\alpha}^l(i), \boldsymbol{\xi}_i^l >$ indicates the inner product of two vectors $\boldsymbol{\alpha}^l(i)$ and $\boldsymbol{\xi}_i^l$. Then the orthonormal vectors $\boldsymbol{\xi}_i^l = \left(\boldsymbol{\xi}_1^l, \cdots, \boldsymbol{\xi}_t^l\right)$ are the basis of the space $\mathcal{L}^l$. Given the representation vector learned at the kth time point in the PSO as $\boldsymbol{e}_k^l = \boldsymbol{h}^l$, the distance between the representation vector and space $\mathcal{L}^l$ is defined in Equation 3.14.

$$d(\boldsymbol{e}_k^l, \mathcal{L}^l) = \sqrt{|\boldsymbol{e}_k^l| - \sum_{i=1}^{t} < \boldsymbol{e}_k^l, \boldsymbol{\xi}_i^l >} \tag{3.14}$$

The distance reflects the distance between the latent representation and desired output vector space. Therefore, a smaller distance indicates that a better latent representation was extracted. For all the examinees, the fitness function can be defined as in Equation 3.15.

$$fitness(k) = \sqrt{\sum_{l=1}^{L} \left(d(\boldsymbol{e}_k^l, \mathcal{L}^l)\right)^2} \tag{3.15}$$

Different reservoir sizes are attempted in this chapter. They are 500, 1000, 2000, 3000, 4000, and 5000, respectively. This whole process given by the SRM can be described as in Table 3.2.

3.3 Design of Experiment

3.3.1 Experiment purposes and controlled parameters

This experiment design considers employing two simulation studies to demonstrate the performance of SMR. The simulations have two purposes. The first is that the simulation can be used to mimic examinees' action sequences for an item, and the second purpose is to show that the features extracted from the simulated action sequences contain useful information about examinees. In the simulation,

Table 3.2: The process of a simple ESN applied on the MC item process data

Process	Step
1	For each of the reservoir sizes $(500, 1000, 2000, 3000, 4000, 5000)$: Initialize the particles, each particle stands a single solution (a set of singular values) to the reservoir weight matrix.
2	Optimize the particles by the following steps: 1. For each particle, calculate the fitness value by the following steps: (a) Construct the reservoir weight matrix by Equation 3.12; (b) Calculate the feature vector $\boldsymbol{h}^l \in \mathcal{R}^{N_u}$ for the lth examinee by Equations 3.9, 3.10, and 3.12; (c) Apply the Gram-Schmidt process to the output space $\mathcal{L}^l = span\left(\boldsymbol{u}^l(1), \cdots, \boldsymbol{u}^l(t)\right)$ for the lth examinee to yield orthonormal vectors $\boldsymbol{u}_i^l = (\boldsymbol{\xi}_1^l, \cdots, \boldsymbol{\xi}_t^l)$ by Equation 3.13; (d) Calculate the distance between the feature vector $\boldsymbol{h}^l$ and space $\mathcal{L}^l$ by Equation 3.14 for the lth examinee; (e) Calculate the fitness value by Equation 3.15. 2. Record the personal best solution and the global best solution. 3. Update the particles according to Equations 3.9 and 3.10. 4. Repeat until the termination or convergence condition was satisfied.
3	Record the optimal solution.
4	Extract the feature vector $\boldsymbol{h}^l \in \mathcal{R}^{N_u}$ for the lth examinee using steps 2 to 4 in Table 3.1.

several sequences of actions to an item were generated and the proposed SMR model was applied to the simulated response process for extracting features.

In real test situations, many factors are associated with the process data such as the sample size and the total number of unique actions for a single item. As shown in Table 3.3, two factors are explored in generating action sequences, the total number of unique actions n_z and the total number of examinees l. For each of the factors, three levels are considered and each level will be selected sequentially to generate

data in the corresponding simulation scenario. For example, the combination of $n_z = 10$ and $l = 150$ in simulated data indicates that each of the 150 examines may respond to an item with an unlimited number of actions but each action is selected from 10 unique actions. One advantage given by the SRM is the adjustable reservoir weight matrix that is based on PSO and SVD. Therefore, six different levels of reservoir size ($N_x = 500, 1000, 2000, 3000, 4000,$ and 5000) are employed when simulating the SRM to test the effects caused by different reservoir sizes. In the learning process, determining how many features should be extracted by the SRM model for each data set is an exploratory process, which indicates a set of pre-defined numbers that will be used. The SRM will pick up each of the pre-defined numbers to see which one will produce the most accurate results. In this study, the SRM will select the optimal feature number for the process data from the list of $(25, 50, 75, 100, 125, 150, 175, 200)$.

Table 3.3: Factors and levels that are used for generating action sequences in the simulation

Factor	Level and value
The total number of unique actions (n_z)	0: 10
	1: 25
	2: 50
The total number of examinees (l)	0: 150
	1: 1,500
	2: 3,000

3.3.2 Markov chain

For the sake of simulating sequences and extracting features, and inspired by the work of X. Tang et al. (2020) and X. Tang et al. (2021), action sequences were generated by the Markov chain (Athreya et al., 1996) in this chapter. Markov chain is a stochastic model that describes a sequence of actions where the probability of each action simply depends on its previous action. Analyzing behavior processes with the Markov chain has been reported in the literature (e.g., D. Kaplan, 2008; Weingart et al., 1999). A Markov

chain has also been used with process data, such as generating log actions and associated timestamps using the Markov chain (X. Tang et al., 2020; X. Tang et al., 2021). In this way, the Markov chain can simulate the probability associated with a sequence of actions occurring based on the previous action. It was adapted to simulate the process sequence in this chapter.

For zth item, suppose there exist a total of n_z possible unique actions which are denoted as $\boldsymbol{S} = (s_1^z, \ldots, s_{n_z}^z)$, i.e., a total of n_z possible unique actions. In addition, let s_1 indicate the start action and let s_{n_z} indicate the end action for the zth item. Therefore, all examinees' action sequences start from s_1 and end at s_{n_z}. For all the n_z actions in a process, Markov chain will transit from one action to another, and given an action a_t at a particular moment t, the probability of making the next transition a_{t+1} will only depend on the action at the given time t. In this way, a_{t+1} is one of the n_z actions that the process can transit to. In the Markov chain, the probability of transiting from an action a_i to another a_{i+1} is defined by a square transition matrix P as shown in Equation 3.16.

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n_z} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n_z} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n_z,1} & p_{n_z,2} & \cdots & p_{n_z,n_z} \end{bmatrix} \tag{3.16}$$

This matrix $P = [p_{i,j}]_{1 \leq i,j \leq n_z}$ indicates the probability from one action a_i to another $a_j (i < j)$, and $p_{i,j} = P(a_j = a_{t+1} | a_i = a_t)$, and serves as the guiding rule of employing actions. Therefore, examinees' action sequences that were generated from the same Markov chain are believed to have a common latent connection and to have similar test behavior patterns since they have the same guiding rule.

Since s_1 indicates the start action and s_{n_z} indicates the end action for the zth item, two rules should be applied. First, the probability of transitioning from any other action to s_1 is 0 (i.e., there is no transitioning backward to the start action). Second, probabilities of transitioning from s_{n_z} to any other actions are 0, but the probability of transitioning from s_{n_z} to itself is 1. That means, in both transition matrices, $P_{i,1} = 0$ for any i, $P_{n_z,i} = 0$ for $i = 1, \ldots, n_z - 1$, and $P_{n_z,n_z} = 1$. The two matrices have the following format in Equation 3.17.

$$P = \begin{bmatrix} 0 & \cdots & p_{1,n_z} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix} \tag{3.17}$$

3.3.3 Simulation study I: group classification based on action sequences

This simulation considers the utilization of SMR on the problem of group classification. Three latent groups were employed in simulation study I, so three latent Markov chains with corresponding Markov matrices were generated. Denote the upper right sub-matrix with dimension $(n_z - 1) \times (n_z - 1)$ in Equation 3.17 as P', then the objective of the first simulation is to generate three matrices $P'^{(1)}$, $P'^{(2)}$, and $P'^{(3)}$ in order to generate three Markov groups.

First, three uniform matrices $U^{(1)}$, $U^{(2)}$, and $U^{(3)}$ each with dimension $(n_z - 1) \times (n_z - 1)$, were generated. Denote elements of the three uniform matrices as $u_{i,j}^{(1)}$, $u_{i,j}^{(2)}$, and $u_{i,j}^{(3)}$ for $U^{(1)}$, $U^{(2)}$, and $U^{(3)}$, respectively. All elements were generated independently from a uniform distribution $U(-15, 15)$. Then, three matrices $P'^{(1)} = \left(p_{i,j}^{(1)}\right)_{1 \le i,j \le (n_z-1)}$, $P'^{(2)} = \left(p_{i,j}^{(2)}\right)_{1 \le i,j \le (n_z-1)}$, and $P'^{(3)} = \left(p_{i,j}^{(3)}\right)_{1 \le i,j \le (n_z-1)}$

were generated using the following equations in Equation 3.18.

$$\begin{cases} p_{i,j}^{(1)} = \frac{\exp u_{i,j}^{(1)}}{\sum_{j=1}^{n_k-1} \exp u_{i,j}^{(1)}} \\ p_{i,j}^{(2)} = \frac{\exp u_{i,j}^{(2)}}{\sum_{j=1}^{n_k-1} \exp u_{i,j}^{(2)}} \\ p_{i,j}^{(3)} = \frac{\exp u_{i,j}^{(3)}}{\sum_{j=1}^{n_k-1} \exp u_{i,j}^{(3)}} \end{cases} \tag{3.18}$$

where $p_(i,j)^{(1)}$, $p_(i,j)^{(2)}$, and $p_(i,j)^{(3)}$ are elements of the three matrices $P'^{(1)}$, $P'^{(2)}$, and $P'^{(3)}$. Given these three matrices, three Markov matrices were formed using Equation 3.17 and therefore three different guiding rules were generated.

Each sequence of actions is generated by one of the three Markov matrices, and all sequences will be analyzed by the SRM. The feature vectors $\boldsymbol{h}^l \in \mathcal{R}^{N_u}$ extracted from the sequences by the SRM are used to classify each examinee's latent group with a generalized logit model. For comparison, the average of each input vector $\boldsymbol{a}^l$ is computed as baseline features.

3.3.4 Simulation study II: latent trait prediction based on action sequences

This simulation considers the utilization of SMR on the problem of latent trait prediction. Compared with simulation I, each of the l action sequences in simulation II was generated from a unique Markov chain, but all l chains were associated with a common uniform matrix $U^{(4)}$, given the studies of X. Tang et al. (2020) and X. Tang et al. (2021). Similarly, in each Markov matrix, $P_{i,1} = 0$ for any i, $P_{n_z,i} = 0$ for $i = 1, \cdots, n_z - 1$, and $P_{n_z,n_z} = 1$ as shown in Equation 3.17.

First, examinees' latent abilities, $\theta_1, \cdots, \theta_l$, were randomly generated from a normal distribution $N(0,1)$. Then the uniform matrix $U^{(4)}$, with element $u_{i,j}^{(4)}$ and dimension $(n_z - 1) \times (n_z - 1)$, was

generated from a uniform distribution $U(-15, 15)$ as was done in simulation study I. We again denote the upper right sub-matrix with dimension $(n_z - 1) \times (n_z - 1)$ in Equation 3.15 as $P'^{(l)}$. These matrices $P'^{(l)}$ were generated by both the latent abilities θ_1 and the common uniform matrix $U^{(4)}$. That is, for each examinee, the unique matrix $P'^{(l)} = (p_{i,j}^{(l)})_{1 \le i,j \le (n_z - 1)}$ was generated using Equation 3.19

$$p_{i,j}^{(l)} = \frac{\exp \theta_l u_{i,j}^{(4)}}{\sum_{j=1}^{n_k - 1} \exp \theta_l u_{i,j}^{(4)}} \tag{3.19}$$

where $p_{i,j}^{(l)}$ represents elements of the unique matrix $P'^{(l)}$. Given these matrices, l different Markov matrices were formed using Equation 3.17 as we did in simulation study I.

Since each sequence of actions is generated by its unique Markov matrix, the features extracted by applying the SRM on all sequences are used to predict examinees' latent abilities θ_l. Similarly, the average of each input vector $\boldsymbol{a}^l$ is used as baseline features for predicting examinees' latent abilities.

3.3.5 Evaluation metrics

Accuracy is used to demonstrate the classification ability of the extracted features in Simulation I. Table 3.4 shows a confusion matrix example, with each entry showing the number of results.

Table 3.4: Confusion matrix for classification with three categories.

		Model predicted group		
		1	2	3
Actual Markov Group	1	N_{11}	N_{12}	N_{13}
	2	N_{21}	N_{22}	N_{23}
	3	N_{31}	N_{32}	N_{33}

Given the numbers, the classification accuracy is defined in Equation 3.20. In this table, each entry shows the number of results corresponding to the actual group of the row and the model predicted group of the column.

$$CA = \frac{\sum N_{i=j}}{\sum (N_{i=j} + N_{i \neq j})} \tag{3.20}$$

Prediction accuracy of the latent trait estimates was evaluated using the root mean square error (RMSE), between the generating and estimated parameters. Assuming we have L examinees, the RMSE for the latent trait estimates is calculated by Equation 3.21

$$RMSE = \sqrt{\frac{1}{L} \sum_{l=1}^{L} (\hat{\theta}_l - \theta_l)^2} \tag{3.21}$$

where θ_l stands for the generating ability estimation for lth examinee, and $\hat{\theta}_l$ is the estimated parameter for lth examinee using LASSO (least absolute shrinkage and selection operator) regression. In the simulation studies, each simulation condition was replicated 50 times, therefore, average CA and average RMSE are obtained by taking the average to the CA and RMSE values from these replications.

LASSO regression selects variables and regularizes the model to enhance the prediction accuracy and interpretability of the model by adding a penalty term in the error function. The LASSO coefficients, $\hat{\beta}_\lambda$, minimize the quantity in Equation 3.22

$$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{3.22}$$

where$\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2$ is the sum of squared residuals given by the regression, λ is the LASSO constant, and β_j is the L_1 norm of the coefficient vector β_j.

3.4 Results of Simulation Studies

3.4.1 Results of study I

Applying the SRM to the simulated dataset yields a feature matrix with dimensions $l \times N_u$, in which each row represents an examinee and each column is one feature. We first conducted a PCA analysis on one of the feature matrices which was extracted for the condition of $l = 3000$ and $n_z = 10$ (i.e., $3,000$ examinees and 10 features). Generally, for the data set with a total of p extracted principal components, the eigenvalue of each principal component would be decreasing, so it is common to retain the first k principal components (Dunteman, 1989). The selection of k is usually based on two objectives. The first is that k should be as small as possible for the sake of having the simplest component interpretation. For example, if we can explain a large portion of the overall data variation with the first two principal components, then use of the first two principal components alone would simplify our description of the data. The second objective is that the portion of the overall data variation explained by the selected principal components should be as large as possible in order to reduce the loss of information. That is, the ratio between the sum of the eigenvalues of the selected principal components and the sum of the eigenvalues of all selected principal components should be as close to 1 as possible. This is shown in Equation 3.24

$$\frac{\sum_1^k \lambda_i}{\sum_1^p \lambda_i} \approx 1 \tag{3.23}$$

where $\lambda_i (i = 1, \ldots, p)$ indicates the eigenvalue of each principal component.

The first two principal components for each examinee were selected since they would account for 75.3% of the total variation (i.e., 1st principal component: 59.6%, 2nd principal component: 15.7%).

Figure 3.3 plots these two components. In this figure, the x-axis and y-axis are the first and second principal components, respectively. Each dot represents an examinee. The three groups indicate which group each examinee was generated from. It appears clear that three different groups can be distinguished by the features that were extracted from the generated action sequence.

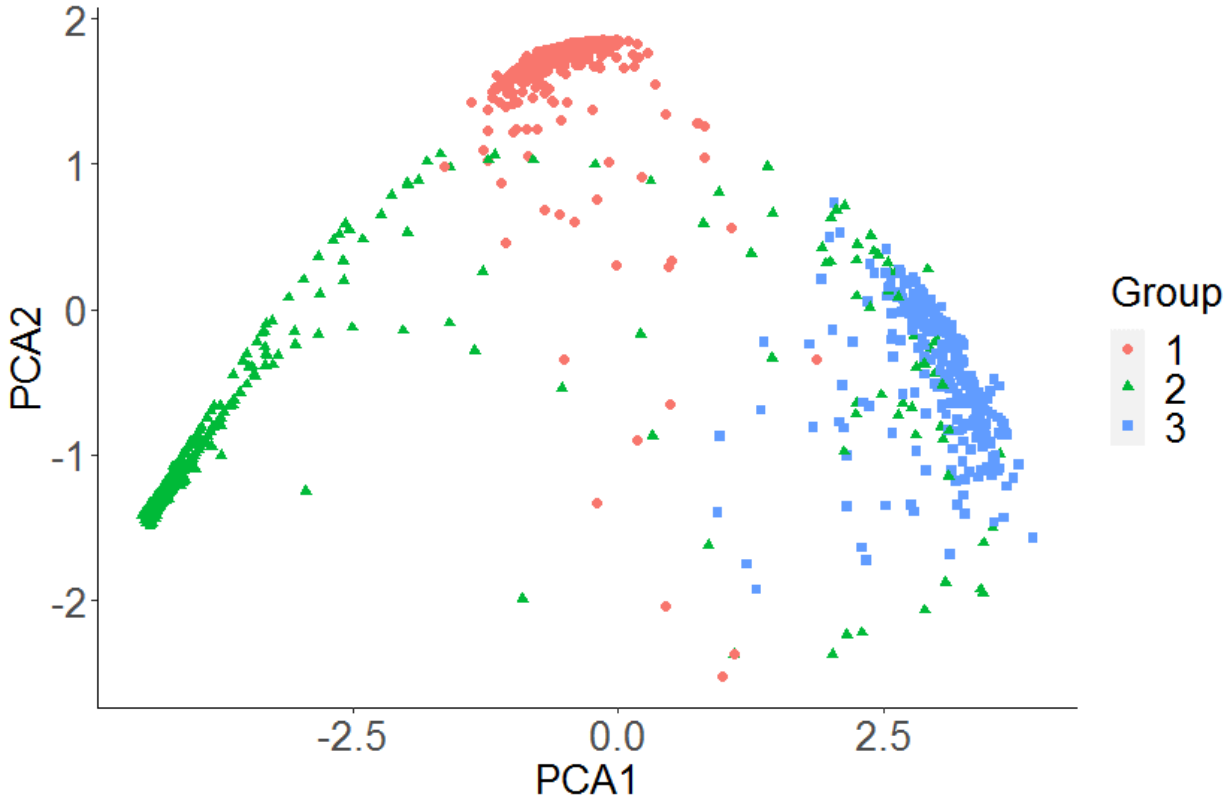

Figure 3.3: Principal component analysis of the features extracted by SRM from Simulation Study I with $l = 3000$ and $n_z = 10$

The extracted SRM features and baseline features are used to predict the group classification under each simulation condition, respectively. Table 3.5 reports the average of the best classification accuracies associated with standard deviations in the parenthesis yielded by the most optimal reservoir size. The smallest and highest classification accuracies yielded by using the SRM features are 0.825 and 0.902, respectively, while values from the baseline features are 0.391 and 0.454, respectively. It can be seen from

the table that first, the accuracy yielded by using the SRM features is much higher than that yielded by using the baseline features under each of the conditions. This indicates that the features extracted by SRM may successfully contain sufficient information about the different Markov groups. Second, it can be observed that, when the sample size increases, the accuracy produced by using the SRM features increases as well. This suggests that by including more cases in the SRM, it is possible to extract features that contain more sequence information. The number of unique actions does not appear to have a consistent effect. We don't find, however, that increasing the number of unique actions either increases or decreases the accuracy. For example, with $l = 150$, when n_z increases from 10 to 25 and further to 50, the accuracy changes only happed at the second or third decimal.

Table 3.5: Average best group classification accuracy with the standard deviations by the extracted features and baseline features using the multinomial logistic regression

l	n	SRM features	Baseline features
150	10	0.825 (0.074)	0.391 (0.137)
	25	0.836 (0.102)	0.400 (0.098)
	50	0.829 (0.096)	0.398 (0.115)
1500	10	0.851 (0.068)	0.427 (0.168)
	25	0.862 (0.057)	0.414 (0.151)
	50	0.865 (0.062)	0.401 (0.099)
3000	10	0.891 (0.059)	0.431 (0.103)
	25	0.902 (0.047)	0.454 (0.077)
	50	0.900 (0.051)	0.448 (0.102)

Classification accuracies for the simulation study are plotted in Figure 3.4 for 500 to $5,000$ examinees. The plots in Figure 3.4 show that average classification accuracy increased with an increase in the number of examinees for each of three different numbers of features. The red lines are changes of the average classification accuracies, and the segments above and below each point are standard deviations. The general trend observed in the figure shows that a larger reservoir size (i.e., a larger number of examinees) increased the accuracy of classifying each generated sequence.

The best accuracies under each combination were all observed when the reservoir size was 5000. In addition, when the reservoir size changed from 500 to 1000, the accuracy increased more rapidly, and the standard deviation yielded by the reservoir of 500 was larger than the other sizes. This suggests that a reservoir of 500 may provide a less classification accuracy than the desired level of accuracy. The results from an increase in sample size also suggests that including more cases in the sample should improve the classification accuracy. Finally, the patterns in each of the plots in Figure 3.4 appear roughly the same suggesting that the number of unique actions, n_z, does not have much differential impact on the accuracy of feature extraction.

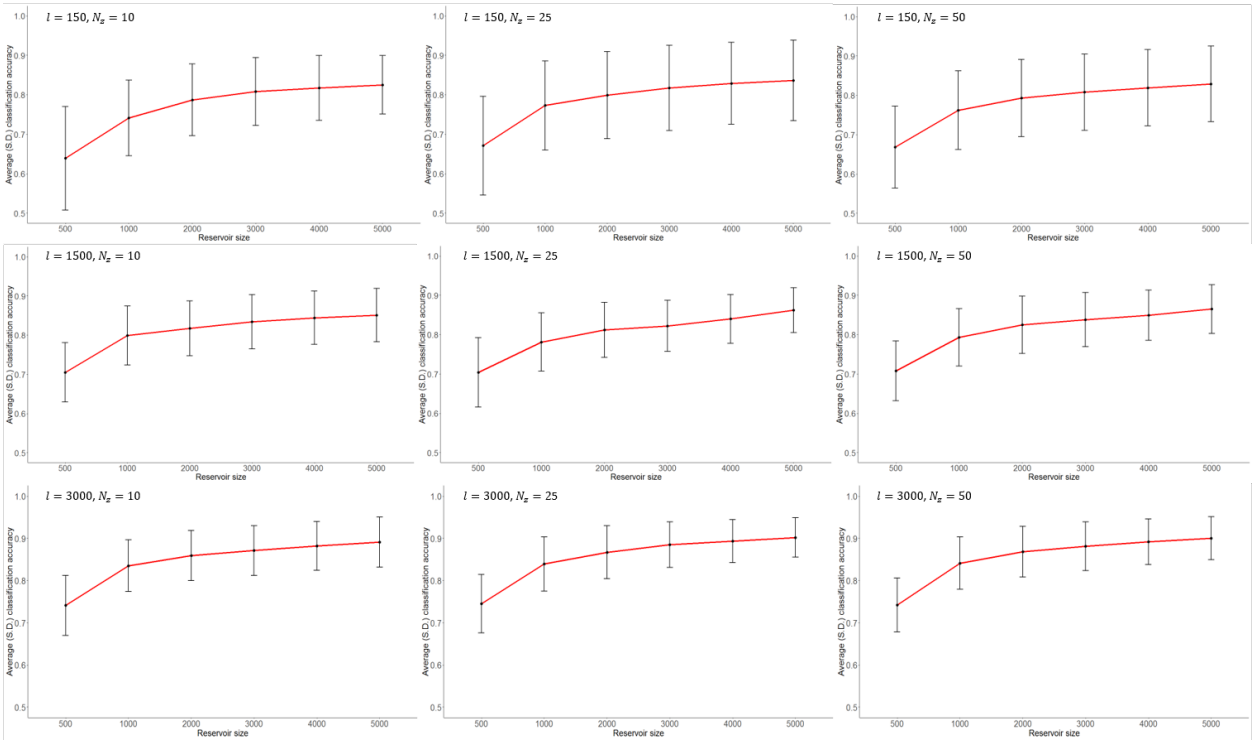

Figure 3.4: Average group classification accuracy with standard deviation yielded by an increasing reservoir for each combination of Number of Examinees and Number of Features

3.4.2 Results of study II

After applying the SRM on the simulated dataset in Simulation Study II, we obtained a feature matrix with dimension $l \times N_u$, in which each row represents an examinee and each column is one feature. The PCA analysis was applied to the feature matrix that was extracted for the $l = 3000$ and $n_z = 10$ condition.

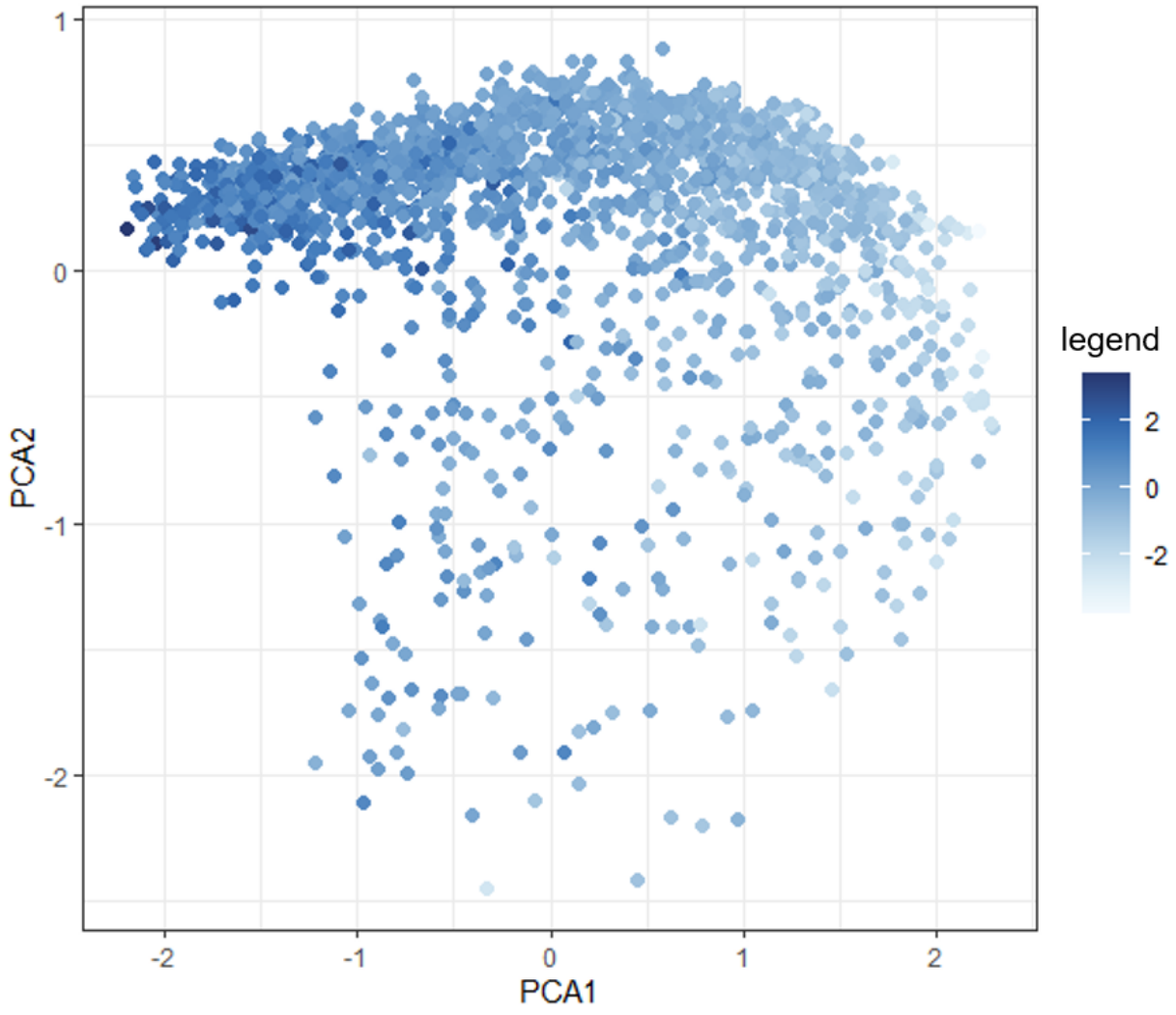

Figure 3.5: Principal component analysis to the features extracted by SRM from Simulation II with $l = 3000$ and $n_z = 10$

As shown in Study I, in Figure 3.5 we again show a plot of the first and second principal components for each examinee from the PCA analysis. In this figure, the x-axis and y-axis are the first and second principal components, respectively. Each dot represents an examinee. The legend on the right of the figure indicates the generated θ for each examinee. The darker dots indicate a higher value θ and the lighter dots indicate a lower value θ. It is interesting to see a clear fading pattern from higher ability to lower ability levels by the two principal components analyzed from the features that were extracted from the generated action sequence. In another word, the examinees located closer to each other have similar latent trait levels, and their ability information can be well represented and compressed in the features extracted by SRM.

The extracted SRM features and baseline features are used to predict the latent abilities θ_j under each simulation condition, respectively. Table 3.6 reports the average of the RMSEs associated with their stand errors yielded by the most optimal reservoir size. The smallest and highest RMSEs yielded using the baseline features are 0.896 and 1.636, respectively. RMSE values from using the SRM features were only 0.301 and 0.587, respectively. This is similar to what was obtained in Simulation Study I. First, the RMSEs obtained by using the SRM features were much lower than those from using the baseline features in each of the conditions. Second, using a larger sample size appeared to reduce the prediction error. That is, the RMSE produced by using the SRM features decreased with an increase in simulated examinees. Compared with the group classification, however, a larger number of n_z seems to have a negative on the latent trait prediction. It can be seen that, when the number of n_z increased, the RMSEs increased under each of the three levels of l. For example, when $l = 150$, for $n_z = 10$, 25, and 50, the regression RMSEs yielded by using the extracted features were 0.481, 0.532, and 0.587, respectively. This may indicate that

the extracted SRM features could have a better representation of the sequence and latent abilities, if an item has fewer unique actions for examinees.

Table 3.6: Average best latent trait RMSEs with standard deviations by extracted features and baseline features using the LASSO regression

l	n	SRM features	Baseline features
	10	0.481 (0.104)	1.048 (0.187)
150	25	0.532 (0.115)	1.397 (0.214)
	50	0.587 (0.138)	1.636 (0.296)
	10	0.408 (0.045)	0.965 (0.095)
1500	25	0.465 (0.085)	0.999 (0.101)
	50	0.496 (0.117)	0.994 (0.099)
	10	0.301 (0.036)	0.896 (0.083)
3000	25	0.358 (0.041)	0.925 (0.094)
	50	0.383 (0.042)	0.913 (0.091)

The RMSEs in Figure 3.6 indicate the change of average RMSEs and standard deviations along with this training process. The general trend observed from the figure shows that a larger reservoir size can better predict the latent trait for each generated sequence. Similarly, the lowest RMSEs were all obtained for the reservoir size of 5000. The highest RMSEs were obtained when the reservoir size was lower than 1000. The standard errors yielded by the reservoir of 500 were also larger than the other reservoir sizes. This further suggests that the use of at least a reservoir size of 1000 may be able to extract useful features. Based on the increase of sample size, we can see that by incorporating larger samples in the model, the features may be able to extract more information and a subsequently higher regression prediction. From this figure, a higher number of unique actions n_z seems to hurt the feature extraction. This is because, when the number of actions changes from 10 to 25 and from 25 to 50, the prediction errors increased under each condition.

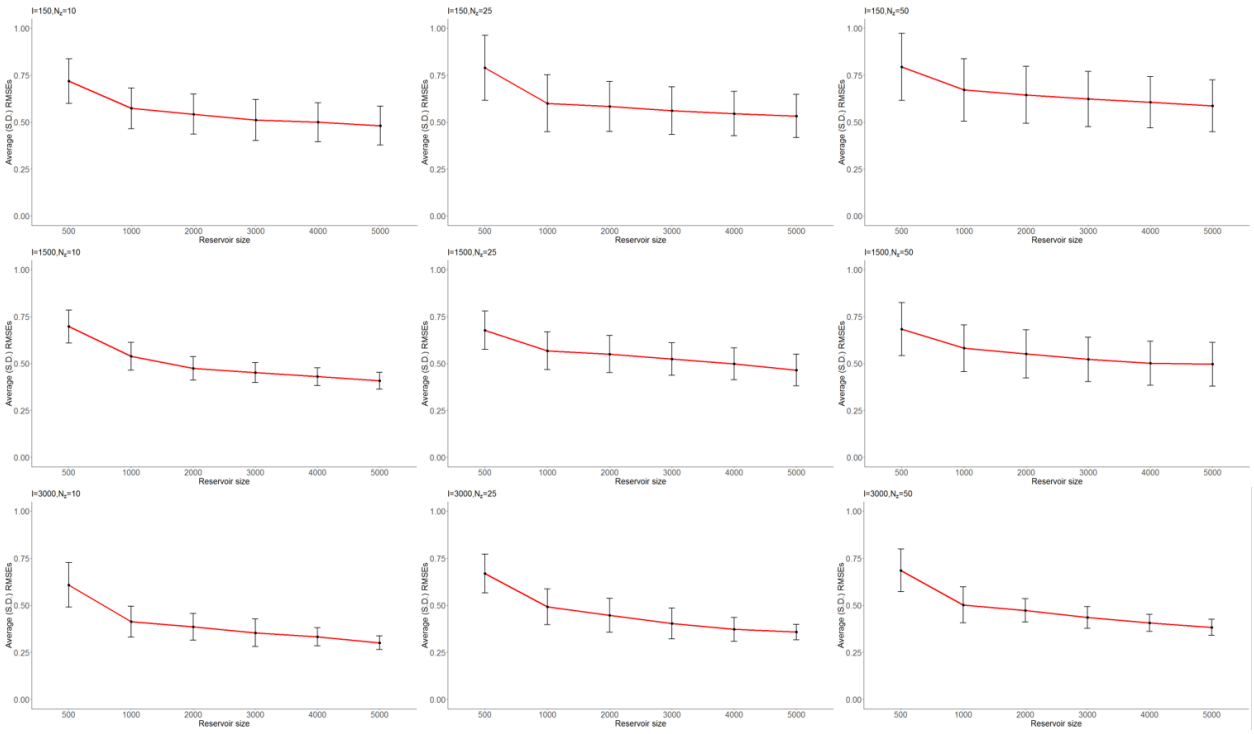

Figure 3.6: Average latent trait RMSEs with standard deviations yielded by an increasing reservoir for each combination of two factors

3.5 An Exploratory Study Using both Process Data and Response Data

In the two simulation studies above, two types of datasets were simulated. The use of extracted features in group classification and the variable prediction was shown for each type. In the second simulation study, we found that the features extracted from the process data could be used to recover the latent trait with a low RMSE when the sample size and the reservoir size were large. However, the use of the features is still separated from the use of response data such as the commonly used dichotomous or polytomous data in

IRT models. That is, we didn't include the response score patterns when constructing the classification and regression models with the extracted features in Study I and II. In this section, we compare two conditions, one is the use of both the process data plus the response data and the other is the use of only the response data. A new simulation study is described below using the Rasch model (Rasch, 1966; Rasch, 1993) to generate item responses and using a Markov chain to generate associated action sequences.

3.5.1 Rasch model

Among all latent trait models proposed for examinees' ability measurement, the Rasch model has the simplest parameterization. It has one ability parameter θ_l for each examinee and one difficulty parameter b_z for each item. Rasch model has been used in many assessments with dichotomous item response data to show the positions of both examinees and items on the latent variable scale (Engelhard Jr, 2013). The model assumes that all the items have identical item discriminations of 1. The model takes the form as shown in Equation 3.24

$$P_z(\theta_l) = \frac{1}{1 + \exp -(\theta_l - b_z)} \tag{3.24}$$

where $P_z(\theta_l)$ is the probability that an examinee with trait level θ_l can be expected to answer correctly that item with a difficulty level of b_z.

3.5.2 Design of the experiment

Based on the preliminary conclusion from the previous simulation studies, we use $l = 3000$, $n_z = 10$, and a reservoir size of 5000. Six test lengths were generated in this study : $z = 5$, 15, 25, 35, 45, and 55 items. For each of the six test length conditions, we assume that all items have the same number of

unique actions, and different examinees will have different guiding rules, but an examinee will use the same guiding rule for all items. Here the guiding rule refers to the transition probability among actions for each examinee (i.e., the Markov matrix). Therefore, these two assumptions indicate that, first, the numbers of unique actions provided for all items are the same but each individual item's unique actions and their meanings could be different. For example, for a test with 5 items, each item may have 10 unique actions for examinees to use during the test, but the 10 actions for item 1 could be different from those in item 2 except for the actions of "begin item" and "end item". Second, by assuming that an examinee will use the same guiding rule for each item, we make sure that each simulated individual examinee may respond to items following a consistent testing behavioral rule.

For each test length condition, examinees' latent abilities, $\theta_1, \cdots, \theta_{3000}$, and the item difficulty parameters b_z were randomly generated from a normal distribution $N(0, 1)$. By employing the Rasch model in Equation 3.22, the response matrix can be generated, and each examinee's response vector can be represented as π^l. Then, by using the same latent abilities, we generated 3000 sequences for each item using the method of simulation study II. That is, for each of the z items, each of the 3000 sequences was generated from a unique Markov chain, and all 3000 chains were associated with a common uniform matrix as we saw in Simulation Study II (i.e., section 3.3.4 the common uniform matrix $U^{(4)}$).

3.5.3 Linear model and fit indices

For each item, the features are extracted by applying the SRM on its 3000 sequences. For all items, their feature matrices will be column concatenated together such that $\boldsymbol{h} = (\boldsymbol{h}_1 : \cdots : \boldsymbol{h}_z)$. The features are used together with the generated responses to predict examinees' latent abilities in a linear model as

specified below, in Equation 3.25.

$$\boldsymbol{\theta} = X\beta + \epsilon \tag{3.25}$$

where $X = [\mathbf{1} : \boldsymbol{\pi} : \boldsymbol{h}]$ means the column concatenation of vector $\mathbf{1}$, response matrix $\boldsymbol{\pi}$, and feature matrix $\boldsymbol{h}$. LASSO regression is used for selecting useful features for training this linear model. LASSO selects meaningful features and adds them as additional exploratory variables into the model with only the responses to predict the latent ability values. As a comparison, the response matrix itself is also fit into the other linear model with the matrix specification of $X = [\mathbf{1} : \boldsymbol{\pi}]$.

Denote the linear model using only responses as *RspData*, and the other linear model with both responses and features as *Rsp+ProcData*. We then can evaluate the two models by the recovery of ability and model fit indices. RMSEs are used to help indicate the accuracy of the recovery of the simulated latent ability. Model fit indices used in this study are residual standard error (RSE) and R-squared (R^2). Both are used to measure how well a regression model fits the data and predicts the latent variable. RSE measures the standard deviation of the residuals in a regression model by calculating Equation 3.26

$$RSE = \sqrt{\frac{\sum(y - \hat{y})^2}{df}} \tag{3.26}$$

The measure of R^2 represents the proportion of the variance for the predicted variable that is explained by the predictors in the regression model by computing Equation 3.27

$$R^2 = 1 - \frac{RSS}{TSS} \tag{3.27}$$

where the RSS is the residual sum of squares and TSS is the total sum of squares.

3.5.4 Results discussion

The RMSE values of the recovery of the latent ability given by the two models are shown in Table 3.7. The linear model with the addition of process data provides better ability recovery than the linear model with only the response data. In addition, by using more items in the test, two linear models both yield higher ability recovery accuracy, but the improvement based on the *Rsp+ProcData* model is smaller than the *RspData* model. For example, the ranges of RMSE values from the *Rsp+ProcData* model and *RspData* model are $(0.263, 0.399)$ and $(0.298, 0.711)$, respectively. When the number of items is small, such as 5 items, adding process data into the response data yielded a lower RMSE value than the model using only the response data. The recovery analysis confirms that the addition of process data can provide additional useful information about examinees' latent ability information.

Table 3.7: RMSEs of the latent ability recovery given by the Rsp+ProcData model and the RspData model

	Number of items in the test					
	5	15	25	35	45	55
Rsp+ProcData	0.399	0.351	0.331	0.303	0.287	0.263
RspData	0.711	0.513	0.430	0.387	0.346	0.298

Figure 3.7 plots the two model fit indices of the *Rsp+ProcData* model versus the *RspData* model. In this figure, the x-axis indicates the fit index values from the *RspData* model and the y-axis indicates the fit index values from the *Rsp+ProcData* model. The number of items used in the test is represented by increasing radius sizes in this figure, i.e, 5, 15, 25, 35, 45, and 55. Red and green colors indicate the index of RSE and R^2, respectively. The diagonal line separates an upper triangle section and a lower triangle section, and the dots on this separating line are relatively close to the line suggesting the two fit indices from the two models indicate similar levels of fit. The symbols in the upper triangle section indicate that the fit index value from the *Rsp+ProcData* model is higher than those from the *RspData* model, and the

dots in the lower triangle indicate the fit from the *Rsp+ProcData* model is lower than from the *RspData* model. Further, all R^2 values fall into the upper triangle section, while all RSE values fall into the lower triangle section.

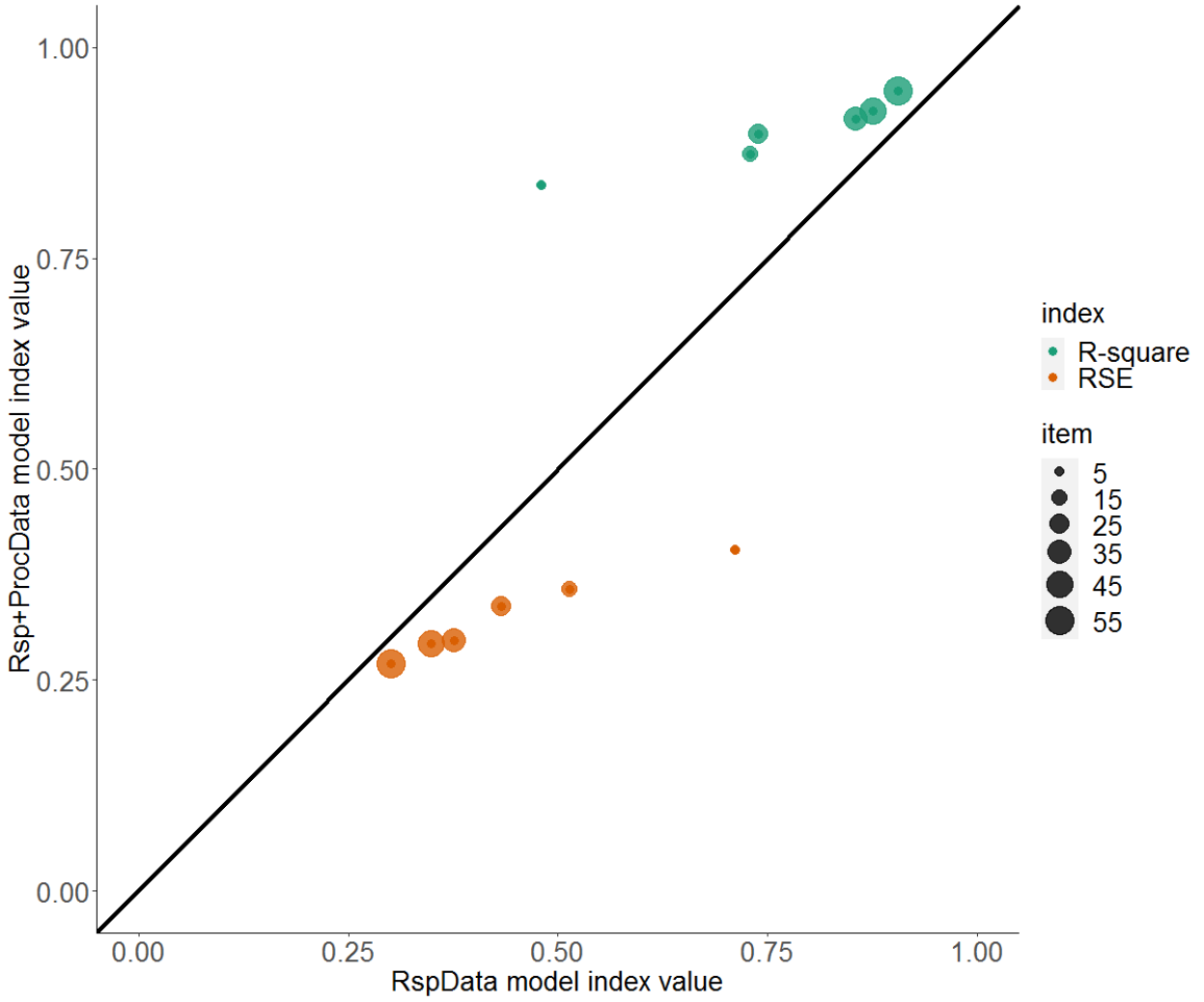

Figure 3.7: The model fit index values of the *Rsp+ProcData* model versus the *RspData* model

This figure tells us that the model fit given by the *Rsp+ProcData* model is better than by the *RspData* model. In other words, the additional information from the process data appears to result in a better fit to the linear model than the use of only response data. Further, adding more item responses in the linear

model appears to result in the symbols trending toward the line. This may indicate that the differences between the two models might diminish if more item responses are included.

3.6 Application of Sequence Reservoir Model (SRM) to the Empirical Data

In this section, an empirical dataset is analyzed to demonstrate the SRM model performance. The empirical data set contains individual event process histories that are used by examinees during the assessment. In this section, we illustrate that the extracted features have some actual meaning and can be used for predicting examinees' ability information.

This empirical dataset is from the National Assessment of Educational Progress and Educational Testing Service (https://sites.google.com/view/dataminingcompetition2019/home). It contains a deidentified compilation of action sequences made by $2,463$ examinees who took an 8th-grade mathematics test in the $2016-2017$ academic year. This mathematics test covered the domains of algebra and geometry. The items included stimulus material in a text or figural format. The assessment was digitally administered on tablet computers with keyboards. The tests contained mixed-format items such that examinees were provided with multiple-choice items, drag and drop response items, or constructed-response questions. For some items, an on-screen calculator and drawing tools were available. This enabled examinees to calculate some operations and make handwritten annotations to some questions. There was also a text-to-speech feature that allowed examinees to listen to the task materials. Therefore, the process data set of this assessment included each examinee's actions and the associated response time for each item. These actions include starting an item, clicking a response, typing something for constructed-response items,

response revisions, and use of additional tools such as text-to-speech, calculator, or drawing tool. Table 3.8 lists the names and associated interpretations of each variable in this data.

Table 3.8: Variables of the NAEP process data

Variable	Meaning
STUDENTID	Examinees' unique identification number
Block	Block number in the NAEP assessment
AccessionNumber	Item unique identification number
ItemType	Type of the item
Observable	The action that is used by the examinee at the current moment
ExtendedInfo	Additional information on the examinee action
EventTime	The timestamp of when the action was taken
EfficientlyCompletedBlockB	Examinees' efficiency level of block B

In this assessment, examinees responded to two "blocks", here we refer to them as Blocks A and B. Examinees were able to navigate between items within the same block. Block A contained 19 items and Block B contained 15 items. Each examinee had a 30-minute time limit to complete the problems in a given block. After finishing the last item in each block, a review screen was presented to examinees which indicated the end of the block. At that point, the examinee could either navigate away from the review screen back to given items to make changes, move forward to the next block, or end the test.

The nature of the exam allowed examinees to complete items at their own pace and, if so desired, to skip items. Once the 30 minutes was reached, the examinee was automatically cut off from further activities in the block, regardless of how many problems they have completed. The purpose of using these data in the analysis was to apply the SRM to the process data to extract features and to identify whether these features could be useful for helping to understand examinees' test-taking behaviors or to predict examinees' latent ability status.

3.6.1 Exploratory data analysis

A response variable was assigned to each examinee after they completed the block to determine whether an examinee efficiently finished block B. Examinees who were able to allocate a reasonable amount of time on each item and complete the block were labeled as having efficiently finished block B. On the other hand, some examinees went too fast and may not have actually carefully read the items. Responding fast, for example, may be a sign of guessing or responding based on item preknowledge (D. Wang et al., 2018). On the other hand, some examinees may respond to each item at a slower pace with the result that they miss a certain number of items due to the time limit. Examinees were identified for these two scenarios as not efficiently finishing each block.

The process data set was split into two subsets. The first subset (Subset 1) contained $1,232$ examinees' action sequences within the 30 minutes for each item in block A. Also included were the response variables and efficiency rating for each examinee on block B. The second subset (Subset 2) contains the remaining log data for the $1,232$ examinees obtained within the time limit allotted for block A. This second subset was stratified into three portions. The first portion contained 411 examinees' first 10 minutes of process data from the beginning of Block A. The second portion contained another 411 examinees' first 20 minutes of process data from the beginning of Block A. The third and last portion had the remaining 410 examinees' process data for the complete 30 minutes allotted for Block A. The structure of these two subsets and two blocks is presented in Figure 3.8.

The descriptive statistics about the two subsets are listed in Table 3.9. In this table, Subset 1 means the first subset, and from Portion 1 to Portion 3 they indicate the three portions from Subset 2. The meaning of each column is presented as follows: Total length means the total number of actions for all examinees

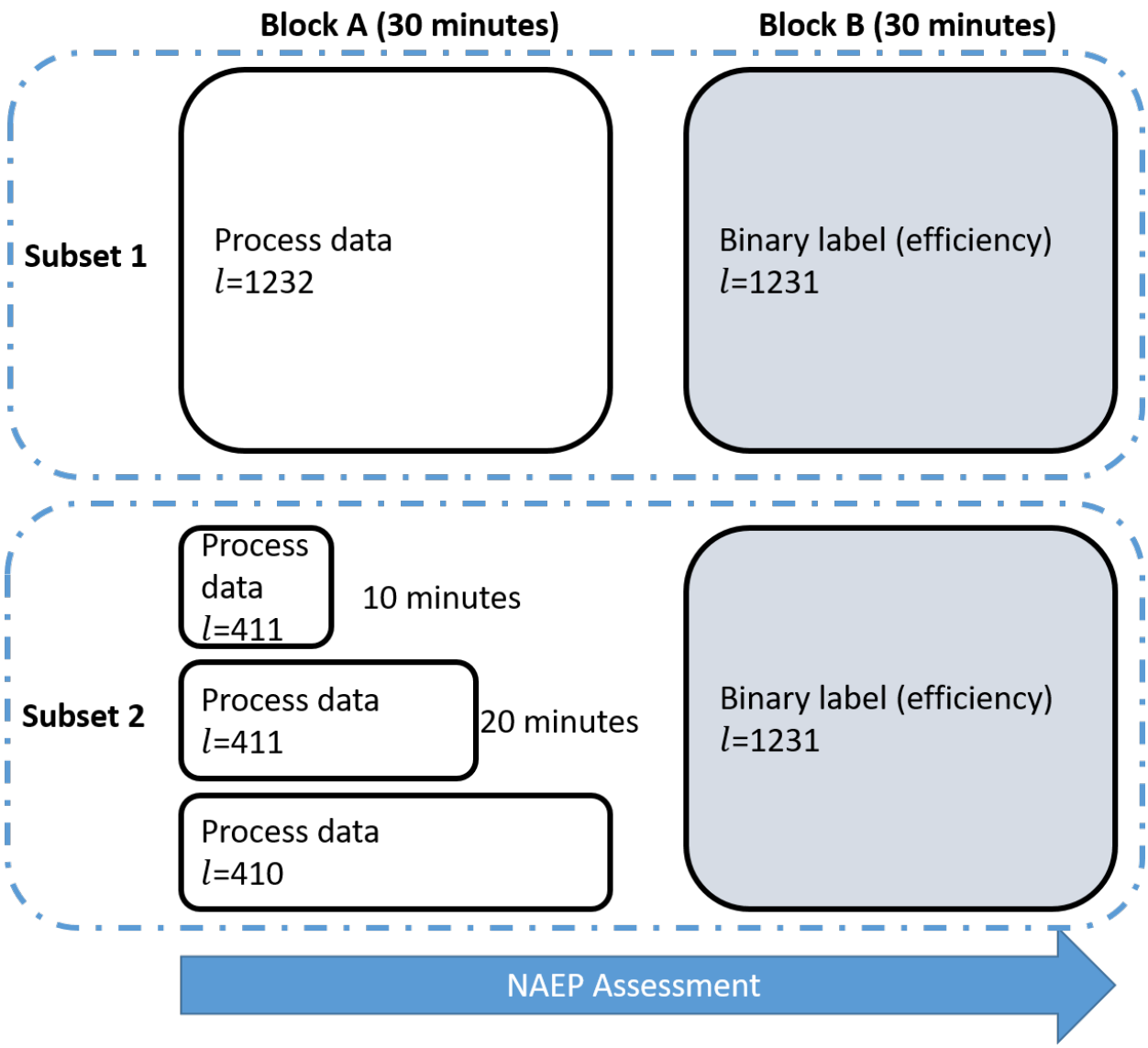

Figure 3.8: Data sets and assessment block structure

stored in each data set; Item stands for the total number of item formats in the data; Unique actions mean the number of unique actions in the log data; and Average length represents the mean action length per examinee with standard deviations in the parenthesis about the data.

Table 3.9: Descriptive statistics for the NAEP math assessment Subset 1 and Subset 2 process data

	Total length	Item	Unique actions	Average length	Efficient	Inefficient
Subset 1	438,291	10	42	356 (166)	744	488
Portion 1	47,563	9	39	116 (57)	248	163
Portion 2	110,481	9	41	269 (116)	248	163
Portion 3	143,880	10	42	351 (158)	248	162

The number of examinees falling into each efficiency level for each subset is also included in this table. For example, the numbers of examinees that completed Block B efficiently and inefficiently are 744 and 488 for Subset 1, respectively. If we sum up the two columns for the three portions of Subset 2, we find the numbers of examinees that completed Block B efficiently and inefficiently are also 744 and 488. The plots in Figure 3.9 show the action frequency rank for all actions appearing in each subset. It can be seen from the figure that the top two highest frequency actions are Draw and Math Keypress. These indicate examinees' annotations and keyboard click actions. The action frequency ranks of Subset 1 and Subset 2-portion 3 are similar to each other, since they were both 30-minute time periods, although they have different numbers of examinees.

3.6.2 Support vector machine and evaluation metrics

Extraction of process data from the feature matrices is enabled by the use of a support vector machine (SVM; Noble, 2006). The SVM is a supervised algorithm that learns either the linear or nonlinear decision boundary to classify samples with labels. For example, for the input data $\boldsymbol{x} \in \mathcal{R}^O$, the SVM can transform the input data into a newly created feature space, in order to make classifications by identifying

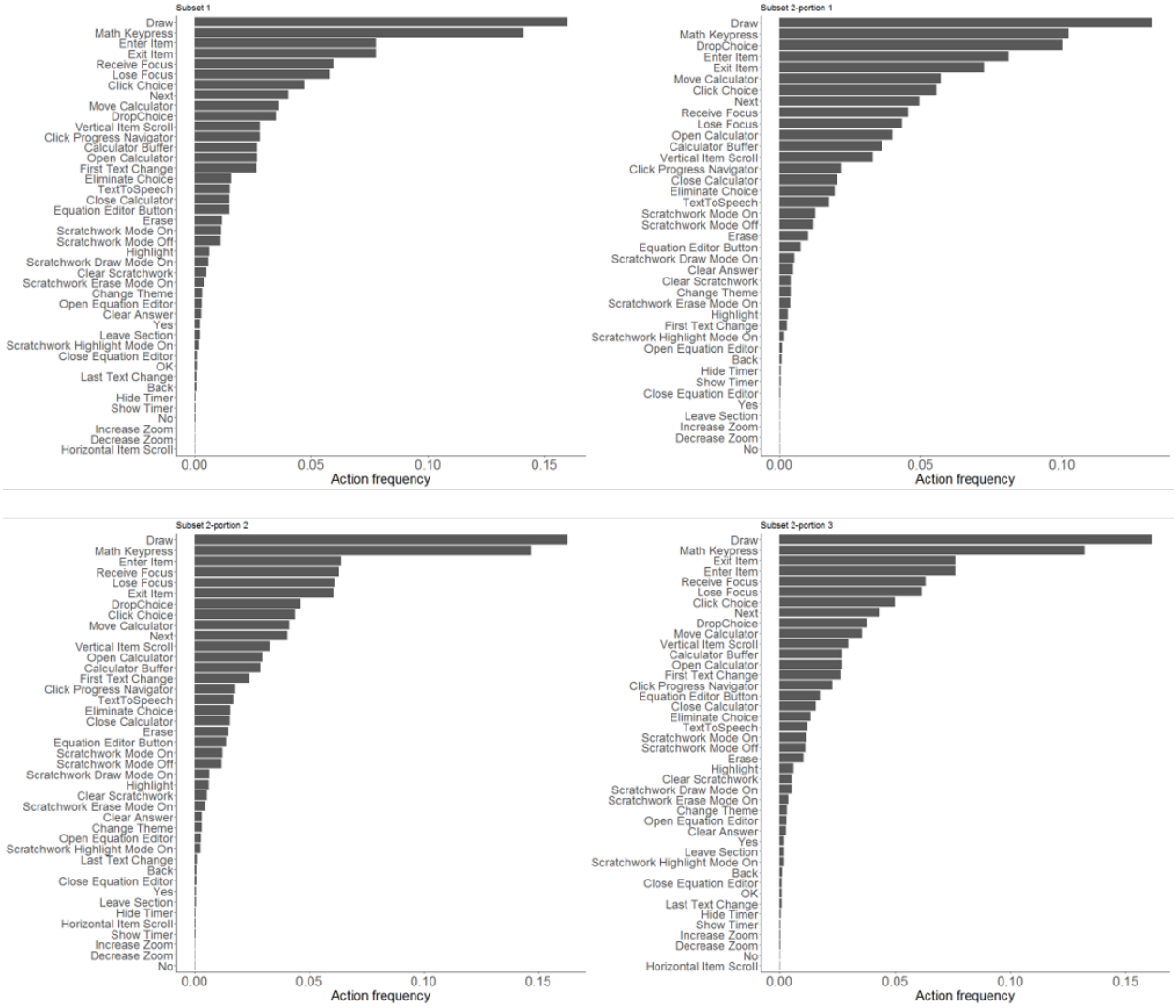

Figure 3.9: Action frequency in each of the subsets of process data

a boundary between classes based on the transformed features $\boldsymbol{f} = \phi(\boldsymbol{x}) \in \mathcal{R}^D$. Figure 3.10 describes this transformation process in which new features are created from the original data points. In this way, they provide the boundary to distinguish between classes. It is important to note that, sometimes the dimension of the created new feature space might be higher (i.e., $\mathcal{R}^O \subseteq \mathcal{R}^D$) than the original space, as shown in Figure 3.10. In such a case, it is possible to map two-dimensional data onto a three-dimensional coordinate system to achieve clear separation between the two classes, Efficient and Inefficient.

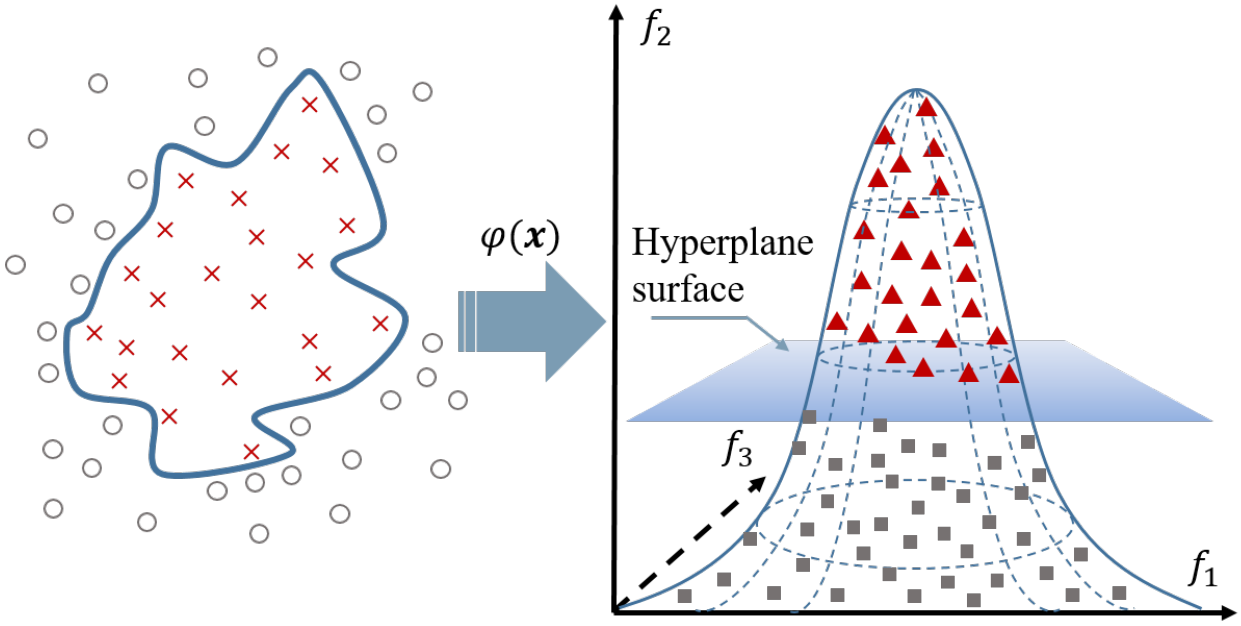

Figure 3.10: SVM transformation process

This decision boundary is a separator that divides data points into their respective classes, where the separator is referred to as a hyperplane. Figure 3.11 shows that the SVM uses the transformed features $\phi(\boldsymbol{x})$ to decide the hyperplane $\mathcal{H} : g(\phi(\boldsymbol{x})) = \mathbf{0}$ and distinguish the classes, where the transformed feature data points are indicated above the upper dotted and below the lower dashed boundaries with distances d_1

and d_2 to the hyperplane. These are the support vectors and the distances d_1 and d_2 from the hyperplane to the support vectors are called the margins.

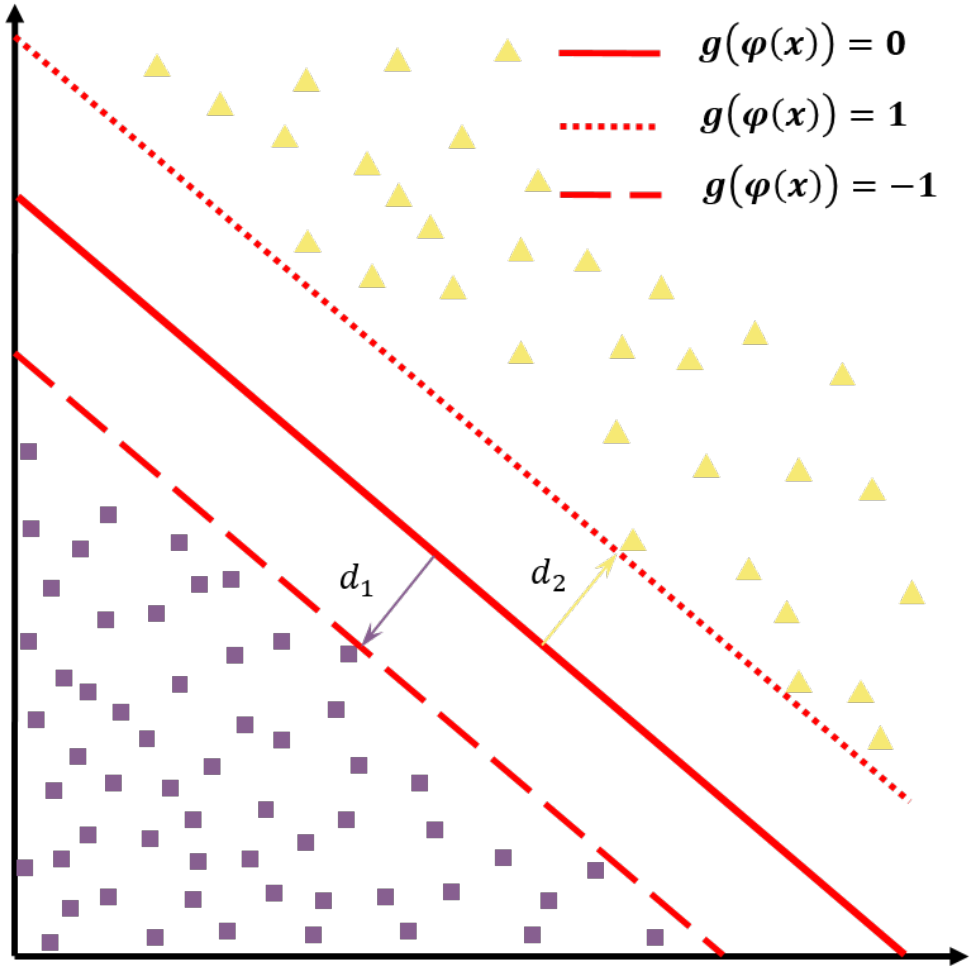

Figure 3.11: SVM decision boundary and margins

Although there are many possible candidate hyperplanes, the SVM maximizes the minimum distance to decide the optimal hyperplane rather than minimizing the margin. One thing that needs to note here is that some classification models, such as logistic regression, predict the probability of each class as the outcome instead of predicting the labels themselves directly. That is, a data point may be classified as positive if the predicted probability of a positive class is greater than or equal to a threshold such as 0.5.

We know that SVM can give each data point's class directly as the outcome but not the class probability. As shown in Figure 3.11, the SVM transforms the original data into a new space and uses a hyperplane to distinguish new features by the two margins d_1 and d_2.

To reiterate, the objective in this study of using the process data was to predict examinee efficiency on Block B. In this section, the model prediction is evaluated with the adjusted Area Under the Curve (AUC; Myerson et al., 2001; Rakotomamonjy, 2004). AUC compares the false positive rate to the true positive rate from the model and measures how well the model can predict the outcome. We know that, for a classification task with two classes, random guessing will yield an accuracy of 0.5. So the adjusted AUC measure is defined as in Equation 3.28:

$$AUC_{adj} = 2 \times (AUC_{ori} - 0.5) \tag{3.28}$$

where AUC_{ori} is the original area value under the curve. Basically the value of $AUC_{ori} \geq 0.5$ means this model's performance is not worse than random guessing.

3.6.3 Feature extraction and model evaluation based on Subset 1

The objective of this study using Subset 1 is to extract features from Block A process data for each examinee and then to construct a classification model to predict each examinee's efficiency level on Block B. The original process data in this subset was split by personal ID. We reorganize the process data by both the personal ID and the item identification number. That is, the actions for each item are first aggregated, then we further organize the actions produced by different examinees. Therefore, the SRM is applied to the process information for each item for the $1,232$ examinees in this subset. Suppose for

zth item, the extracted high-dimensional feature matrix with a certain number of columns (i.e., a certain number of features) is denoted as $\boldsymbol{h}^{(z)}$. Finally, a series of SVM models are constructed, with each done by adding a given item's feature matrix one at a time. The adding of a given item's feature matrix is done by horizontally (column) concatenating all examinees' matrices as we showed in section 3.5.3 such that $H = (\boldsymbol{h}^{(1)} : \cdots : \boldsymbol{h}^{(z)})$. In this study, SVM-Recursive Feature Elimination (SVM-RFE; Guyon et al., 2002; Rakotomamonjy, 2003) method is used to determine a set of selected features. It ranked features concerning their relevance to the cost function based on a backward sequential selection. That means one starts with all the candidate features and removes chunks of features at a time, and finally finds a subset of features that may produce the best classification result. The removed features are the ones whose removals have the least effect on the variation of the SVM weight vector norm.

One thing that needs to be noted is, we have introduced an advantage with the SRM. This is the adjustable reservoir weight matrix in section 3.3.1. It has been described in sections 3.4.1 and 3.4.2. In this study, the same set of reservoir sizes, i.e., from 500, 1000, 2000, 3000, 4000, and 5000, was applied and the model was trained for each of the sizes to learn features as shown in sections 3.4.1 and 3.4.2. Similarly, the same set of feature numbers is used by the SRM to select the optimal feature for the log data of each item as we indicated in section 3.3.1. The SRM will attempt to use each of the pre-defined numbers (from 25, 50, 75, 100, 125, 150, 175, 200), and select the one producing the best result. These indicate that, for each item's log data, one of the feature numbers will be selected by the SRM.

Table 3.10 shows the length of all action sequences for each item and the corresponding optimal feature number selected from the pre-defined number set. From this table, it can be seen that SRM selected different feature numbers for each item, per the sample size and data dimension. The smallest number of features is 50, and the largest number of features is 150 for both item VH134366 and item VH139196. The

feature matrices for all items were successively horizontal-concatenated by the list order in Table 3.10. That means, at each time, a feature matrix for the next item will be concatenated to the previous one, so this set of concatenated feature matrices can be used to predict each examinee's efficiency level using information from the different number of items.

Table 3.10: Item sequence length and number of features

Item	Number of examinees	Length of sequence	Number of Features
VH098519	1232	16,768	100
VH098522	1057	18,837	100
VH098556	1097	6,808	50
VH098597	1121	9,286	50
VH098740	1229	12,235	75
VH098753	1229	19,270	100
VH098759	1229	20,905	100
VH098779	1083	8,897	50
VH098783	1219	18,584	100
VH098808	1229	18,756	100
VH098810	1232	9,178	50
VH098812	1210	11,823	75
VH098834	1070	8,351	50
VH098839	1157	11,896	75
VH134366	1230	71,979	150
VH134373	1184	34,382	100
VH134387	1226	34,118	100
VH139047	1228	29,260	100
VH139196	1201	61,587	150

Figure 3.12 presents the AUC_{adj} changes against the adding of a given item's feature matrix one at a time in the SVM models. Each tick on the x-axis indicates the addition of the corresponding item's features into the SVM classification model. The initially used features in the SVM are from item VH098519, which creates an AUC_{adj} of 0.006. We can also observe that, first, the addition of item process information can help to classify examinees' efficiency levels. It is possible to see that, with more items incorporated, the SVM model AUC_{adj} gradually increases and finally results in a value of 0.481.

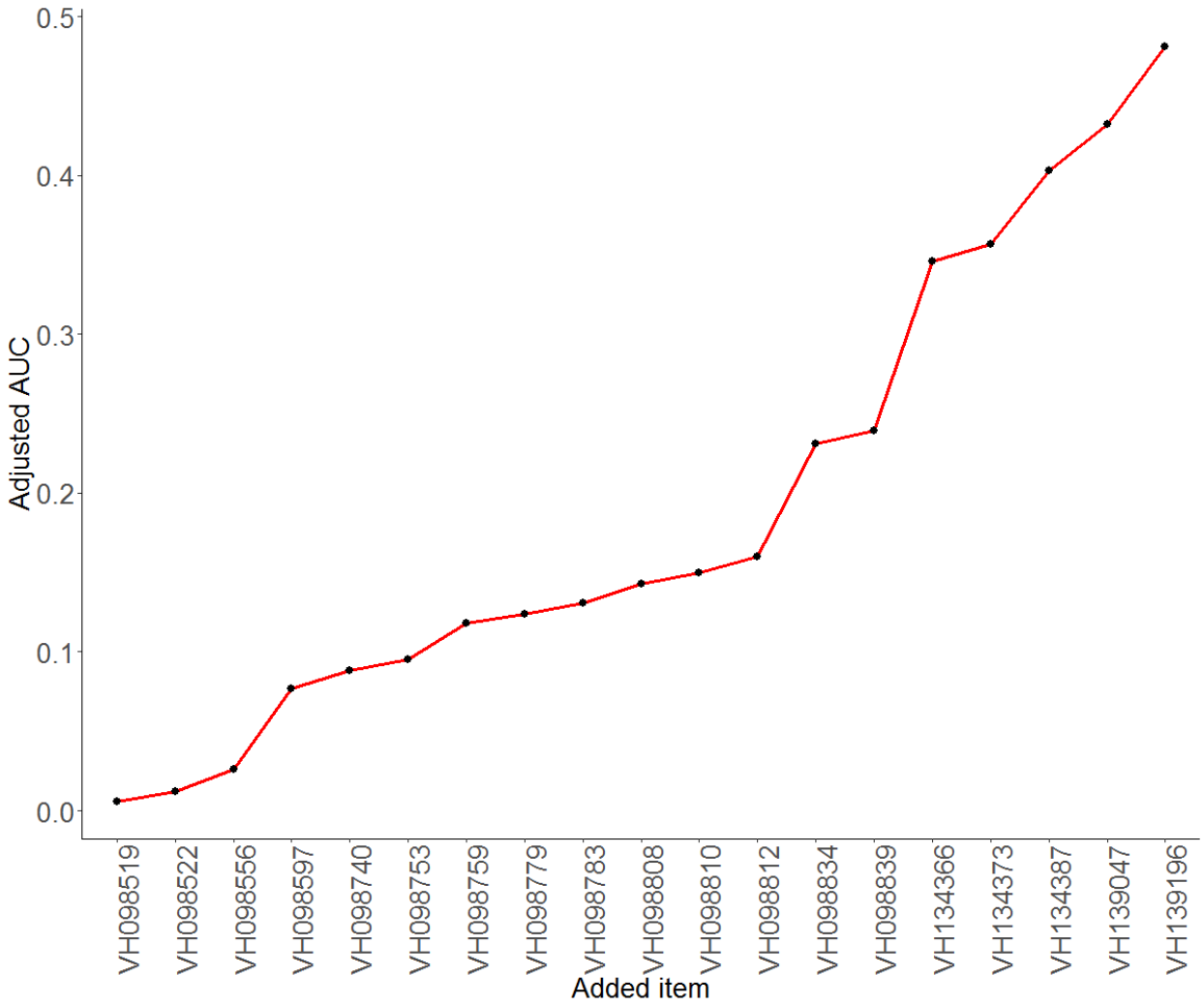

Figure 3.12: Adjusted AUC value changes against the addition of new item's features in the SVM

The other finding is that some item process information appears to contribute more than other items. That is, the addition of some process features for some items may be more useful for classification accuracy. Table 3.11 shows the number of unique actions and the corresponding increase of AUC_{adj} value for each of these items. The top three highest AUC_{adj} values are bolded. From this table, we can see that the use of features from item VH134366 resulted in the AUC_{adj} values of 0.107; while the addition

of Item VH098522 or Item VH098779 only resulted in a slight increase of 0.006 in the AUC_{adj}. The correlation between the number of unique actions and the corresponding increase of AUC_{adj} values was only 0.221, which suggests there is no strong relationship between the number of unique actions and the corresponding increase of AUC_{adj} values.

Table 3.11: Number of unique actions provided by each item and their corresponding contribution to the model classification AUC_{adj} values

Item	Number of unique actions	Corresponding increase of AUC
VH098519	29	-
VH098522	29	0.006
VH098556	27	0.014
VH098597	27	**0.051**
VH098740	30	0.011
VH098753	29	0.007
VH098759	28	0.023
VH098779	25	0.006
VH098783	29	0.007
VH098808	30	0.012
VH098810	27	0.007
VH098812	30	0.010
VH098834	25	0.071
VH098839	28	0.008
VH134366	32	**0.107**
VH134373	32	0.011
VH134387	32	**0.046**
VH139047	28	0.029
VH139196	34	**0.049**

The final SVM model with the areas under the curve for all items features. Variable importance analysis is performed along with the SVM-RFE. The variable importance refers to the measure of the extent to which the model uses a given variable to make an accurate prediction or classification. The more that model relies on a variable to make the inference, the more important that variable is for that model. The use of variable importance has been reported in machine learning and regression studies to select important features (e.g., Dewi and Chen, 2019; James et al., 2013; Sanz et al., 2018; H. Wang et al., 2015). It does not

restrict how one should define the measure of variable importance. gave several definitions to the process of measuring a variable's importance. Based on those definitions, Wei et al. (2015) further summarize a number of measures to identify a variable's importance with respect to different methods such as the entropy-based measure and variance-based measures. In the importance analysis, then, importance is defined as the weight of each feature in the SVM (Guyon et al., 2002; Huang et al., 2014). This weight is scaled to be between 0 and 100. In Figure 3.13, we plot the top ten important features (i.e., V1-V10). From this figure, we can see that some of the top features may have close values of importance. See, for example, Features 3, 4, also of Features 8, 9, and 10. Below, we interpret the meanings of the importance of these ten features in Table 3.12.

Since the features were automatically selected by the SRM and they don't have any pre-defined meanings, an exploratory analysis is used. In this analysis, correlations are computed between each feature vector and specific actions in Block A (see Table 3.12). The defined variables (i.e., actions) for each item include each examinee's median action length, each examinee's action length, how many changes each examinee made when responding to MC items, and how many items each examinee responded to more than one time.

Table 3.12 lists each feature and the defined variable that has the highest correlation with this feature. The feature will be interpreted using the meaning of the defined variable. For instance, the 1st feature has a correlation of -0.43 with the defined variable of action length of item VH134387. The critical correlation value, r^*, is used to decide to reject or not reject a null hypothesis of a correlation significance test. Here the critical correlation value for the two-tail test is $r^*(df = 1230, \alpha = 0.05) = 0.056$, therefore -0.43 is significant under the $\alpha = 0.05$ level because $|-0.43| > 0.056$. Similarly, all the correlations were significant for the two-tail test under the $\alpha = 0.05$ level. The action length of item

VH134387 represents each examinee's sequence length of item VH134387. This may be interpreted to indicate that, if the examinee's sequence length of item VH134387 is high, then this feature would have a lower value, suggesting that it could result in an important change to the determination of level in the SVM. On the countrary, the 8th feature can be explained as representing the number of eliminating choices, since this variable has a correlation of 0.38 with the 8th feature. So, it could be interpreted to mean that, if an examinee had a larger number of eliminated choices, the examinee would have a higher value on the 8th feature.

Table 3.12: Interpretation to the first ten important features

Feature	Interpretation of this feature	Correlation
V1	Action length of item VH134387	-0.43*
V2	Action length of item VH098834	-0.54*
V3	Action length of item VH098759	-0.62*
V4	Action length of item VH134366	-0.67*
V5	Number of eliminating choice	0.38*
V6	Action length of item VH139196	-0.55*
V7	Action length of item VH098597	-0.61*
V8	Number of clicking progress navigator	0.40*
V9	Action length of item VH139047	-0.44*
V10	Number of opening calculator	-0.39*

* indicate a significant correlation at $\alpha = 0.05$

3.6.4 Feature extraction and model evaluation based on Subset 2

Subset 2 contains three different process portions, each standing for different testing periods for the three random groups of examinees. Using the results for these three different groups could potentially be used to help us to explore two questions: first, whether processes across different items can be combined together, and second, whether a proportion of the whole process is sufficient to predict the examinees' efficiency level.

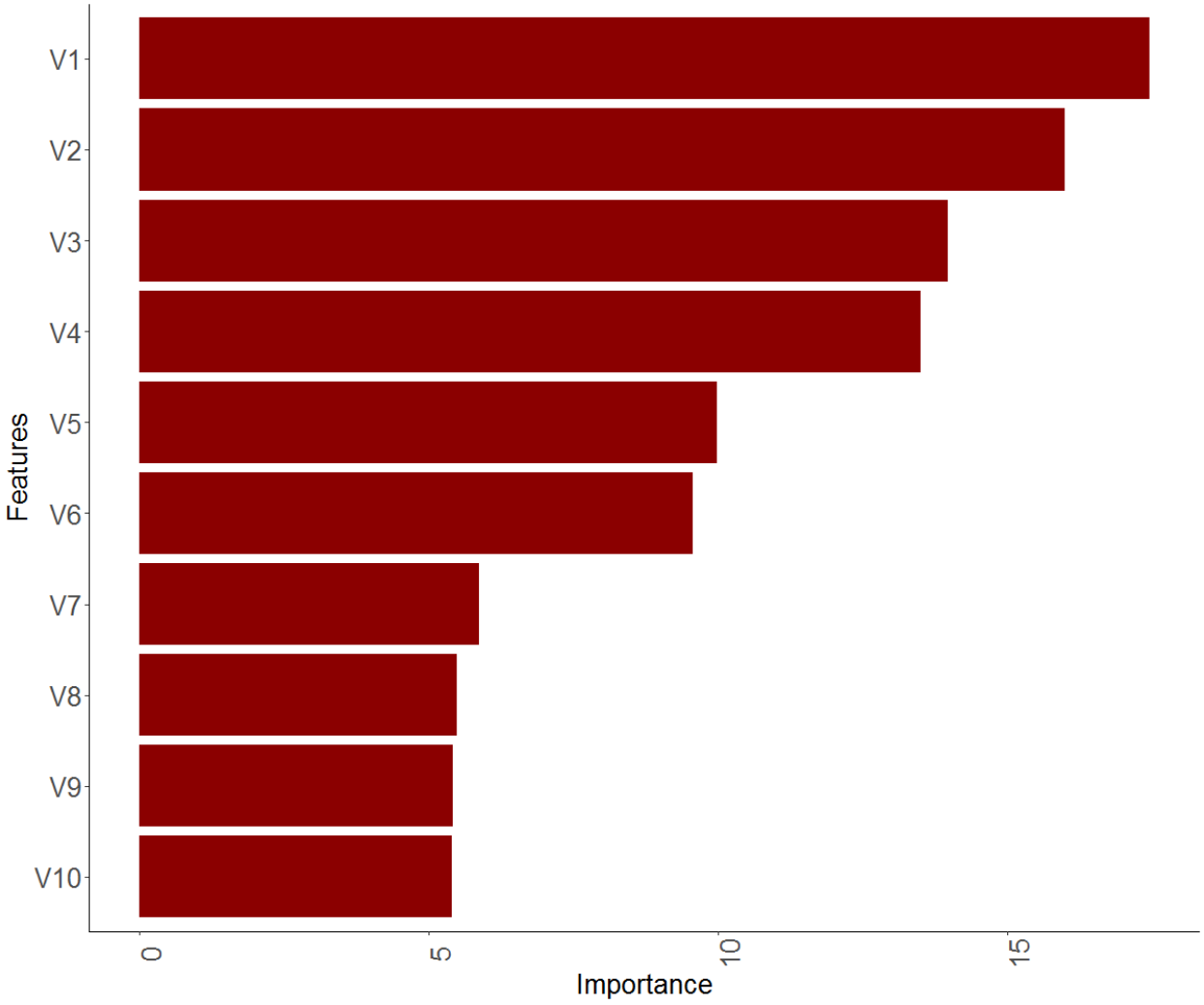

Figure 3.13: The first ten important features from subset 1 used in the final SVM model

Within each portion, below we applied the SRM to examinees' process information for that amount of time. An increasing reservoir from 500 to 5000 was applied and the model was trained to learn features with numbers from $(25, 50, 75, 100, 125, 150, 175, 200)$. Table 3.13 shows the sample size, sequence length, extracted feature numbers, and AUC_{adj} values for each portion. The SRM appears to select more features for data with a longer sequence. The AUC_{adj} values for the three portions are 0.12, 0.34, and 0.39, respectively. This suggests at least two results. First, longer process information appears to yield a higher classification accuracy, although the result from the 20-minute data group is close to that from the 30-minute data group. One take-away is that this may suggest that a proportion of the whole process might contain some useful information and a 20-minute set of the data might be sufficient to predict the examinees' efficiency levels. Second, it is interesting to see that combined processes from different items can also be used to predict examinees' efficiency levels. However, although the classification AUC_{adj} seems to be lower than what we obtained from subset 1, in which we first separated the item process and then combined the features from different items, it does not indicate that using features extracted from processes combined across different items will reduce the classification accuracy. Possibly this is because the sample size in this study for each of the three portions (411, 411, and 410) is smaller than the larger sample of $1,232$ that we had for subset 1.

Table 3.13: Feature number and adjust AUC value for each portion in Subset 2

Data	Sample size	Length of sequence	Feature number	AUC_{adj}
Portion 1	411	47,563	75	0.12
Portion 2	411	110,481	100	0.34
Portion 3	410	143,880	150	0.39

The plot and interpretation of the top ten important features from data portion 3 is given as an example. Figure 3.14 similarly shows that the importance among these features could be different, and this could be because these features are not extracted from only one item's process so features may focus

on different aspects of the combination of item processes. Similarly, the correlation analysis between each feature vector and the defined variables was conducted. Table 3.14 lists some possible interpretative information regarding each feature. For instance, the 1st feature can be interpreted as demonstrating the importance of the total sequence length of all items in Block A. The 2nd feature has a correlation of 0.64 with the number of items the examinee responded to more than once. This may suggest that, if the examinee responds with a given activity for more items more than one time, this feature value will likely be higher. One possible reason is that the examinee may have sufficient time to go back to items. Compared with the features extracted from subset 1, which mainly focuses on the information from a single item, the feature extracted based upon a combination of all item processes tends to contain some global information such as how many times the examinee used the math keypress or how many items did the examinee responded to more than one time. The critical correlation value for the two-tail test here is $r^*(df = 1230, \alpha = 0.05) \approx 0.097$, therefore all the correlation values are significant under the $\alpha = 0.05$ level.

Table 3.14: Interpretation to the first ten important features

Feature	Interpretation of this feature	Correlation
V1	The total sequence length of all items in Block A	-0.59*
V2	The number of items did the examinee enter more than once	0.64*
V3	Action length of item VH098834	-0.58*
V4	The median number of times the examinee enter each item	0.47*
V5	Action length of item VH134366	-0.66*
V6	The median number of changes made to each item	-0.48*
V7	Action length of item VH139196	-0.71*
V8	Action length of item VH098597	-0.39*
V9	The total number of times the examinee used the math keypress	-0.44*
V10	The median number of times the examinee losing focus	-0.37*

* indicate a significant correlation at $\alpha = 0.05$

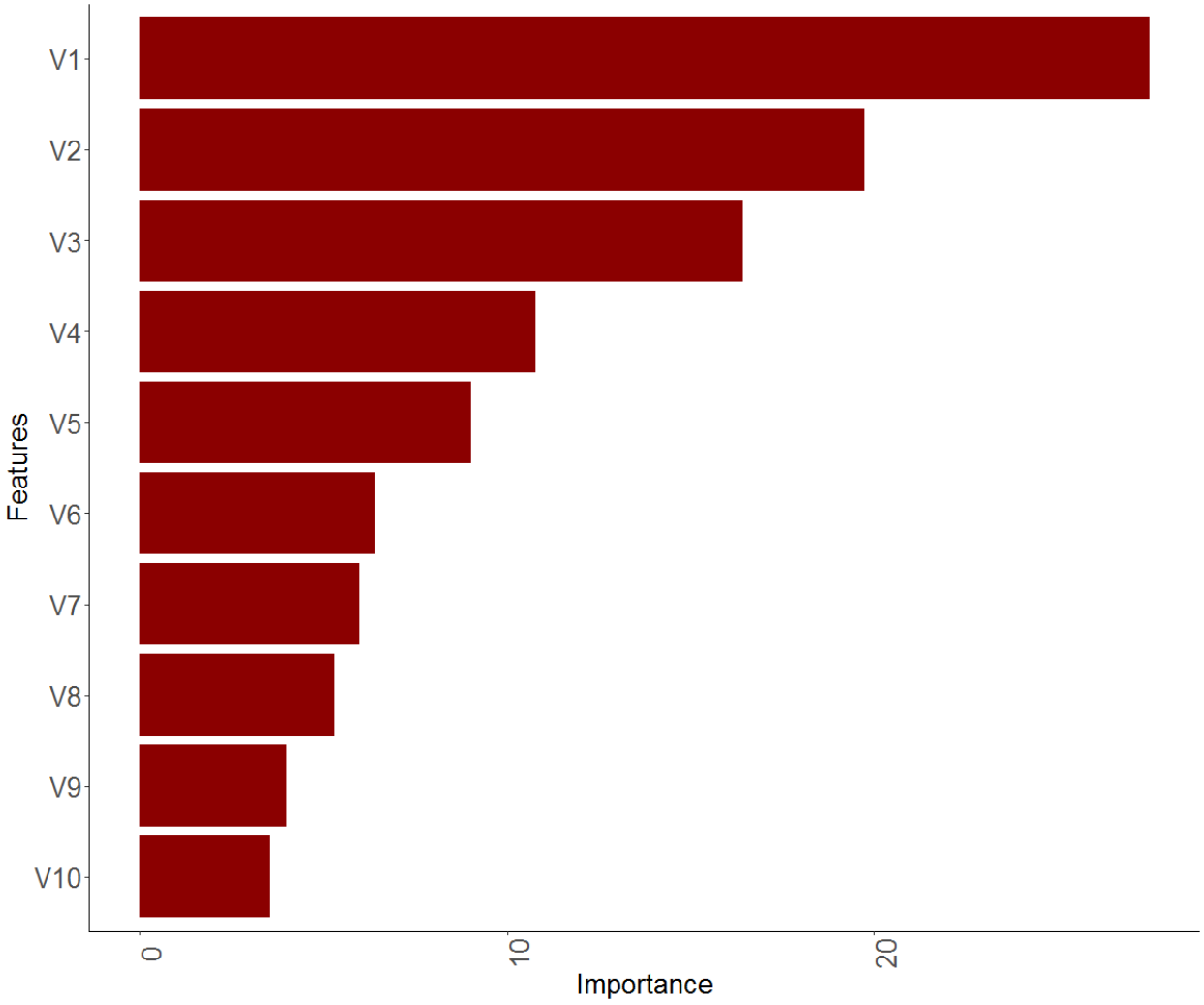

Figure 3.14: The first ten important features from subset 2 portion 3 used in the final SVM model

3.7 Summary and Discussion

In this chapter, the SRM is investigated with a focus on feature extraction from the process data from MC items. The sequence of actions by each examinee on each item is likely to be different within a single test and between examinees in the population. By applying the SRM, the varying-length sequence data was shown to be successfully transformed for each examinee into different fixed-length vectors that can be stored into a single feature matrix. Three simulation studies were used to demonstrate SMR feature utilization. Simulation Study I used SMR to classify samples that were generated from three different Markov groups. Results suggest that the group classification accuracy using the SRM features was much higher than that from using the baseline features. So, it seems possible that the features extracted by the SRM may contain meaningful process information about the different Markov groups. Results also suggest that a larger sample size and a larger reservoir could produce a more accurate classification. Simulation Study II attempted to merge examinees' latent trait values into the process data. It then used the extracted features to predict each examinee's latent trait value. Similar to Simulation Study I, results from Simulation Study II suggested that using the SRM features can produce lower errors with respect to the estimate of the latent trait than using the baseline features. Another interesting result from Study II was that the extracted SRM features appear to have a potentially better representation of the sequence and the latent abilities, if the item has fewer unique actions for examinees.

Based on these two simulations, it appears that features can be extracted from the simulated two genres of datasets (i.e., the *RspData* data set and the *Rsp+ProcData* data set). Further, the features appear to be useful for group classification and latent trait prediction. In the second simulation study, results suggested that the features can be used to recover the generating latent trait values. Therefore, in the

third simulation study, we tried to used both the extracted features and the response score pattern to see whether the addition of features could provide a more accurate ability parameter estimate and a better model fit. In that study, RSE and R^2 were used as model fit indices for two linear models, one *RspData* model contained only response patterns and the other *Rsp+ProcData* model contained both the response patterns and process features. Results suggest that the model fit given by the *Rsp+ProcData* model was better than that by the *RspData* model. Thus, the use of process data appears to provide better fit to the linear model than the use of response patterns alone.

Results from the application of the SRM to the empirical data from the NAEP math assessment were in agreement with some of the conclusions from the simulation studies. The NAEP assessment was split into two subsets, and both subsets were modeled by the SRM to extract features that were later used by the SVM. In subset 1, examinees' process information was divided by item identifications and ll item features matrices were concatenated together. These were then analyzed by the SVM to predict each examinee's efficiency on Block B. Results suggested that with more items, the SVM model AUC_{adj} increased, and finally resulted in a value of 0.481. In addition, some item process information may contribute to greater effect than other items. Thus, addition of the process of these features may be more useful for classification accuracy. The important features were interpreted based on their correlations with the defined variables noted, such as the item's sequence length. In subset 2, each examinee's process information was used based on the whole 30-minute time limit. Results suggest that a proportion of the whole process, i.e., the 20-minute period, may contain some information for predicting examinees' efficiency level. In addition, longer strings of process information might yield a better classification result. Finally, features extracted based upon a combination of all item processes tended to contain some useful information with respect

to predicting ability such as how many times did the examinee use the math keypress and how many items did the examinee enter for more than one time.

Chapter 4

Topic model: an interpretable algorithm for analyzing constructed-response item process data

In this chapter, the use of the topic model and its extensions is applied to the textual responses collected from the examinees' CR item responses. Specifically, the unsupervised topic model and supervised topic model are both applied to empirical data to help understand examinees' writing structure and to predict examinees' writing scores. Next, a framework is provided describing an automated CR scoring engine using a supervised topic model. Finally, a reliability study is presented using a Rasch model and a topic model.

4.1 Unsupervised and Supervised Latent Dirichlet Allocation

LDA is an unsupervised learning algorithm. It is referred to as unsupervised as the only information used to guide the model in detecting latent topics are the words in the documents in the corpus. LDA assumes that the order of words in a response does not matter, and this is referred to as the "bag-of-words" assumption. Based on this assumption, LDA is designed to detect topics in a corpus and the proportions of topics in each response. In the LDA, each response is made up of various words, and each topic also has various words belonging to it. The goal of LDA is to find topics in a response based on the words in the response. In other words, the words are the observed variables. Given a collection of responses, let us assume there are K topics, where the number K is assumed to be known and fixed. Then for each of the K topics, the words that belong to the topic or the probability of words belonging to the topic will be estimated by the LDA.

The LDA algorithm is a generative probabilistic model. That is, the term generative means this is an assumption that describes how the word-topic distribution and the topic-document distribution in a corpus are generated. This means that fitting the generative model is fitting a statistical model. It does not necessarily mean that this is an indication of model validity.

Suppose we have a response in a corpus such that $d \in D = \{d_1, ..., d_M\}$, and also we have each of the words in a response such that $w \in W = \{w_1, ..., w_N\}$. Then LDA defines the following generative steps for each response in the corpus D:

1. Select a response d_i with probability P_{d_i};

2. Choose the number of words N_i from the Poisson distribution with parameter ξ, i.e., $N_i \sim Poisson(\xi)$;

3. Draw the topic distribution θ_i from the Dirichlet distribution with parameter α, i.e., $\theta_i \sim Dir(\alpha)$;

4. For each of the N words w_n in the response d_i:

   (a) choose a topic $z_{i,j}$ from the Multinomial distribution with parameter θ_i, i.e., $z_{i,j} \sim Multinomial(\theta_i)$;

   (b) draw a word distribution $\phi_{z_{i,j}}$ from Dirichlet distribution with parameter β, i.e., $\phi_{z_{i,j}} \sim Dir(\beta)$;

   (c) choose a word $w_{i,j}$ from the Multinomial distribution $\phi_{z_{i,j}}$, i.e., $w_{i,j} \sim p(w_{i,j}|z_{i,j}, \beta)$, a Multinomial probability conditioned on topic $z_{i,j}$.

Some notations are explained as follows:

- ϕ_k: the probability distribution of the words for topic k;
- θ_i: the probability distribution of topics for response i;
- α: the prior distribution parameter to the topic distribution θ_i;
- β: the prior distribution parameter to the word distribution ϕ_k;
- N_i: total number of words for a response;
- M: total number of responses in the corpus.

The Dirichlet distribution is the conjugate prior distribution to the multinomial distribution. The basic idea behind the conjugate prior is that the prior distribution and the posterior distribution are of

the same form. The graphical model representation of the above steps can be given in Figure 4.1, and there are three levels in the LDA model. The parameters α and β are at the corpus level, which will be sampled once in the process of generating a corpus. The variables θ_i's, sampled once per response, are at the response level, Finally, the variables $z_{i,j}$, $w_{i,j}$ and ϕ_k are at the word level, and they are sampled once for each word in each response. In this model, the only observable variable is the $w_{i,j}$, which is the observed response data. The other variables such as ϕ, z, and θ are all unknown variables.

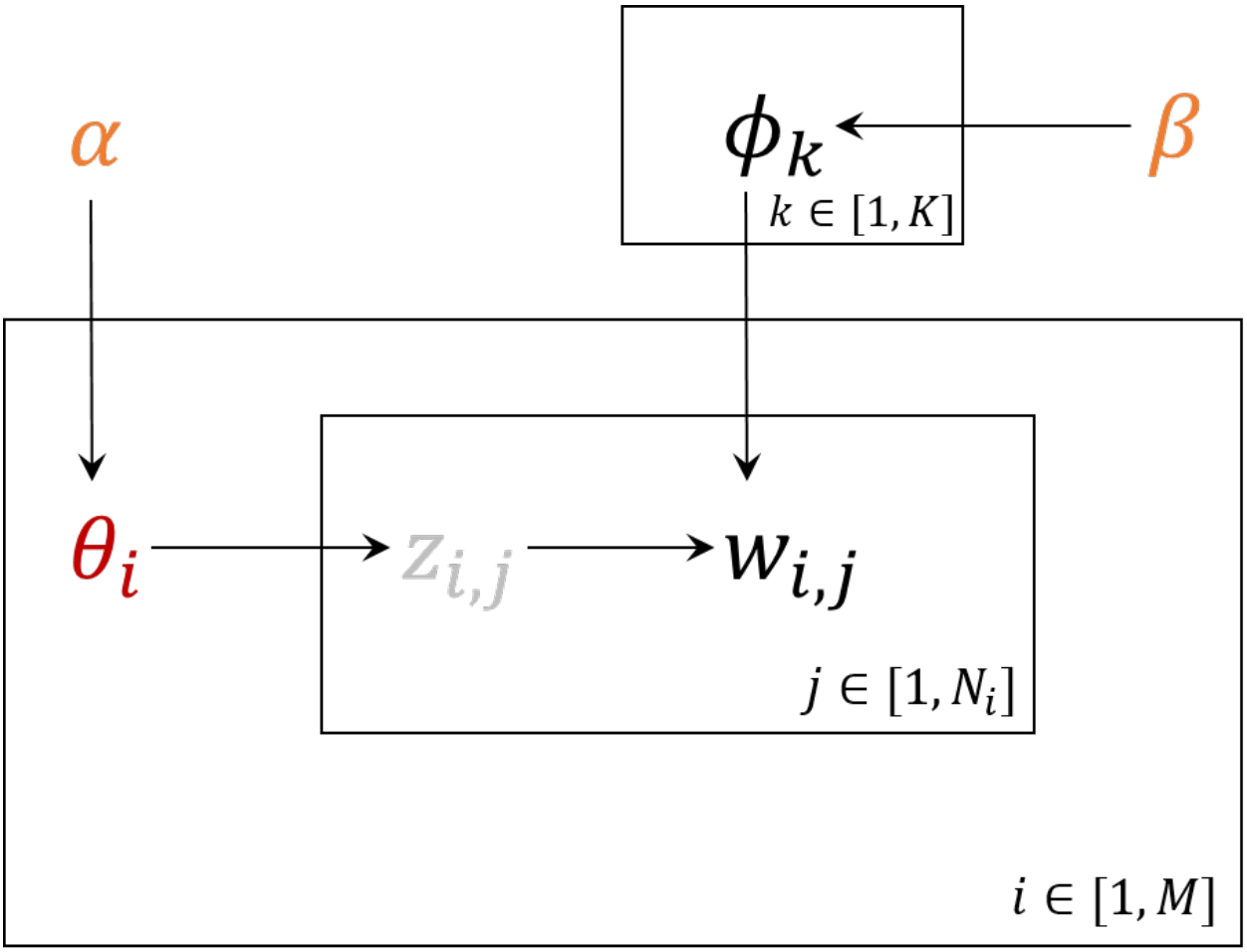

Figure 4.1: A graphical model representation of the LDA model*

*drawn based on (Blei et al., 2003)

Given the parameters α and β, the joint distribution of a topic mixture θ_i, a set of topics $z_{i,j}$, and a set of words $w_{i,j}$ is given by Equation 4.1:

$$p(w, z, \theta) = \prod_{j=1}^{N_i} p(z_{i,j}|\theta_i)p(w_{i,j}|z_{i,j}) \tag{4.1}$$

Supervised LDA (sLDA; Mcauliffe and Blei, 2007) uses external information to help guide the LDA model. In this way, sLDA is an extension of the LDA model that includes additional information, referred to as labels. In the context of CR answers, the labels are the rubric-based scores of examinees' responses. sLDA is different from the unsupervised LDA model in that it jointly models the text along with the associated supervisory label to estimate appropriate latent topics which can predict the label for future responses. The label could be of various types such as real values or ordered class labels as might be obtained as a rubric-based score.

Suppose there are K topics $\beta_{1:K}$ in the responses. With the Dirichlet parameter α, response parameter η and σ^2, the sLDA model estimates the response and response label in the following steps:

1. The topic proportions $\theta|\alpha$ are drawn from $Dir(\alpha)$.

2. The topic assignments $z_n|\theta$ are drawn from $Multinomial(\theta)$.

3. The word $w_n|z_n$ is dawn from each topic z_n, where $\beta_{1:K}$ follows $Mult(\beta_{z_n})$.

4. The response variable $y|z_{1:N}, \eta, \sigma^2$ is then drawn from $N(\eta', \bar{z}, \sigma^2)$.

where the $\bar{z}$ here is defined to be $\frac{1}{N}\sum_{n=1}^{N} z_n$. The natural parameter ζ and dispersion parameter δ were used in the canonical link function under the generalized linear model. Therefore, the response variable

has the following distribution Equation 4.2 under the general version of sLDA:

$$p(y|z_{1:N}, \eta, \delta) = h(y, \delta) \exp \frac{\eta'(\bar{z}y) - A(\eta'\bar{z})}{\delta} \tag{4.2}$$

where $\eta'\bar{z}$ is the linear predictor and is set to be identical to the parameter ζ; $h(y, \delta)$ is the base measure; y is a sufficient statistic; and $A(\eta'\bar{z})$ is the log-normalizer. The graphical model representation of the sLDA steps is shown in Figure 4.2.

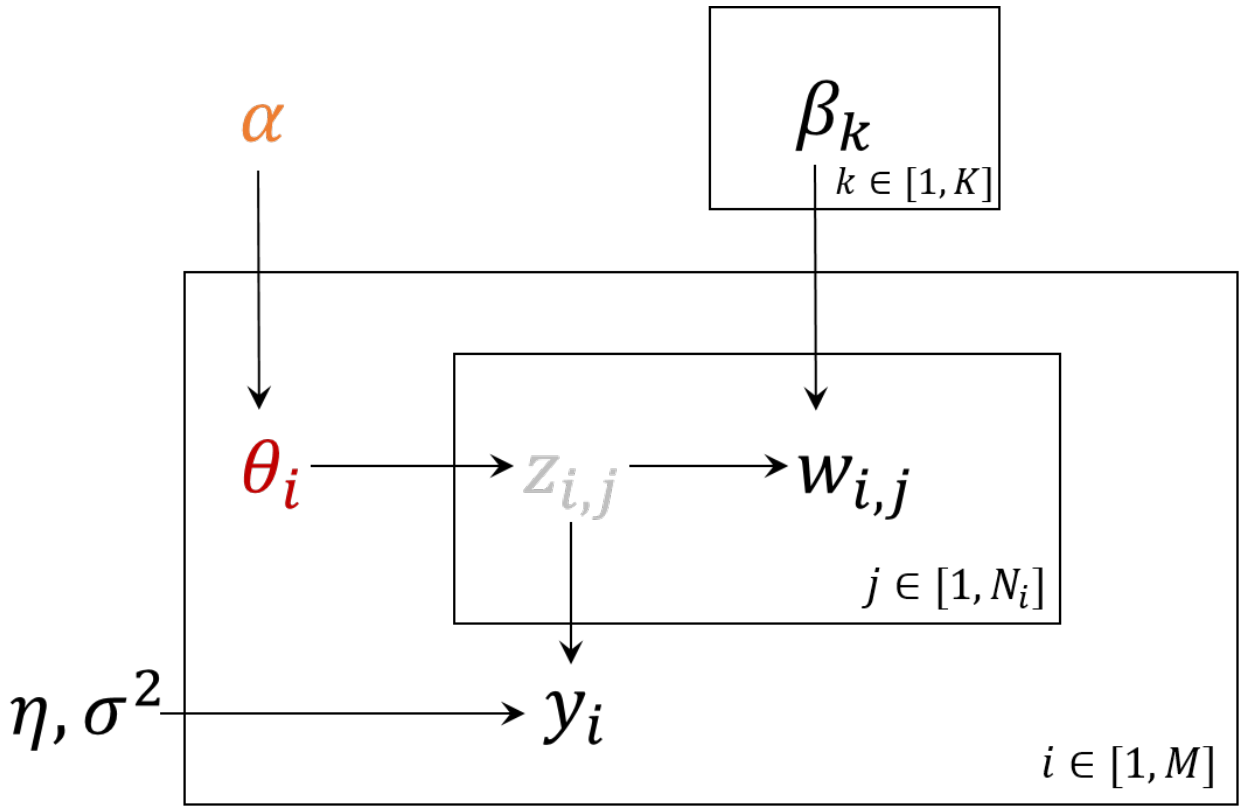

Figure 4.2: A graphical model representation of the sLDA model*

*drawn based on (Mcauliffe and Blei, 2007)

The sLDA model uses a linear model to predict an outcome variable using the topic model proportions. For instance, if the outcome variable indicates the rubric-based score, then this score is regressed on the

topic proportions. Then the following Equation 4.3 is the regression model for the sLDA:

$$Y_i = \beta_X \tag{4.3}$$

where $Y_i = (y_1, \ldots, y_n)'$ is the observed label, X is the topic proportion matrix, and β represents regression coefficients.

4.2 Comparison of Unsupervised and Supervised Model Analyses of Constructed-response Answers on Two Social Study Assessments: An Empirical Example

In this section, results for two different topic models, latent Dirichlet allocation (LDA) and supervised LDA, were compared for their utility in detecting different latent thematic patterns in examinees' responses to two social studies assessments: A U.S. History assessment and an Economics assessment. Their results are also compared.

Topic models have been used in the past in social science research. For example, Roberts et al. (2013) used a topic model to detect the latent structure in open-ended responses to a social science survey; Grimmer (2010) used a topic model to detect the latent structure in political rhetoric; and Yang et al. (2011) used topic modeling to detect historical trends from passages.

4.2.1 Data description

Responses from two items were analyzed in this study: one are the answers to a CR item from a U.S. History assessment, and the other the answers to a CR item from an Economics assessment. The U.S. History assessment was administered to 722 examinees in Grade 9 to Grade 12. The economics assessment was administered to 663 examinees in Grades 9 to Grade 12.

Both assessments were developed to be aligned to the state standards in the respective subjects. There were 22 multiple-choice items, 2 short-answer CR items, and 1 extended CR in each assessment. The CR items were designed to require extended reasoning and critical thinking. The two short-answer CR items were scored from 0 to 2 points and the extended response item was scored from 0 to 4 points. Only the answers to the extended response items in each assessment were analyzed. For both assessments, the extended response item consisted of a question followed by two passages describing the context for the response.

4.2.2 Data cleaning

A series of pre-processing steps was applied to clean the original data. The pre-processing procedure included word stemming, lemmatization, removal of stop words and whitespaces, changing numerical digits to text, changing upper case letters to lower case, correcting typo graphical errors or other possibly non-standard English, and removal of punctuation characters. Some examples are changing the plural words into singular words, and changing gerunds and past tense into the stem format. The lemmatization uses the context in which the word is being used and changes the word into the base forms for irregular verbs and irregular plural nouns.

The stop word refers to commonly used words such as "*the*". These are considered high frequency, low information words. Removal of these kinds of wods is necessary to encourage the algorithm to make interpretable clusters. Stop words are a necessary aspect of the language but need to be ignored in topic modeling. The stopwords for this study are shown in Table 4.1.

Table 4.1: Stop words for the U.S. History assessment and the Economics assessment

U.S. History item						Economics item					
next	into	according	not	their	this	next	not	their	this	only	one
much	can	yet	for	every	and	yet	for	and	are	that	what
what	him	with	but	out	his	but	out	his	who	from	will
will	they	also	which	other	you	which	other	you	still	our	all
all	how	than	two	after	many	two	after	many	have	both	there
there	just	now	have	one	that	every	into	its	when	while	then
only	are	who	still	from	our	much	can	they	also	just	now
both						him	with	how	than	about	yes

Responses with less than 10 words after data cleaning were also excluded from the analysis as words with such low frequencies would be too sparse to be detected as belonging to a topic. Therefore, the number of responses was reduced following data cleaning. Descriptive statistics of the numbers of words, number of responses, and average response length are given in Table 4.2. It can be seen that all the numbers are reduced for both items after processing. This also saves the model estimation and computation time.

Table 4.2: Number of responses, number of words, and average response length before and after data cleaning

	U.S. History		Economics	
	Before cleaning	After cleaning	Before cleaning	After cleaning
Number of responses	722	416	663	482
Number of unique words	583	296	332	145
Total Number of words	22,203	9726	19,526	9,143
Average length	53	23	40	19

4.2.3 Model selection

Exploratory use of a topic model typically consists of estimating models with different numbers of latent topics. The best-fitting model of these candidate models then needs to be determined. As topic models are not nested, selecting the best fitting model typically is informed using one or more information criterion indices. When the topic model is estimated using a Bayesian algorithm, the Deviance Information Criterion (DIC; Spiegelhalter et al., 1998) is often used to inform model selection. Denote the deviance in Equation 4.4:

$$D(\theta) = -2log(p(y|\theta)) + C \tag{4.4}$$

where y indicates the data, θ is the unknown model parameter, $p(y|\theta)$ is the model likelihood, and C is a constant. Then DIC can be defined as follows in Equation 4.5

$$DIC = D(\bar{\theta}) + 2p_D \tag{4.5}$$

where p_D is the difference between the $\bar{D(\theta)}$ and the $D(\bar{\theta})$ based on the posterior means. Therefore, lower DIC values may indicate a better model fit. The plots of DIC values for with from 2 to 10 latent topics for two assessments are given in Figure 4.3. In this study, DIC suggested a 4-topic model for the U.S. History item and a 3-topic model for the Economics item, since the lowest DIC for each assessment is taken as the suggested model.

The plots of DIC values for with from 2 to 10 latent topics for two assessments are given in Figure 4.3. In this study, DIC suggested a 4-topic model for the U.S. History item and a 3-topic model for the Economics item, since the lowest DIC for each assessment is taken as the suggested model.

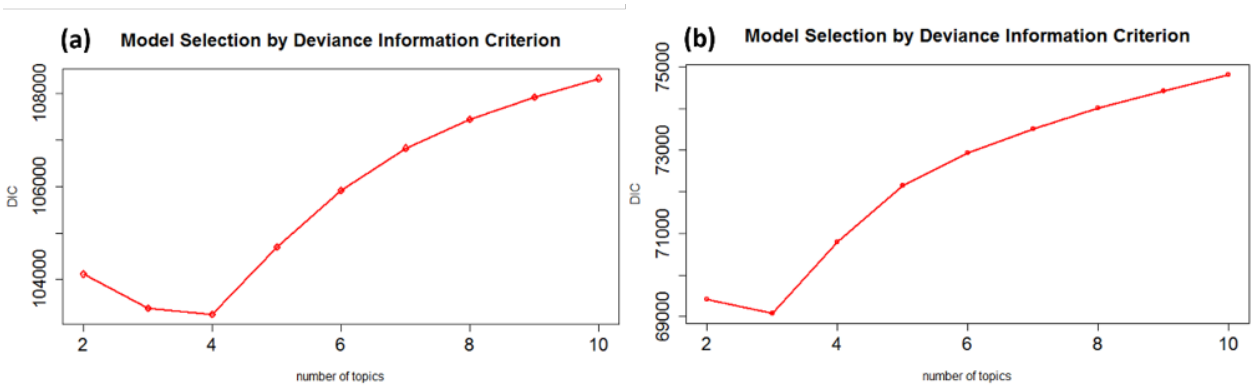

Figure 4.3: Plots of DIC for topic models estimated for the U.S. History and Economics items

4.2.4 LDA analysis results

The top 15 highest probability words for each topic extracted from the U.S. History item are given in Table 4.3. Inspection of these high probability words for each topic can often help to interpret the latent theme captured by the topic. Topic 3 contains high-frequency words that could be characterized as the use of *Everyday Language* in their responses to the item. Topic 4 consists of words about *U.S. Presidents and Civil Rights*. Examinees who used these words followed instructions from the prompt and tried to integrate information in the passages for the item and used this information as evidence to support their responses.

The correlation between the rubric-based score and the topics estimated in the 4-topic model is also given in Table 4.3. Correlations between the topic and the rubric-based score are listed in the heading for each topic. Correlation is calculated between item score and logit of the proportion of topic usage. For example, Topic 1 has a moderate negative correlation ($r = -0.361$) with the rubric-based score, and Topic 4 has a moderate positive correlation ($r = 0.443$) with the rubric-based score. It also is useful to

analyze responses by examinees who make the highest use of each topic. For example, examinees who had the highest use of Topic 1 typically wrote responses to the question that simply copied the information in the stem or passages. Topic 2 has some important words from the item, but examinees using this topic tended to take sentences directly from the item question or passage without trying to integrate them into a response. Topic 3 consists of an integrated structure of both everyday words with language from the passages, however, the response did not include a clear argument. Examinees who made most use of Topic 4 typically used words from the passages and integrated them to provide evidence for their conclusions.

Table 4.3: Top 15 highest probability words for the 4-topic LDA model for U.S. History item

Topic 1 ($r = -0.361$)*		Topic 2 ($r = -0.162$)		Topic 3 ($r = -0.010$)		Topic 4 ($r = 0.443$)	
part	0.274	randolph	0.040	right	0.093	march	0.057
know	0.049	labor	0.025	part	0.081	martinlutherking	0.049
help	0.036	black	0.025	civil	0.042	randolph	0.048
want	0.017	america	0.023	people	0.032	presidentwashington	0.040
follow	0.010	african	0.020	movement	0.024	america	0.039
work	0.009	racial	0.014	protest	0.022	leader	0.025
learn	0.009	social	0.013	get	0.021	african	0.024
non	0.008	first	0.011	fight	0.019	protest	0.021
get	0.008	civil	0.011	want	0.017	right	0.019
like	0.008	world	0.010	equal	0.012	discrimination	0.018
people	0.008	brotherhood	0.009	make	0.011	civil	0.016
african	0.007	war	0.009	randolph	0.011	industry	0.016
white	0.007	movement	0.009	because	0.010	lead	0.016
presidentwashington	0.007	during	0.009	impact	0.010	war	0.014
could	0.006	philip	0.009	start	0.010	federal	0.013

Table 4.4 presents the topic structure of the Economics response. Topic 1 has a moderate negative correlation ($r = -0.403$) with the score and Topic 3 has a positive correlation ($r = 0.272$) with the score. The correlation for Topic 2 ($r = 0.124$) is low albeit positive. Topic 2 and Topic 3 can both be characterized as the use of academic language related to interest calculation. Examinees who use words mainly from Topic 2 were typically repeating the definitions in the passages while calculating the simple interest posed in the question. Examinees who use more words from Topic 3 provided responses that included choices and computation of the principle. Their responses also provided a convincing rationale. Topic 1 contains simple words but does not directly relate to the question. Some of the words for Topic 1 are not relevant for the response to the item.

Table 4.4: Top 15 highest probability words for the 3-topic LDA model for the Economics item

Topic 1 ($r = -0.403$)		Topic 2 ($r = 0.124$)		Topic 3 ($r = -0.010$)	
part	0.385	interest	0.236	interest	0.196
money	0.047	compound	0.055	compound	0.079
compound	0.034	principal	0.053	amount	0.068
because	0.025	simple	0.051	pay	0.057
dollar	0.022	rate	0.038	money	0.049
interest	0.020	loan	0.035	simple	0.049
rate	0.019	time	0.026	year	0.048
know	0.019	calculate	0.023	because	0.036
simple	0.019	retire	0.016	beneficial	0.022
time	0.016	deposit	0.012	part	0.015
take	0.013	save	0.012	rate	0.011
make	0.012	period	0.012	add	0.011
add	0.010	addition	0.010	save	0.010
retire	0.009	get	0.010	investment	0.009
good	0.009	good	0.010	time	0.009

4.2.5 sLDA analysis results

One thing to be noted here is that there is no intercept β_0 in the sLDA regression model as the topic proportions sum to 1, i.e., $\sum_1^k \theta_{nk} = 1$. The topic structure of the 4-topic sLDA model for the U.S.

History item from the sLDA analysis is given in Table 4.5. This model shows a similar pattern of topic proportions to those obtained from the LDA model.

Table 4.5: Top 15 highest probability words of the 4-topic sLDA model for U.S. History item

Topic 1 ($\beta = -0.159$)		Topic 2 ($\beta = -0.015$)		Topic 3 ($\beta = 1.169$)		Topic 4 ($\beta = 2.530$)	
part	0.533	randolph	0.063	right	0.153	march	0.098
know	0.078	labor	0.041	civil	0.100	randolph	0.082
help	0.058	black	0.039	protest	0.065	america	0.079
want	0.054	america	0.030	people	0.061	african	0.063
get	0.048	racial	0.022	movement	0.060	discrimination	0.035
make	0.028	war	0.021	because	0.034	lead	0.033
give	0.022	first	0.020	fight	0.031	work	0.033
thing	0.014	african	0.020	equal	0.029	president	0.031
same	0.012	union	0.019	impact	0.028	industry	0.027
null	0.012	during	0.019	influence	0.027	leader	0.025
everyone	0.012	social	0.019	direct	0.023	federal	0.024
stand	0.009	world	0.018	leader	0.022	order	0.019
cause	0.008	car	0.017	like	0.020	threat	0.019
good	0.008	philip	0.017	follow	0.020	equality	0.018
man	0.008	group	0.016	peace	0.018	government	0.018

The regression coefficients are computed for the regression of Observed Score on Topic Proportions. Topic 1 has a coefficient of $\beta = -0.159$ which means examinees who mostly used words from Topic 1 tend to have a low score. Similarly, Topic 2 has a coefficient of $\beta = -0.015$, which also means examinees who used words mostly from Topic 2 also have a low score. Topic 4 has a coefficient of $\beta = 2.530$, which means examinees who used words from this topic tended to have a score of 2.53 points. Differences between the observed score and the predicted score from the sLDA model are shown in the scatter plot in the left graph of Figure 4.4. The mean for these differences is given by $\mu = \sum_{i=1}^{n} |y_i - \hat{y}_i| = 0.598$ and the standard deviation 0.538. Compared with the score range from 0 to 4, the average difference of 0.598 between the obversed human rater score and predicted score is relatively small and indicates that the predicted scores are close to the obversed scores.

After ranking examinees' observed scores according to each topic's proportion, we took the top 50 examinees' observed scores because top 50 observed scores can be used as representatives to the small size data (i.e., less than 1000). The frequency of each score is shown in the histogram in the right graph in Figure 4.4. As is evident from the regression coefficients in Table 4.5, examinees who used words mainly from Topic 3 or Topic 4 have higher scores than examinees who used words mainly from Topic 1 or Topic 2. A full credit score of 4 does not occur for examinees who used words mostly from Topics 1 or 2. In addition, by comparing the number of zero scores among all the topics, the number of examinees who used Topic 4 is the lowest.

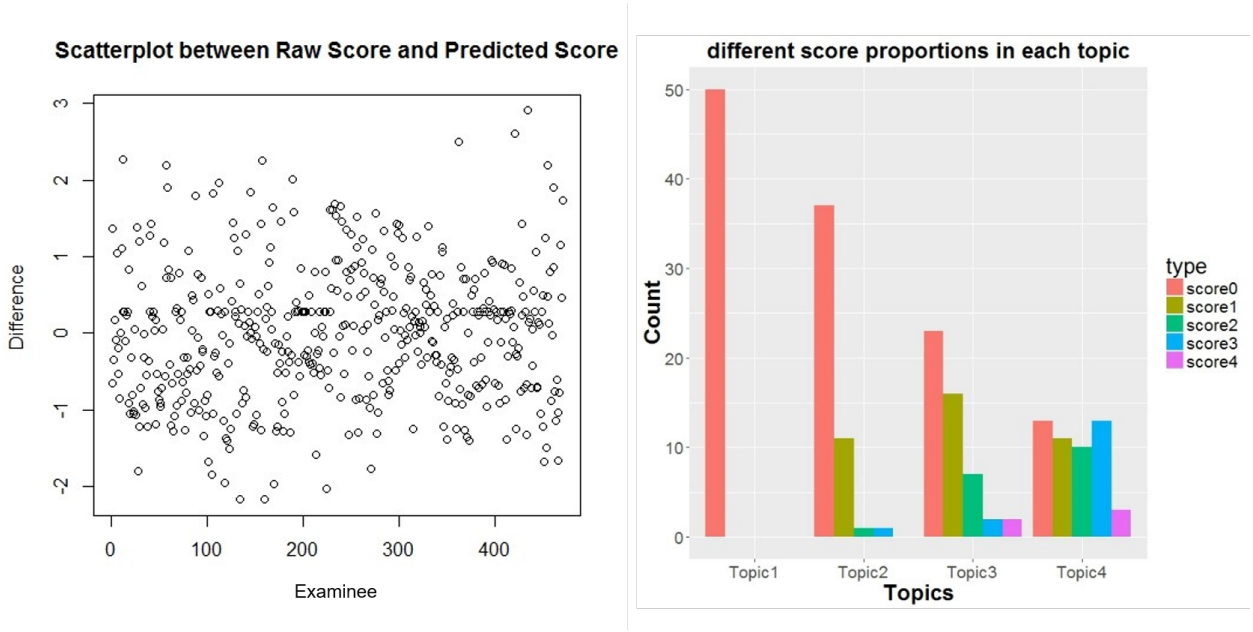

Figure 4.4: The left graph is a plot of the difference between the observed scores and predicted scores; the Right graph is a histogram of topic proportions for each observed score category

The topic structure of the Economics item in Table 4.6 shows similar characteristics to the results of the LDA. Topic 1 has a coefficient of $\beta = -0.281$, which indicates that the examinees who mostly used words from Topic 1 tended to have a lower score. Topic 2 has a coefficient of $\beta = 0.400$, which

indicates the examinees who mostly use words from Topic 2 may get few points. Topic 3 has a coefficient of $\beta = 3.671$, which indicates that examinees who used mostly words from Topic 3 tended to have the highest scores.

Table 4.6: Top 15 highest probability words of the 4-topic sLDA model for the Economics item

Topic 1 ($\beta = -0.281$)		Topic 2 ($\beta = 0.400$)		Topic 3 ($\beta = 3.671$)	
part	0.485	interest	0.305	interest	0.203
simple	0.092	pay	0.070	compound	0.190
because	0.076	rate	0.068	amount	0.092
money	0.054	principal	0.067	year	0.068
save	0.033	simple	0.052	money	0.060
know	0.028	loan	0.042	time	0.056
get	0.027	retire	0.034	add	0.030
take	0.026	calculate	0.029	beneficial	0.029
make	0.018	good	0.020	over	0.023
bank	0.018	charge	0.017	principle	0.021
back	0.015	deposit	0.017	account	0.017
null	0.014	period	0.016	earn	0.017
little	0.010	investment	0.013	build	0.014
double	0.009	long	0.011	borrow	0.013
help	0.009	sum	0.011	end	0.013

Differences between the observed score and the predicted score from the sLDA model are shown in the scatter plot in the left graph of Figure 4.5. The mean for these differences is given by $\mu = \sum_{i=1}^{n} |y_i - \hat{y}_i| = 0.695$ and the standard deviation is 0.530. Similar to what we explained for the U.S. History item, the average difference of 0.695 between the observed human rater score and predicted score is relatively small to the score range and which indicates that the predicted scores are close to the obversed scores.

Similarly, after ranking examinees' observed scores according to each topic's proportion, we took the top 50 examinees' observed scores and present the frequencies of each score in the histogram in the right graph in Figure 5. Examinees who used words mainly from Topic 3 tend to have higher scores than examinees who used words mainly from Topic 1 or Topic 2. Few examinees who used more words from

Topic 3 had zero scores. A score of 4 does was not observed for examinees who used words mostly from Topic 1.

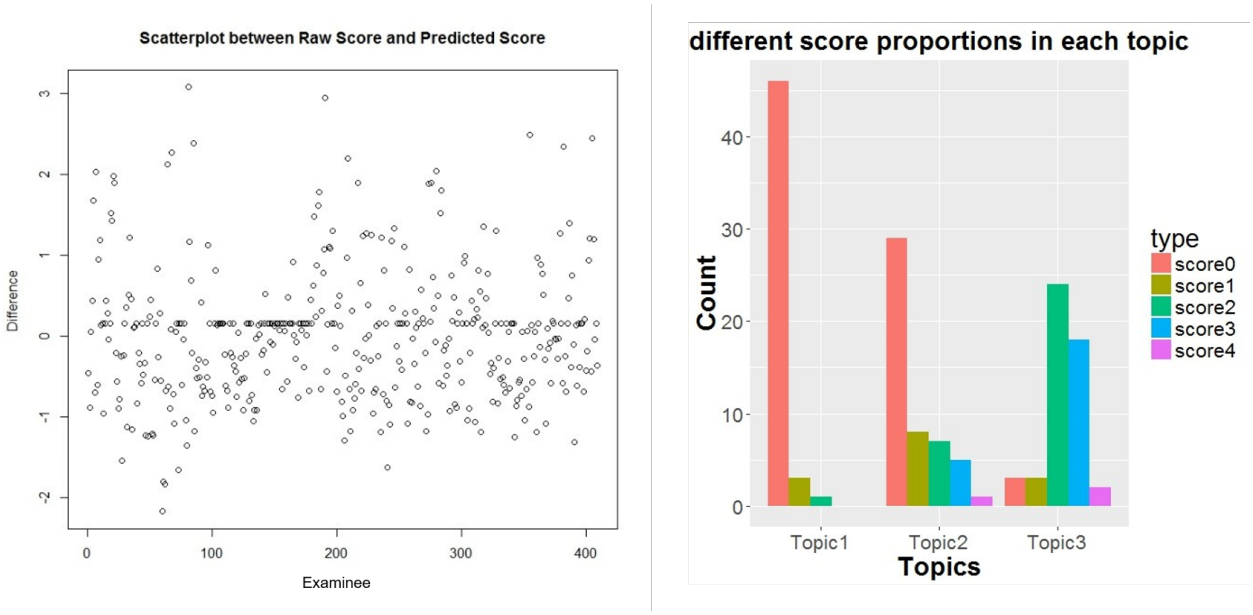

Figure 4.5: The left graph is a plot of the difference between the observed scores and predicted scores; the Right graph is a histogram of topic proportions for each observed score category

4.2.6 Discussion

The topic structures detected by LDA and sLDA show similar topic structures for the responses for each of the two items. Correlations for the U.S. History item between the observed score and the topic proportions from the LDA model indicate that the use of Topic 4 was modestly related to a higher score and the use of Topics 1 or 2 was more likely related to a lower score. The regression coefficients from the sLDA suggest a similar outcome as the use of words from Topic 4 was associated with a higher predicted score than the use of words from Topics 1 or 2. For the Economics item, correlations between topic proportions from the LDA and observed score suggested use of words from Topic 3 were moderately

related to higher scores and use of Topic 1 was moderately related to lower scores. Similarly, the use of words from Topic 3 was associated with a high predicted score and the use of words from Topic 1 was associated with a low score of effectively zero. Furthermore, what seems evident from the results from both assessments is that information about the latent thematic structure of the responses can extend what can be learned from the analysis of CR answers. The topic structure can also provide information on the examinees' thinking. For example, some topics in the example indicate that examinees simply copied the information in the stem or passages, and some topics indicate that examinees tried to use everyday words but integrated them with language from the passages to answer the questions.

4.3 Developing an Automated CR Answer Scoring Engine Using SLDA and A Generalized Logit Model

Constructed responses may be scored by human raters or through an automated scoring engine. Topic modeling provides a tool for mining textual data in an effort to detect the latent semantic structures. The supervised Latent Dirichlet Allocation model is one of the topic models widely used in text analysis. In the previous sections, we described use of the sLDA for a history test item and an economics test item. In this section, we examine the utility of different sLDA models for detecting the latent topic structure and scoring under a generalized logit model framework of an English American Literature test.

Onan et al. (2016) used LDA to extract topic proportions and constructed several different models including a logistic model and a support vector machine for sentiment analysis. In this section, the response variable is polytomous, therefore, a generalized logit model is used.

Generalized logit models describe the effects of covariates on the log odds of being in each category compared to some reference category, which is often taken to be the first or last category. In general, for a total of K category responses, the generalized logit model takes the form $y_i \sim Multinomial(\pi_i)$ and we assume the last category K as a reference level. This can be changed, however, to whichever score category might be desired. Here the logit models take the following Equation 4.6

$$\begin{cases} \log(\frac{\pi_{i1}}{\pi_{iK}}) = X_i^T B_1 \\ \log(\frac{\pi_{i2}}{\pi_{iK}}) = X_i^T B_2 \\ \vdots \\ \log(\frac{\pi_{i(K-1)}}{\pi_{iK}}) = X_i^T B_{K-1} \end{cases} \tag{4.6}$$

where X_i stands for the explanatory variable matrix and B_K represents model coefficients.

4.3.1 4.4.1. Constructed response item and its scoring

Constructed responses can be scored by human raters or through an automated essay scoring algorithm. Conventional human-rater scoring typically requires a rubric that clearly defines scoring procedures to maximize the reliability and the validity of the final scores (Hogan and Murphy, 2007). The ratings from different raters, however, could be prone to some error due to variation and discrepancies in rater training from one testing time to another (Ercikan et al., 1998). To minimize the differences between individual raters, this process usually requires rater training and monitoring of the score accuracy. Consequently, the associated time and expense involved in the scoring process are two important concerns in human scoring.

Compared with the human raters, the automated essay scoring algorithms have attracted many researchers due to their stable scoring results and their economical property. The accuracy and reliability of automated scores for writing assessments have been found to have a high agreement with human raters (Attali, 2004; Landauer et al., 1997; Nichols, 2005; Sebrechts et al., 1991). Traditional automated essay scoring algorithms depend on the linguistic features of the response content (Dzikovska et al., 2012; Livingston, 2009).

4.3.2 Data description

The data used in this example were written responses to an extended CR item from an English American Literature narrative assessment administered to high school 9th Grade examinees ($n = 1,273$). This item notes that surrealist artists often use symbolism and bizarre visual images to create dream-like landscapes in their artwork. As a result, the prompt asked examinees to imagine themselves inside a surrealistic painting, and to write a narrative describing their experience. The scores by human rater are used as the supervisor label for each response in the sLDA analysis. The scores for this item were ordered categorically and are summarized in Table 4.7. There were no extremely low or high numbers of examinees across the five score categories, so this training set could be considered as providing a relatively balanced distribution of labels.

Table 4.7: Number of scores in each score category of the extended CR item

	0	1	2	3	4
Count	351	221	273	318	109

Examinees' responses to the extended CR item were cleaned using the same procedure as described in the previous section. The remaining responses after cleaning were used in the sLDA model. Table 4.8 shows descriptive statistics for the number of words before and after the data cleaning process. It can be

seen that the number of words decreased after cleaning although only 9 answer documents were actually dropped from the sample.

Table 4.8: Descriptive statistics before and after the data clean

	Number of responses	Total words	Unique words
Before data clean	1,070	312,226	11,752
After data clean	1,061	131,659	6,544

4.3.3 Evaluation criteria

The classification accuracy (CA) was used again for evaluating the classification results. As described in Chapter 3, CA is defined as the fraction of correct predictions from the model. For a classifier with N classes, an $N \times N$ confusion matrix is created and a CA measure is calculated by Equation 4.7:

$$CA = \frac{\sum_i^N \sum_j^N n_{ij(i \neq j)}}{\sum_i^N \sum_j^N n_{ij}} \tag{4.7}$$

where the n_{ij} are the counts in the ith row and jth column in the matrix. In this study, the accuracy of the predicted scores by sLDA for the human raters' scores was of primary interest.

In addition to the CA, the other criterion used here was the Quadratic-Weighted Kappa ($QW - \kappa$; Fleiss and Cohen, 1973). The classical Kappa coefficient Landis and Koch (1977) proposed divisions on the Kappa coefficient and suggested the following intervals for interpretation: poor (≤ 0.00), slight ($0.00 - 0.20$), fair ($0.21 - 0.40$), moderate ($0.41 - 0.60$), substantial ($0.61 - 0.80$), and almost perfect ($0.81 - 1.00$). $QW - \kappa$ varies from 0 (trivial agreement between ratings) to 1 (complete agreement between ratings) and was used in this study to quantify the amount of agreement among multiple raters. For a given $N \times N$ confusion matrix, the $QW - \kappa$ score can be represented as in Equation 4.8

$$K_w = \frac{\sum_i \sum_j w_{ij} P_{ij} - \sum_i \sum_j w_{ij} P_{i\cdot} P_{j\cdot}}{1 - \sum_i \sum_j w_{ij} P_{i\cdot} P_{j\cdot}} \tag{4.8}$$

where the $w_{ij} = 1 - \frac{(i-j)^2}{(N-1)^2}$ are the quadratic weights and $P_{i\cdot}$ and $P_{j\cdot}$ are marginal probabilities of the ith row and jth column of the matrix, respectively. In machine learning, $QW - \kappa$ is typically used to measure the agreement between a human rater's label and an algorithm's prediction on the same observation. This paper adopted a $QW - \kappa$ threshold of 0.70 which suggests a high human-machine score agreement (Williamson et al., 2012).

Different n-grams were used as tokens in building up the various sLDA models because one of the n-gram models might be useful depending on the empirical data and sometimes the use of unigram, bigram, trigram, and their combination may yield a better result (Tripathy et al., 2016). N-gram means a contiguous sequence of n items from a given sample of responses. In this study, we estimated four models using four different n-gram sizes, namely, unigram, bigram, trigram, and mix-gram models since they were commonly seen in references (Beebe et al., 2013; Dey et al., 2018; Tripathy et al., 2016), where the mix-gram model used a combination of unigrams and bigrams. Each model's performance was compared over real data. The response length was also included in each of the four models as a covariate. These four n-grams were evaluated in terms of accuracy to predict the response label in sLDA.

4.3.4 Classification results

One of the steps in fitting an sLDA model is to determine the number of topics. More topics does not necessarily indicate better model fit. There are many model selection measures, such as the log-likelihood, deviance information criterion, and harmonic mean (Griffiths and Steyvers, 2004; Wallach et al., 2009;

Xiong et al., 2019), however, there is no standard method of selecting the number of topics in advance. Since the ultimate goal of the sLDA is prediction, this study considered the CA as a measurement criterion for selecting the optimal number of topics. Figure 4.6 presents CA results for each condition. It shows the optimal number of topics is not identical across the different n-gram models. The CA selects three topics for the unigram and bigram model, six topics for the trigram model, and five topics for the mix-gram model.

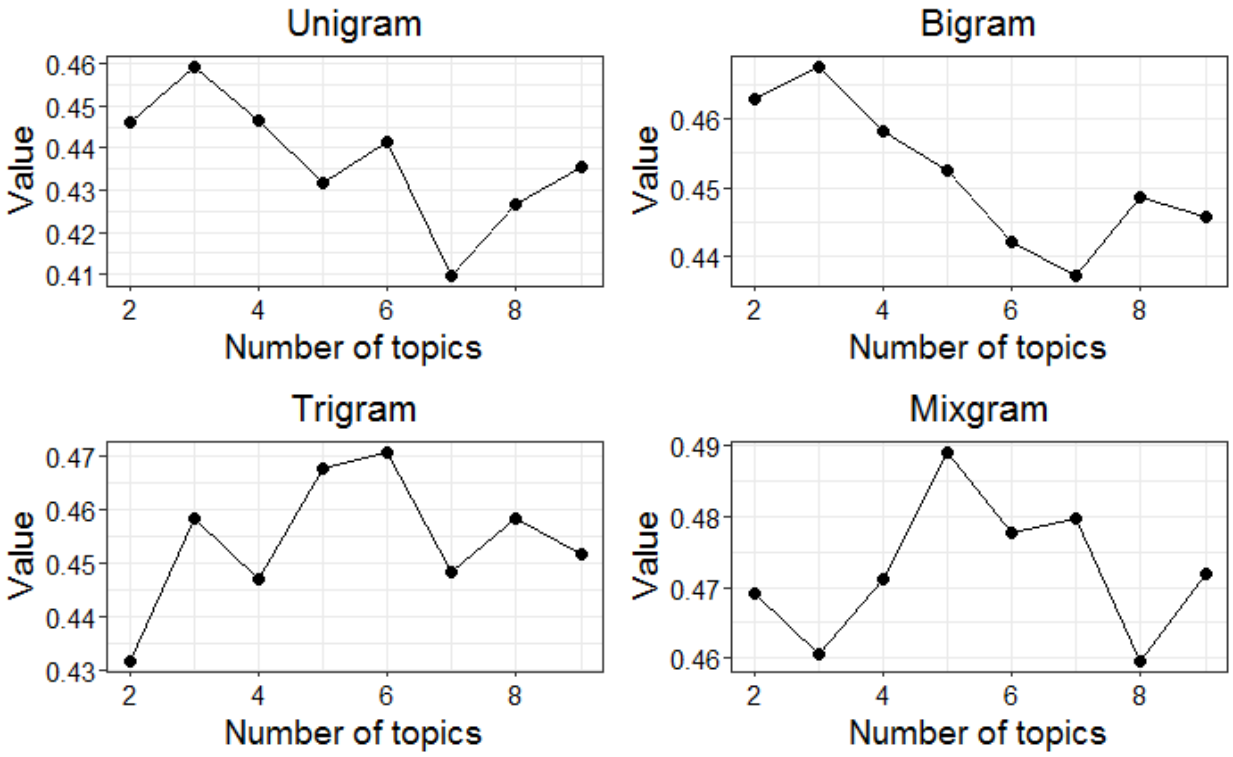

Figure 4.6: Classification accuracy yielded by the number of topics under each n-gram condition

After determining the number of topics, four separate augmented n-gram sLDA models were estimated using the sLDA topic proportions and scaled response length. The response length is the length of the examinee's response after data cleaning. The scaled response means that for a response length counted by the number of words in that response, we scale the length between 0 and 1, because the topic propor-

tions used in the generalized model range between 0 and 1. If we directly use the response length, the value of a response length could be much larger than the topic proportion, which may yield a very small coefficient for topic proportion. The sample of examinees' responses was randomly split into five folds (i.e., mutually exclusive subsets). For each model, four of the five folds were used as the training set and the remaining fold was used as the test set to measure the model's performance. This process was used repeatedly so that each fold was used as the test set once. Figure 4.7 presents the accuracy and $QW - \kappa$ scores from the 5-fold cross-validation in the four n-grams augmented models. All models used the scaled response length as a covariate within the generalized logit model.

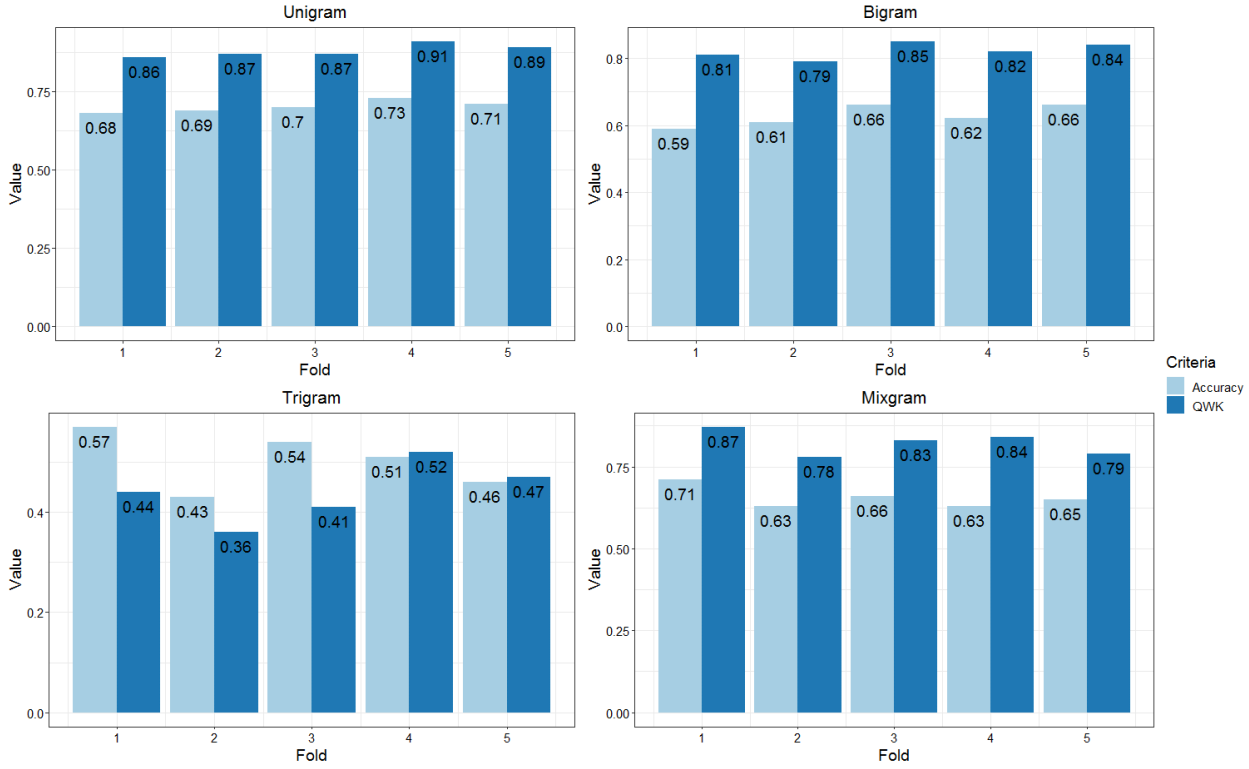

Figure 4.7: Different model classification accuracy and $QW - \kappa$ scores under optimal topic numbers

The average CA and $QW - \kappa$ scores from the five folds for each of the four models are summarized in Table 4.9. The Unigram model shows the highest CA and $QW - \kappa$ and the trigram model shows the lowest CA and $QW - \kappa$ scores.

Table 4.9: The average of classification accuracy and quadratic weighted kappa for different n-grams

Models	CA	$QW - \kappa$
Unigram	0.702	0.880
Bigram	0.628	0.822
Trigram	0.502	0.440
Mixgram	0.656	0.822

The classification accuracy indicated the unigram model was optimal, so the augmented unigram generalized logit model was fitted to all of the response data. The results from the multi-category sLDA model provided the following logits for each score category as shown in Equation 4.9

$$\begin{cases} \log(\frac{\pi_1}{\pi_0}) = -1.49\beta_0 + 1.98\beta_1 + 0.93\beta_2 - 1.99\beta_3 \\ \log(\frac{\pi_2}{\pi_0}) = 2.19\beta_0 + 7.09\beta_1 + 3.60\beta_2 - 2.78\beta_3 \\ \log(\frac{\pi_3}{\pi_0}) = 4.99\beta_0 + 8.73\beta_1 + 3.06\beta_2 - 9.76\beta_3 \\ \log(\frac{\pi_4}{\pi_0}) = 6.53\beta_0 + 7.928\beta_1 - 1.67\beta_2 - 16.65\beta_3 \end{cases} \tag{4.9}$$

where the $\pi_i (i = 0 \sim 4)$ is the probability of getting a score i; β_0 is the scaled response length; and $\beta_j (j = 1 \sim 3)$ are the topic proportions for each response. The overall accuracy from the unigram model is 0.702, and the $QW - \kappa$ score is 0.880, which surpasses the threshold of 0.70.

The confusion matrix in Table 4.10 shows the predictions against the human rater scores for each score category. The cells on the diagonal show the number of cases where the unigram model and human raters are in good agreement. We were also interested in the diagonal cells because they can provide information beyond the model precision, such as categorical sensitivity. Sensitivity means the number

of correctly classified numbers divided by the total classified number. For example, for responses that received a score of 3 by the human rater, the unigram model predicts 240 correctly, which indicates a 76% ($\frac{240}{240+56+21} = 76\%$) sensitivity for score 3. The unigram model classifies 49 responses into score 3 that were a human-rater score of 4, so sensitivity for score 4 is only 55% ($\frac{59}{59+49} = 55\%$), which means the unigram model may be less accuracy to higher scores than lower scores.

Table 4.10: The confusion matrix yielded by the unigram 3-topic sLDA model

	Human rater score				
Prediction	0	1	2	3	4
0	84	43	3	0	0
1	38	147	23	0	0
2	8	31	197	56	0
3	4	0	50	240	49
4	0	0	0	21	59

The examinees' responses can also be reflected in the unigram model topic structures table. Table 4.11 summarizes the top 10 words from each of the three topics in the unigram model. Since this item asked examinees to imagine themself inside a surrealistic painting, and to write a narrative describing their experience, Topic 1 can be identified as a topic related to the *Prompt of the Item*. For example, examinees may use words from this topic to begin with their experience description. Topic 2 can be understood as actions related with the *Experience*, and Topic 3 can be identified as containing words about the *Surrealistic Painting*.

4.3.5 Discussion and conclusion

This study proposed an automated scoring engine using the sLDA and generalized logit model as the foundation. The sLDA uses a supervisor variable that estimates latent topics to help understand the examinee's written response in relation to the supervisor label. A critical question in this study was to find

Table 4.11: Unigram model topic structures

Topic 1		Topic 2		Topic 3	
clock	0.018	paint	0.024	paint	0.064
see	0.016	see	0.021	art	0.027
around	0.013	know	0.014	surrealist	0.022
walk	0.010	think	0.012	world	0.017
eye	0.009	wake	0.011	artist	0.015
feel	0.009	ask	0.010	feel	0.013
melt	0.008	start	0.010	mean	0.012
begin	0.007	walk	0.009	movement	0.010
myself	0.007	come	0.008	time	0.010
open	0.007	dream	0.008	surrealism	0.009

the appropriate token dimension to represent the item response. Four different n-gram tokens, namely, unigram, bigram, trigram, and mix-gram were used to compare model performance. The classification accuracy was used as a criterion to select the best number of topics for each sLDA model. Four augmented generalized logit model models based on the n-gram tokens and scaled response length were built and compared.

The results from the empirical data showed that the sLDA and generalized logit model with unigram performed best with the highest human-machine score agreement. The models were tested further using the 5-fold cross-validation. Each model incorporated a covariate for the response length. Among these four models, the unigram, bigram, and mix-gram models yield similar model precision, but the unigram sLDA model showed the highest classification accuracy based on a 0.880 $QW - \kappa$ score. The overall CA from the unigram model was 0.702, however, the classification sensitivity for the perfect scores was not ideal as we discussed above, which suggests the unigram model might be less accuracy to higher scores than lower scores. Future studies could consider word embedding or suchlike to overcome the problem. The model could also be further pruned to yield higher accuracy by adding effective features.

4.4 A Hybrid Framework Using Rasch Measurement Models and Topic Model

In this study, an analytic framework is proposed to assess the reliability of a mixed-format assessments. The mixed format assessment consisted of responses to multiple-choice and CR items. A five-step framework is proposed and an empirical dataset from a Grade 8 English Language Arts test was used for illustration.

4.4.1 Analytic framework

In this study, we propose an analytic framework with five steps.

1. Step 1: examine the dimensionality of the assessment by applying exploratory factor analysis to the mixed-format score data;
2. Step 2: select and apply candidate measurement models to assess the internal structure of the mixed-format data;
3. Step 3: evaluate model-data fit to determine the best model for measuring the underlying constructs;
4. Step 4: explore examinees writing with topic models to uncover their writing structures;
5. Step 5: use the resulting topic structures to help interpret calibration results.

Exploratory factor analysis (EFA; Fabrigar and Wegener, 2011) was used to determine the dimensionality of the score data. Each latent factor indicates a dimension in the EFA. The eigenvalues for each of the latent factors were calculated and used to aid in determining the best number of factors for the data.

This study uses parallel analysis (Horn, 1965) to evaluate the best number of factors. The parallel analysis uses the number of eigenvalues that are larger than those which result from factoring random data. In other words, the parallel analysis compares the eigenvalues generated from the data matrix to the eigenvalues generated from a Monte-Carlo simulated matrix created from random data of the same size. Longman et al. (1989) recommended comparing the original data's eigenvalues with 95th percentile from the parallel samples for the sake of determining the best number of factors.

4.4.2 The partial credit model and the bi-factor model

The partial credit model (PCM; Masters, 1982) is a unidimensional IRT Model that can be used with polytomous items. In this regard, polytomous responses to items are viewed as ordered as $0, 1, 2, \ldots, k_i$. Each examinee j is assumed to have an ability θ_n, and each item is assumed to have a set of k_i parameters $\delta_{i1}, \delta_{i2}, \cdots, \delta_{ik_i}$, each of which can be represented as a location on the latent variable being measured (θ). The parameter δ_{ik} indicates the probability of scoring k rather than $k-1$ on item i. The PCM assumes that an examinee's ability of scoring k rather than $k-1$ is independent of all other possible outcomes. The model can be specified in Equation 4.10

$$\ln\left[\frac{P_{j(x_i=k)}}{P_{j(x_i=k-1)}}\right] = \theta_j - \delta_{ik} \tag{4.10}$$

where the $P_{j(x_i=k)}$ represents the probability of examinee j receiving a score k on item i.

The bi-factor model (Md Desa, 2012) is a confirmatory factor model with a specification of two dimensions. The model assumes that every item is dominated by two types of factors, a general factor and a specific factor. This general factor influences all items, and the other specific factors affect different and

mutually exclusive groups of items. In addition, all the specific factors are orthogonal with each other and with the general factor. Therefore, the model restriction requires that each item load on a primary dimension of interest and no more than one secondary dimension or subdomain. The secondary dimension can be nuisance variable such as the content domain from which the items are sampled. For example, Tanaka et al. (2020) used the bi-factor model to measure the food insecurity from Household Food Security Survey Module (Bickel et al., 2000) data, and the general factor used in that study referred to household food insecurity while the specific factors were about adult and child food insecurity, respectively.

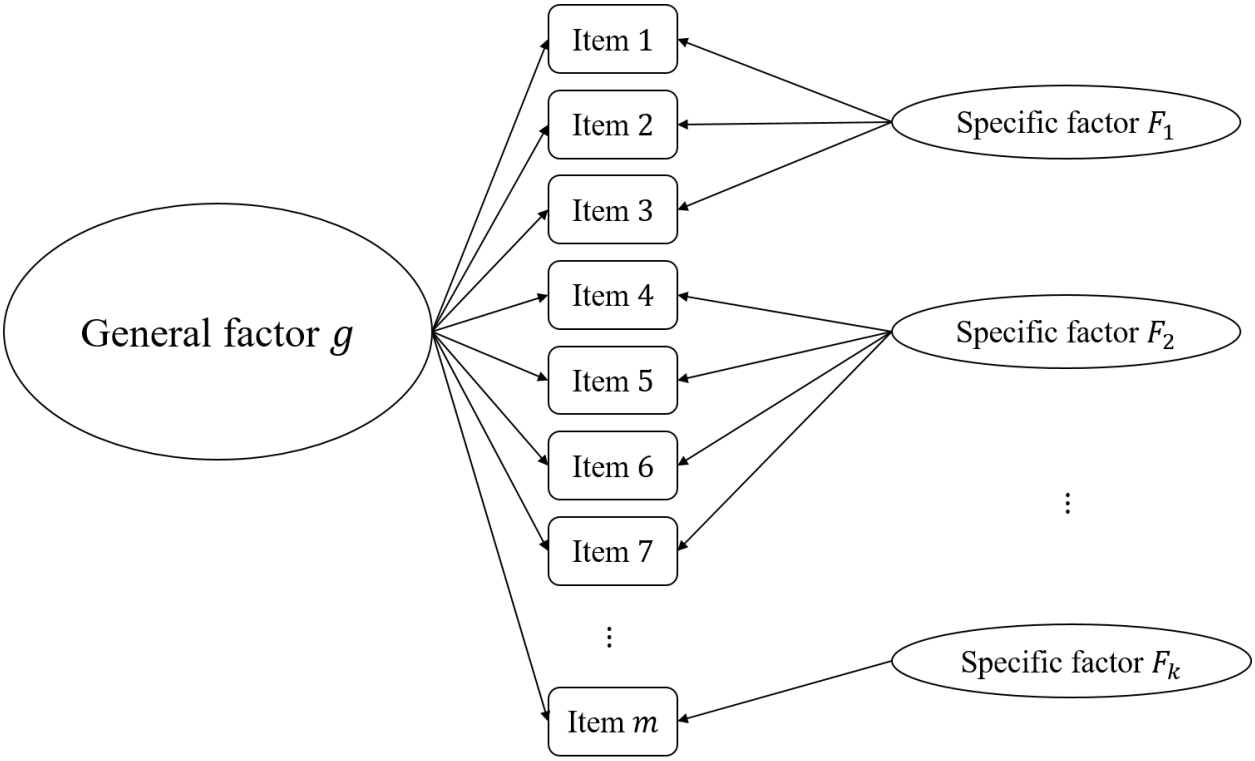

Figure 4.8: A structure illustration of the bi-factor model

A general structure of a bi-factor model is shown in Figure 4.8. F_k's are specific factors and g denotes the general factor. The general factor is measured by all the items while the specific factors are measured by a subset of one or more items possibly within the same content domain. The general factor and the

specific factors are statistically independent in the bi-factor model. The bi-factor analytic model can be expressed in Equation 4.11:

$$P(X_i|\theta_{jg}, \theta_{js}) = \alpha_{ig}\theta_{jg} + \alpha_{is}\theta_{js} + \delta_{ik} \tag{4.11}$$

where θ_{jg} means examinee j's ability on the general factor, θ_{js} means examinee j's ability on the specific factor, α_{ig} is item i coefficient on the general factor, α_{is} means item i coefficient on the specific factor, and δ_{ik} stands for step k coefficient to item i. When the bi-factor model is constrained, referred to as the constrained bi-factor model, the item slope parameters are fixed to be equal within the general (domain) and specific (sub-domains) factors (Tanaka et al., 2020).

4.4.3 Fit indices

Fit indices were used to compare the model fit of candidate models. For example, the item and person fits of the PCM are evaluated using Infit and Outfit statistics. Infit and Outfit statistics are Rasch-based individual-level fit indices. They are based on residuals and quantify the distance between observed proportions and model-based probabilities. The Outfit statistic is outlier sensitive and the Infit statistic is sensitive to unexpected response patterns. The expected value for both Infit and Outfit statistics is 1.0. Infit and Outfit can be expressed as follows in Equations 4.12 and 4.13 (Wright and Masters, 1982)

$$Outfit = \sum_{n=1}^{N} \frac{z_{ni}^2}{N} \tag{4.12}$$

$$
\begin{aligned}
Infit &= \frac{\sum_{n=1}^{N} z_{ni}^2 w_{ni}}{\sum_{n=1}^{N} w_{ni}} \\
&= \frac{\sum_{n=1}^{N} y_{ni}^2}{\sum_{n=1}^{N} w_{ni}}
\end{aligned}
\tag{4.13}
$$

where N is the number of examinees, x_{ni} is the observed response, $y_{ni} = x_{ni} - E_{ni}$ stands for score residual while E_{ni} is the expected mean of x_{ni}, W_{ni} represents the variance of x_{ni}. and $z_{ni} = \frac{y_{ni}}{\sqrt{w_{ni}}}$ means the standardized residual.

In addition, Akaike's information criterion (AIC) and Bayesian information criterion (BIC) were used to inform model selection. Models with lower AIC and BIC values may indicate better model-data fit (Burnham and Anderson, 2004). Stucky and Edelen (2014) suggested comparing the general factor to the specific factors to identify subsets of items for which the multidimensionality might be weak enough to ignore. In this study, this comparison is achieved by indicators called explained common variances (ECVs; Bentler, 2009). ECVs can be calculated using the estimated factor loadings of the general and specific factors of a bi-factor model, the general form of an item ECV can be given as

$$
ECV = \frac{\sum \lambda_g^2}{\sum \lambda_g^2 + \sum \lambda_s^2} \tag{4.14}
$$

where λ_g is the factor loading for the general factor, and λ_s is the factor loading for the specific factors. Based on ECV values, recommendations can be made for choosing between unidimensional or multidimensional models, as well as whether sub-domain scores have added value over a total domain score. Estimated results that have a large general factor loading compared to specific factors have high ECV. Another commonly used ECV value, ECV_SG, stands for ECV of a specific factor concerning the general

factor, was simply referred to as specific-dimension ECV (Stucky and Edelen, 2014). The ECV_SG is computed for the items of the general factor using the specific factor loadings in the numerator in Equation 4.14. The analysis was done using the R package mirt (Chalmers, 2012).

4.4.4 Data description

The ELA mixed-format data used in this study are from a formative assessment designed to measure examinees' understanding of the information in a passage and usage of the information in the written response. The assessments are aligned to the state standards of a Southeastern state in the U.S. and are used in statewide assessments for end-of-grade and end-of-course assessments.

The assessment was administered to $5,986$ examinees in local schools in the state. The assessment contains one reading passage with five questions. There were three MC items and two CR items in the assessment: One of the CR items was a short-answer item and the other an extended response item. The MC items assess examinee comprehension of the reading passage and were scored dichotomously (i.e., 0 for incorrect and 1 for correct). The short-answer CR item was scored from 0 to 2. The extended response item was scored from 0 to 4. A summary of the five questions and the rating categories for this assessment is given in Table 4.12.

Table 4.12: Description of items on ELA assessment

Items	Description	Item format	Score type	Category
1	MC1	Selected-response item	Binary	0, 1
2	MC2	Selected-response item	Binary	0, 1
3	MC3	Selected-response item	Binary	0, 1
4	CR1	Constructed-response item	Polytomous	0, 1, 2
5	CR2	Extended writing prompt	Polytomous	0, 1, 2, 3, 4

4.4.5 Results

Step1: Examine dimensionality of assessment

Figure 4.9 indicates the result of the EFA based on a parallel analysis. Parallel analysis is used for determining the number of factors to retain from factor analysis. It creates a random dataset with the same size and same number of variables as the original data. After that, a correlation matrix is computed from the randomly generated dataset and then eigenvalues of the correlation matrix are computed (Franklin et al., 1995). The 95% of the generated data eigenvalue and average eigenvalue are compared to the original data eigenvalue. The comparison between these eigenvalue indicates that a single factor seems to be supported for this data set and the two-factor model is right about the same as the 95% and Average eigenvalue lines. Next, a specific exploratory study showing the factor loadings of the one-factor model and the two-factor model was conducted to determine the optimal number of factors.

Table 4.13: Factor loadings for two EFA models

	One-factor model	Two-factor model	
Items	Factor 1	Factor 1	Factor 2
MC1	0.032	0.998	-0.002
MC2	0.205	0.010	0.302
MC3	-0.074	0.006	-0.474
CR1	0.653	-0.003	0.654
CR2	0.622	0.000	0.621

Factor loadings from the two EFA models are given in Table 4.13. Factor loadings indicate that the first and third items do not load on the single factor very well. The factor loading for the first MC item is 0.032 while the factor loading for the third MC item is -0.074. On the other hand, the factor loadings for the two-factor model show that the first MC item primarily loads on the first factor and the other items load on the second factor, although the MC2 and MC3 loadings are lower than the two CR items

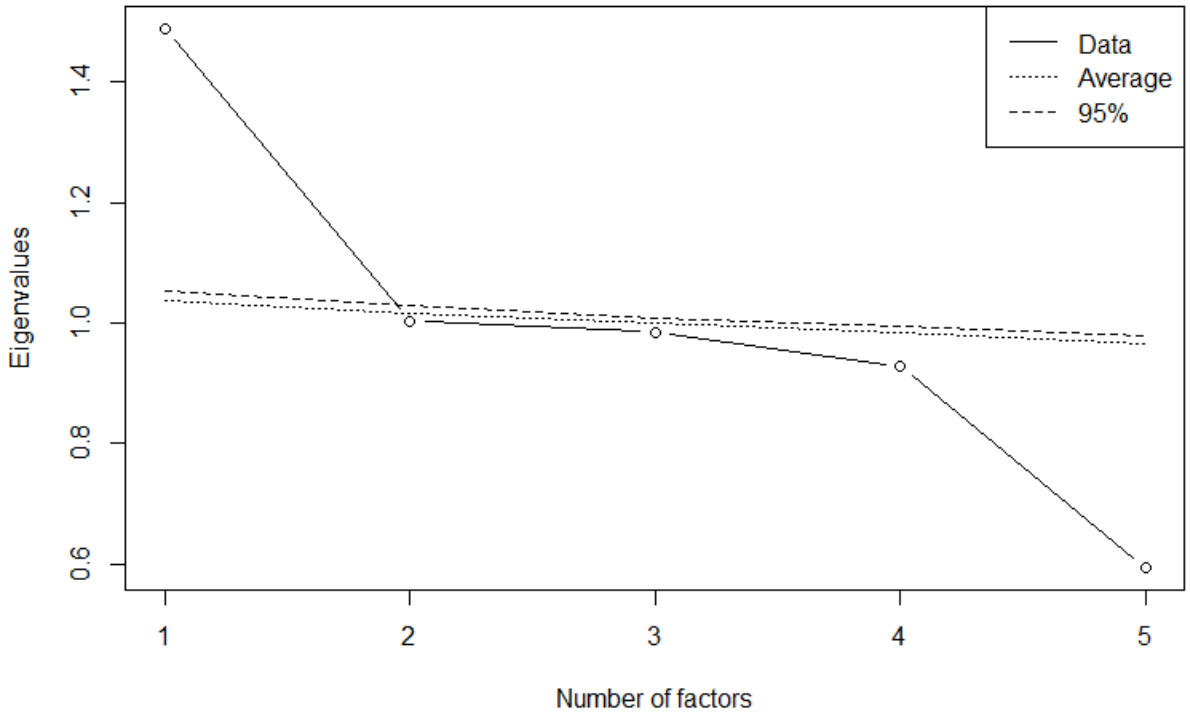

Figure 4.9: Parallel analysis of the ELA data

on the second factor. The correlation between the two factors is -0.035 which indicates that two factors are independent. If we adopt 0.3 as a low factor loading criterion based on Shevlin and Miles (1998), then the two-factor model shows better factor loading than the first one-factor model because under one factor, the loading of item MC3 is lower (-0.074) but all the loadings of all items are higher than 0.3 or lower than -0.3 when using two factors. Therefore, in the following steps, the PCM, bi-factor model, and constrained bi-factor model, are fitted to the mixed-format data.

Step2: Apply candidate measurement models

The item parameter estimates for the PCM are listed in Table 4.14. The first column shows the discrimination parameter, which is equal to "1" for all items in the PCM. In the following columns, we see the estimated item difficulty (d), and thresholds or step parameters ($\delta_{i1}, \delta_{i2}, \delta_{i3}, \delta_{i4}$). From this table, MC3 is the most difficult item and CR2 step 1 is the easiest one.

Table 4.14: Item parameter estimates from the PCM

	a	d	δ_{i1}	δ_{i2}	δ_{i3}	δ_{i4}
MC1	1	0.450	-	-	-	-
MC2	1	-1.577	-	-	-	-
MC3	1	1.997	-	-	-	-
CR1	1	-	0.010	0.329	-	-
CR2	1	-	-2.222	-0.109	0.624	2.102

The Infit and outfit statistics for the PCM are shown in Table 4.15. From this table, we can see the mean ability measure for the examinees is 0.00 with a standard deviation of 0.38. The Outfit and Infit MS statistics are residual-based statistics with an expectation of 1.0. The mean Infit is 0.89 with a standard deviation of 0.66, while the Outfit is 1.17 with a standard deviation of 1.58. As for the item measure summary, the mean item measure for the 5 items is 0.178 with a standard deviation of 1.425. The mean Infit is 1.00 with a standard deviation of 0.21, while the Outfit MS is 1.30 with a standard deviation of 0.56.

Table 4.15: Summary statistics from PCM

		Examinee	Item
Measure	Mean	0.000	0.178
	SD	0.380	1.425
Infit	Mean	0.890	1.000
	SD	0.660	0.210
Outfit	Mean	1.170	1.300
	SD	1.580	0.560

The constrained bi-factor model in this example considers a general factor (English language proficiency) that is measured by all the items and two specific factors (Reading proficiency and writing proficiency) with each factor being measured by items in different formats (MC and CR). The item slope parameters on the general factor are equal, which means $a_{11} = a_{21} = a_{31} = a_{41} = a_{51}$. The first three MC items measure the first specific factor, and their slope parameters are constrained to be equal. The two CR items were designed to measure the second specific factor also with equal slope parameters. That is, $a_{12} = a_{22} = a_{32}$ and $a_{43} = a_{53}$. Therefore, the constrained bi-factor model employed in this study can be illustrated using the matrix B as shown in Equation 4.15,

$$B = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 0 \\ a_{41} & 0 & a_{43} \\ a_{51} & 0 & a_{53} \end{bmatrix} \tag{4.15}$$

The estimated coefficients from the bi-factor and constrained bi-factor model are given in Table 4.16. The bi-factor model has varying item slope coefficients for each item. The MC3 has a negative coefficient value. This suggests caution in accepting these results as the probability of endorsing the correct response should not decrease as the examinee's ability increases. Considering that MC3 is the most difficult item, it may be that this item is measuring something other than what the rest of the assessment is measuring. In other words, this item may need to be dropped from the assessment. For the constrained bi-factor model, the item slope estimate is 0.388 for the general factor, 0.002 for the first specific factor for MC items, and 1.632 for CR items. The item step coefficient estimates for MC items indicate the difficulty of reaching

category 1 from 0. For CR items, the difficulty estimate of each category reflects how difficult it is to reach the next adjacent category. For both models, easier items have higher step difficulty estimates.

Table 4.16: Item coefficients from the bi-factor and constrained bi-factor model

		α_1	α_2	α_3	d	δ_{i1}	δ_{i2}	δ_{i3}	δ_{i4}
Bi-factor mode	MC1	0.074	0.408	-	-0.431	-	-	-	-
	MC2	0.664	0.098	-	1.599	-	-	-	-
	MC3	-0.248	0.090	-	-0.893	-	-	-	-
	CR1	1.749	-	1.008	-	0.805	-1.479	-	-
	CR2	1.328	-	0.644	-	3.722	0.580	-1.434	-3.976
Constrained bi-factor model	MC1	0.388	0.002	-	-0.429	-	-	-	-
	MC2	0.388	0.002	-	1.513	-	-	-	-
	MC3	0.388	0.002	-	-1.922	-	-	-	-
	CR1	0.388	-	1.632		0.732	-1.333	-	-
	CR2	0.388	-	1.632		3.939	0.625	-1.526	-4.209

4.4.6 Step 3: Evaluate model-data fit

Statistics listed in Table 4.17 show that the bi-factor model and the constrained bi-factor model fit better than the PCM. The bi-factor model (AIC=46680.66; BIC=46807.90) provides the best fit compared with the constrained bi-factor model (AIC=46833.54; BIC=46913.91). This suggests further examination might be appropriate for the bi-factor model for the ELA measurement.

Table 4.17: Model-fit statistics for ELA measurement models

	AIC	BIC
PCM	47384.90	47451.87
Bi-factor model	46680.66	46807.90
Constrained bi-factor model	46833.54	46913.91

Estimates of factor-loading parameters for the PCM and two bi-factor models are presented in Tables 4.18 and 4.19. The h_2 statistic represents the factor communality estimates. The communality is defined as the proportion of observed variance due to common factors. The communality for the ith item is computed by taking the sum of the squared loadings for that item. For example, the factor loadings for all

items are 0.308, then communalities for all items were $0.308^2 = 0.0949$. The communality for an item can be interpreted as the proportion of variation in that item explained by the factors. For instance, the 0.0949 for each item by the PCM indicates that about 9.49% of the variation in each item was explained by the factor model. The result suggests that the single factor does not explain variation well in each item.

Table 4.18: Single-factor loadings from PCM for ELA data

	Factor	h
MC1	0.308	0.0949
MC2	0.308	0.0949
MC3	0.308	0.0949
CR1	0.308	0.0949
CR2	0.308	0.0949

The factor loadings for each item and ECV values from the two bi-factor models are displayed in Table 19. The first columns under both models represent the general factor loadings, while the second and third column represents the two specific factor loadings, respectively. Although the constrained factor loadings were expected to be equal across the 5 items on the general factor, there was variation related to the specific factors which influenced the estimation of general factor loadings. The ECV_SG can be interpreted as the proportion of common variance of all items which is due to the specific factor. In the bi-factor model, the ECV_SG for the general factor and two specific factors were 76.6%, 4.9%, and 18.6%, respectively. However, the constrained bi-factor model had only an ECV_SG of 17.7% for the general factor, and 0% and 82.3% for the two specific factors, respectively. The result from the bi-factor model indicates there is a strong general factor, which appears to represent English language proficiency, and there is additional evidence for interpretable specific factors representing reading and writing ability. But the results from the constrained bi-factor model show clear multidimensionality of the ELA data. The ECVs indicating how well each item represents the general factor are also listed in this Table. A low ECV value indicates there is a strong association between that item and that specific factor (reading or

writing), and a high ECV value reflects a strong association. In the bi-factor model, except for the MC1, all other items were highly associated with the general factor. In the constrained bi-factor model, the MC items were highly associated with the general factor and the CR items were highly associated with the specific factor (writing).

Table 4.19: Factor loadings from two bi-factor models for ELA data

	Bi-factor model					Constrained bi-factor model				
	g	S_1	S_2	h_2	ECV	g	S_1	S_2	h_2	ECV
MC1	0.042	0.233	-	0.056	0.032	0.222	0.001	-	0.050	0.999
MC2	0.363	0.053	-	0.135	0.979	0.222	0.001	-	0.050	0.999
MC3	-0.144	0.052	-	0.024	0.884	0.222	0.001	-	0.050	0.999
CR1	0.662	-	0.382	0.585	0.750	0.163	-	0.683	0.493	0.054
CR2	0.589	-	0.286	0.429	0.809	0.163	-	0.683	0.493	0.054
ECV_SG	0.766	0.049	0.186			0.177	0.000	0.823		

The comparisons of the ability measures from three measurement models are shown in Table 4.20. Under each model, the mean ability measures are close to 0. The standard deviation for the general factor in the bi-factor model is the largest ($SD = 0.69$). The standard deviation for the reading factor in the constrained bi-factor model is the smallest ($SD = 0.001$), which indicates that examinees' latent traits for the reading dimension are almost around 0. The reliability of the writing factor under the constrained bi-factor model is the highest.

Table 4.20: Ability measure summary from three measurement models

	Factor	Mean	SD	Reliability
PCM	-	0.000	0.381	0.393
Bi-factor model	General factor	-0.000	0.690	0.476
	Reading factor	-0.000	0.198	0.039
	Writing factor	-0.000	0.370	0.137
Constrained bi-factor model	General factor	0.000	0.310	0.096
	Reading factor	-0.000	0.001	0.000
	Writing factor	-0.000	0.740	0.548

4.4.7 Step 4: Topic model selection

DIC was used to determine the number of topics. Figure 4.10 shows the DIC change against the number of topics from 2 to 10 to the CR answers. The model with 5 topics had the lowest DIC value. The results suggest the 5 topic model was the best fit to the ELA data of the models considered.

The top 20 highest probability words for each of the five topics are given in Table 4.21. Topic 1 and Topic 5 can be both characterized as *Integrative Borrowing*. This indicates these topics contain responses that examinees used from the passage to support their argument. Topic 2 is *Everyday Language*. Use of this topic indicated that examinees used everyday language in their responses. Responses of examinees who used this topic indicated lack of necessary details and evidence relative to what was requested in the prompt. Topic 3 and Topic 4 can be viewed as *Simply Borrowing of Words*. This means that examinees simply borrowed vocabulary from the passage or stem rather than integrating them in a way that supported their arguments.

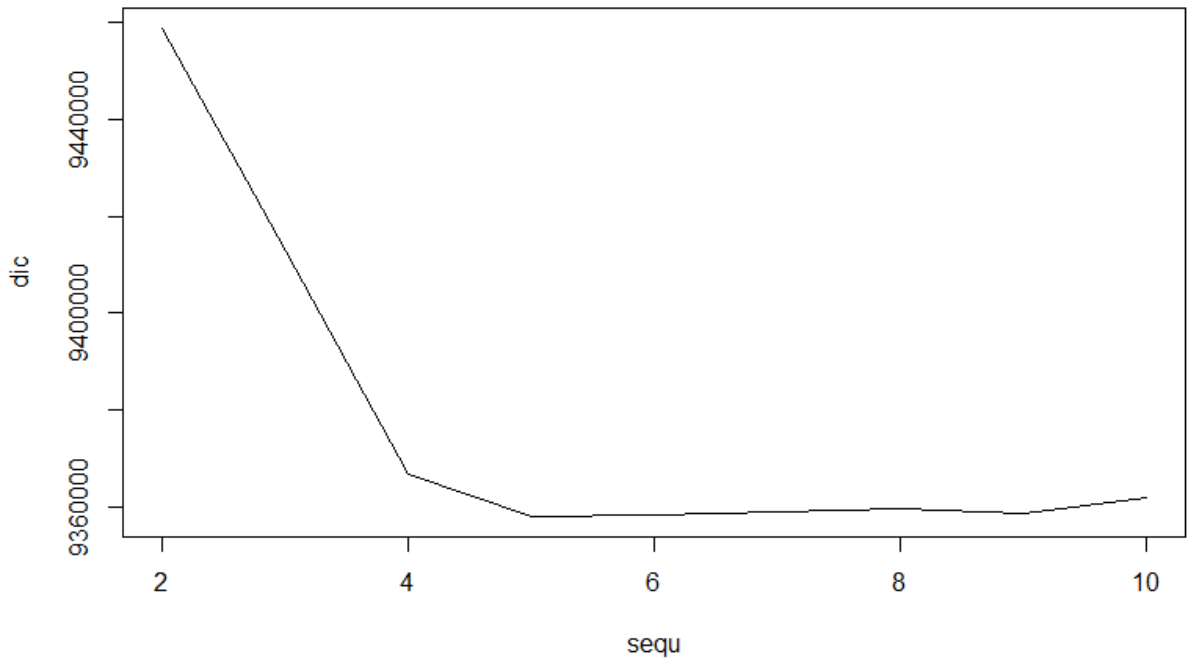

Figure 4.10: DIC index on different number of topics

Correlations with the scores assigned by human rater scores were calculated for each topic as a way of further characterizing the topics. Topic 5 was characterized as reflecting *integratively borrowing*. This topic is positively related to the human rater scores: Topic 5 had a correlation of $r = 0.3802$). The topics indicating *simplely borrowing* of words were Topic 1, Topic 3 and Topic 4. These three topics had correlations almost close to 0 with the scores such that Topic 1 had a correlation of $r = 0.03$, Topic 3 had a correlation of $r = -0.0597$, and Topic 4 correlates $r = -0.0474$ with the scores. These values may indicate that the proportion of this topic may not have strong correlation to the response score. Topic 2

Table 4.21: LDA topic structure with top 20 words for the ELA CR answer

Topic 1		Topic 2		Topic 3		Topic 4		Topic 5	
see	0.039	jessie	0.059	jared	0.080	friend	0.022	jared	0.020
door	0.037	say	0.045	friend	0.074	school	0.022	like	0.017
change	0.036	glass	0.021	jessie	0.045	class	0.021	look	0.017
jessie	0.028	jared	0.018	new	0.032	tell	0.021	say	0.017
new	0.024	see	0.018	come	0.024	summer	0.019	think	0.016
story	0.023	hear	0.016	say	0.023	creek	0.018	girl	0.015
character	0.019	walk	0.016	leave	0.023	house	0.017	come	0.015
friend	0.018	help	0.015	get	0.021	mud	0.017	get	0.013
dynamic	0.017	get	0.014	summer	0.020	leave	0.017	back	0.012
back	0.017	day	0.014	want	0.020	walk	0.015	feel	0.009
hair	0.015	come	0.012	good	0.018	day	0.015	know	0.009
dress	0.014	other	0.012	back	0.017	begin	0.014	friend	0.009
wear	0.014	time	0.012	start	0.016	sit	0.013	walk	0.009
talk	0.014	tell	0.011	school	0.015	other	0.013	want	0.009
make	0.013	leave	0.011	york	0.015	seat	0.013	good	0.009
jared	0.012	hand	0.010	talk	0.015	together	0.012	make	0.008
show	0.012	school	0.009	like	0.014	only	0.012	door	0.007
open	0.012	kid	0.009	tell	0.012	behind	0.011	time	0.007
summer	0.011	door	0.009	again	0.012	cool	0.011	even	0.007
know	0.010	bike	0.009	mad	0.011	next	0.011	thing	0.007

was characterized as indicating use of *everyday language*. It had a negative correlation to the score as well ($r = -0.2981$).

The probability of use of each topic can be provided for each examinee. This means the structure of the response could be vectorized into 5-dimension vectors, in which each dimension denotes the words proportion of use of every topic. For example, one examinee's topic proportion vector was $(0.73, 0.06, 0.06, 0.09, 0.06)$. This indicates that the examinee's writing largely consisted of use of words from Topic 1 with use evenly distributed over the other four topics.

4.4.8 Step5: Interpretation of LDA results with constrained bi-factor model calibration results

The correlations of the topics with the human rater scores can help to interpret the topic meaning of the topics. In this regard, it reflects a possible way to associate the topic structure with the estimate of ability. As indicated in the previous step, the constrained bi-factor model produced three sets of ability levels, one for the general ability (English proficiency), one for the specific ability 1 (reading proficiency), and one for the specific ability 2 (writing proficiency). The estimated abilities from the constrained bi-factor model reflects examinees' abilities on each latent scale. The scores for the CR items were shown to be associated with specific ability 2, writing proficiency.

In the estimation results of the constrained bi-factor model, the minimal writing proficiency value was -1.572 and the maximum was 1.673. The specific ability 2 by quartile is as follows: $Q1 : (-1.572, -0.390)$, $Q2 : (-0.390, 0.098)$, $Q3 : (0.098, 0.566)$, $Q4 : (0.566, 1.673)$. Figure 4.11 plots the distribution of topic proportions for Topic 2 and Topic 5 for each quartile. Topic 2 and Topic 5 are topics that were modestly related to the raw CR scores: Topic 2 has the lowest correlation r=-0.2981 and Topic 5 has the highest correlation $r = 0.3802$. Since Topic 5 proportions represent the integrative borrowing of words to certain topics, a larger proportion of use of Topic 5 would be likely to be associated with higher scores. A larger proportion of use of Topic 2, on the other hand, would be likely to be associated with lower scores. Figure 4.11 presents the distributions of all examinees' Topic 2 and Topic 5 proportions within each ability quaitile, and we see that proportion of use of Topic 5 increasing from $Q1$ to $Q4$. Use of Topic 2 showed a decreasing trend from $Q1$ to $Q4$.

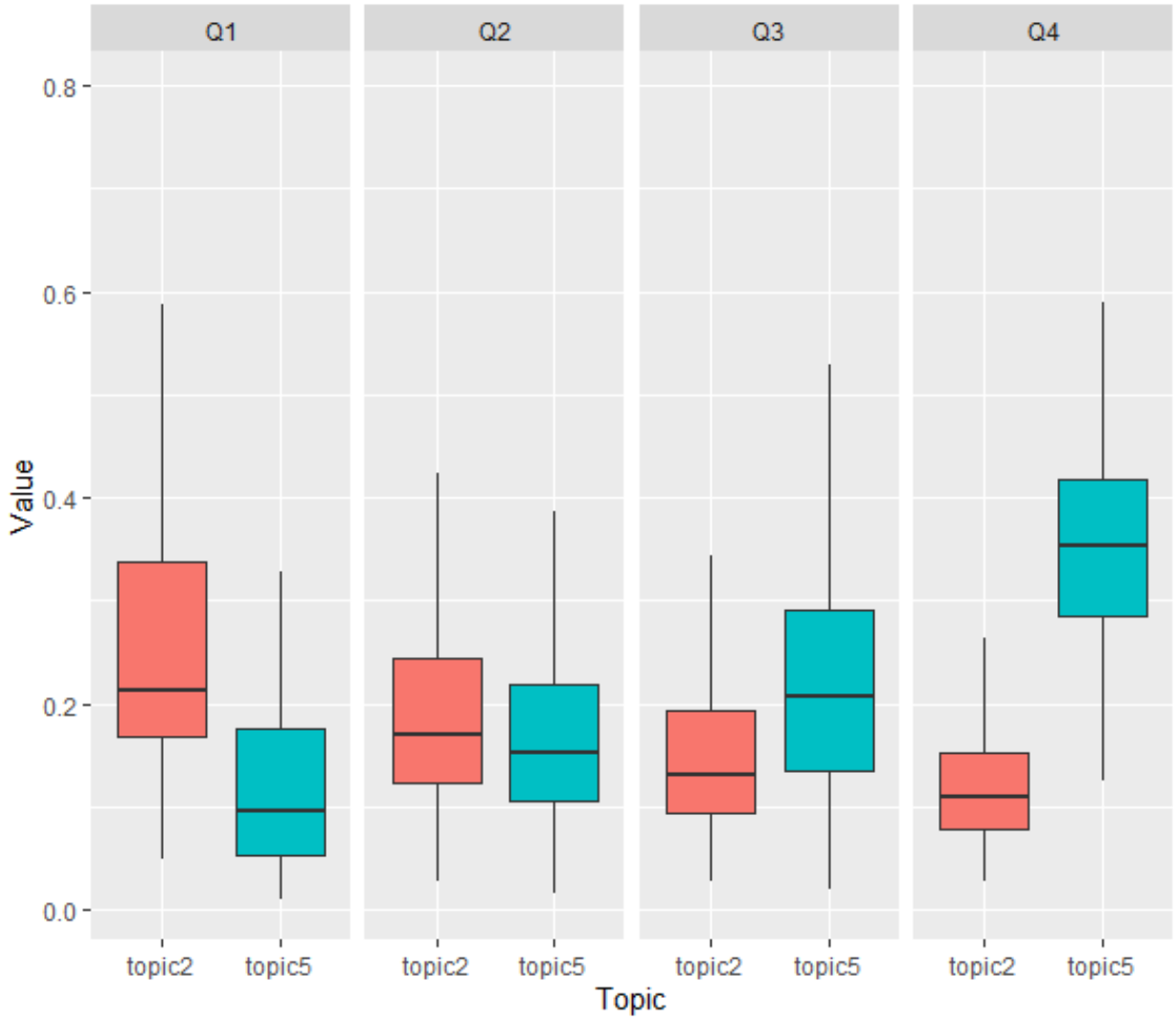

Figure 4.11: Two topic distributions in each ability category

4.4.9 Discussion and conclusion

This study proposed a hybrid framework consisting of a five-step procedure to evaluate the mixed-format score patterns and textual data. This framework was illustrated with data from a mixed-format ELA assessment. The unidimensional PCM and multidimensional bi-factor models provided a potentially useful framework for exploring the dimensionality of scores from the assessment and for a more detailed approach for evaluating the factor loadings and reliability of the assessment. The PCM is sometimes used when a single construct is being measured in an assessment with mixed-format items as it can accommodate polytomous scoring of item responses. The bi-factor models, including the measurement of a general factor and more than one specific factor, may have better factor loadings and model fit with multidimensional data. A factor analysis indicated a unidimensional test.

In the ELA assessment, the MC items measured reading proficiency (specific factor 1), and the CR items measured writing proficiency (specific factor 2). A general factor was interpreted to be English language proficiency. Results suggested the bi-factor model was a better fit for this assessment. Item analysis results based on the PCM indicated that item MC3 did not fit the model very well. A topic model analysis using LDA was applied to detect the latent thematic structure of the constructed responses. The five topics detected from the responses were identified as different word groups with different interpretations. For example, Topic 1 and Topic 5 are both *Integrative Borrowing*, and examinees used words from these two topics to support their argument. Topic 2 is *Everyday Language*, and examinees used everyday language in their responses may indicate a lack of necessary details and evidence relative to what was requested in the prompt. Topic 3 and Topic 4 are interpreted as *Simply Borrowing of Words*, and examinees simply borrowed vocabulary from the passage or stem rather than integrating them in a way that supported their

arguments. Additionally, a correlation analysis between each topic and human rater scores was done to help further interpret the different topics with respect to scores on the CR items. The topic probability distributions were discussed in the context of ability levels and suggested that topics having a modest correlation with the raw scores may be useful as partial indicators of writing proficiency for different levels of ability. This hybrid assessment framework to the mixed-format score patterns and textual data shields light on the possibility of combining the traditional item response analysis and the start-of-art topic model.

4.5 Summary and Discussion

This chapter discussed the use of topic modeling and its extensions from both unsupervised and supervised models for the analysis of textual data in examinees' CR responses. Specifically, the unsupervised topic model and supervised topic model were applied to understand examinees' writing structure and to predict examinees' writing scores.

In the first study, results for two types of topic models, LDA and sLDA, were applied and compared for two social studies assessments, a U.S. History assessment and an Economics assessment, to detect different latent thematic patterns from examinees' responses. The topic structures detected by LDA and sLDA showed similar topic structures for each CR item. For the U.S. History item, the correlations between the human rater score and the topic proportions from the LDA model indicated that the use of Topic 4 was modestly related to a higher score and the use of Topics 1 or 2 was related to a lower score. The regression coefficients from the sLDA suggested a similar outcome as the use of words from Topic 4 was associated with a higher predicted score than the use of words from Topics 1 or 2. For the Economics item, correlations between topic proportions from the LDA and human rater score suggested use of words

from Topic 3 was modestly related to higher scores and use of Topic 1 was modestly related to lower scores. Similarly, the use of words from Topic 3 was modestly associated with a higher predicted score and the use of words from Topic 1 was modestly associated with a score of effectively zero. In addition, it was shown that the latent thematic structure of the text of responses could extend what could be learned from the analysis of CR answers beyond the scores alone. The topic structure can be used to provide information on how examinees come up with their answers, that is, it can provide information about examinees' reasoning and thinking as reflected in their textual answes. What appears to be evident from the topic model results is that differences in examinees' reasoning might be reflected in differences in their use of topics in the model. It would be useful to examine this conjecture in future research.

The second study proposed a framework for an automated scoring engine using the sLDA and generalized logit model as the foundation. Constructed responses may be scored by human raters or through an automated scoring engine. Topic modeling provides a tool for mining textual data in an effort to detect the latent semantic structures. Therefore, the use of sLDA may be useful to both detect the latent semantic structure and to predict the scores. The utility of different sLDA models for detecting the latent topic structure and scoring on an item of English and language arts were compared. Specifically, four different n-gram tokens, namely, unigram, bigram, trigram, and mix-gram were employed and compared. Classification accuracy (CA) of the different tokens was used as a criterion to select the best number of topics for each sLDA model. The results from the empirical data showed that the model with unigram tokens performed best in that it had the highest human-machine score agreement. The models were tested further using 5-fold cross-validation. Each model model incorporated a covariate, the scaled response length which value is between 0 and 1, to predict examinee's writing score. Among these four models, the unigram, bigram, and mix-gram models yielded similar model classification. The unigram sLDA

model showed the highest CA of a 0.880 $QW - \kappa$ score. The overall CA from the unigram model was 0.702. The model dimensionality problem caused by token complexity would seem to be a useful issue to be explored in future research. This might be done by employing methods such as word embedding. In addition, the sLDA model could be further studied to improve higher accuracy by adding effective features.

The third study proposed a five-step analytic framework to analyze and understand both the mixed-format data and textual responses. Along with the introduction of this framework, we showed how the analytic framework worked with an English Language Arts data set. The results of this study aimed to contribute to the literature on mixed-format assessments in several ways. First, the analytic framework provides a conceptual method for modeling mixed-format data. Specifically, it considered the Rasch measurement model for the unidimensional model as well as a structural equation modeling approach for the multidimensional model. Second, by discussing the ability measures estimated from the multidimensional models and the unidimensional model, we advance our understanding of the implications of the multidimensionality that is sometimes present in mixed-format assessments. Finally, by applying topic modeling to analyze the constructed responses, we show the extent to which the ability measures were related to writing structures.

Chapter 5

Conclusion

The central topic of this dissertation was an exploratory analysis of the kind of process data collected from a mixed-format assessment containing both the MC items and CR items. The analysis focused on ways of extracting information from the data of a mixed-format assessment that could improve estimates of ability. This is important because the process data collected during the assessment may provide information that can help to determine which strategies are used by examinees in their answers. Therefore, one conjecture is that the use of process data may help provide additional information about examinees' latent information related to their strategies and behaviors as reflected in their responses.

The process data generated during examinees' responses to the MC items are referred to as the log-file data, and the process data generated during examinees' responses to the CR items, are referred to as the textual responses. This dissertation began by first reviewing studies that focus on the same or similar issues in mixed format tests. Then it described two methods to deal with the two types of process data, i.e., the log file data and the textual response data. The sequential reservoir model (SRM), which consists of the echo state network, was used to analyze the log-file data and extract features from examinees' sequential

actions in responding to the items on a test. The topic model and its extensions were then used to analyze examinees' textual responses into topics. In the first section of this dissertation, the exploratory analysis of MC item process data was discussed. In the second section, the analysis of the CR item process data was discussed. Within each section, several studies were conducted with an eye to understanding process data and the information that could be extracted from those data. Exploratory approaches to combining the response scores and process data information were also investigated. For example, in the MC item process data section, scores and extracted information were simultaneously used in a linear model to predict examinees' latent ability scores. In the CR item process data, the analytic framework provided a way to both understand examinees' thinking and reasoning and access examinees' latent ability estimates. The results of these were discussed with an eye to providing a better understanding of what we might expect to obtain from examinees' assessment process data, and the extent to which the process data might be able to help understand examinees' latent abilities.

In the description of an exploratory methodology for extracting process data from a log file, the SRM using an optimization algorithm was presented. The original process data in the log file contain the sequence of actions by each examinee on each MC item. Each examinee's process data are likely to differ in length from those of other examinees possibly reflecting use of different response strategies. By applying the SRM, the varying-length sequence data can be successfully transformed into fixed-length vectors that can be stored in a feature matrix. This feature matrix is assumed to contain information that can be used to help interpret the latent information in the response processes associated with the item responses. Three simulation studies were presented to demonstrate the SMR application. Results from Simulation I suggest that the group classification accuracy using the extracted features was higher than that from using the baseline features, where baseline feature are simply those averaged from the input

embedded features. Results from Simulation II used the extracted features to predict each examinee's latent ability score. Results from use of the features extracted using the SRM suggest that a lower error was produced when estimating latent ability than when using the baseline features alone. Results from the third simulation study used both the extracted features and the response scores to predict each examinee's ability. Results suggested a better model fit could be obtained than that from using only the response scores.

Results from the simulation studies in this chapter suggested at least two conclusions. First, SMR appears to be useful for extracting features from examinees' response actions during the assessment. Second, the extracted features appear to be useful for classifying examinees' latent groups, predicting more accurate latent ability estimates, and yielding a better model fit. Based on the results of the simulations, an example using empirical data was presented. Data for this example were taken from a NAEP math assessment. Results from using the SRM on the empirical data were in agreement with some of the conclusions from the simulation studies. In the main, results suggested that a proportion of the process data may contain useful information for predicting whether an examinee efficiently responded to the assessment or not.

Although the simulations and empirical examples provided helpful solutions to understanding the process data, one limitation is that the exploratory study only considered the action in responding itself. It ignored the time points that were associated with each action sequence. Log files typically contain a time point for each response action. Using this timing data, it may be possible to determine the difference in response times between consecutive actions. This could lead to future research on how time could be modeled and the extent to which that information could help to improve the estimate of ability.

The use of the topic model and its use in extracting features of the process data was discussed next. The conjecture for this aspect of the research is that by utilizing topic models, it would be possible to

improve estimates of ability by including information about the latent thematic structure of their written responses. Both unsupervised and supervised topic models were studied to determine their utility for helping detect the use of the latent themes in examinees' responses and for prediction of writing scores. Use of both types of models provided useful information about how response scores and process data might be combined to enhance the interpretations of the estimates of examinees' ability.

The study of CR item process data showed results for the LDA and sLDA for two social studies assessments, a U.S. History assessment and an Economics assessment. The topic structures detected by LDA and sLDA showed similar topic structures for each CR item. Correlations between the topic proportions and scores suggested that both LDA and sLDA could be usefully employed to understand the relationship between examinees' use of latent themes in their responses and their scores on the writing assessment.

Based on results from the first study of CR item process data using sLDA, therefore, the next study described a proposed automated scoring engine using the sLDA and generalized logit model. The use of different sLDA models for detecting the latent topic structure and scoring on an item of English and language arts were compared. Classification accuracy was used as a criterion to select the best number of topics for each sLDA and generalized logit model. A comparison of four different n-gram tokens, the unigram, bigram, trigram, and mix-gram, on empirical data suggested that the sLDA and generalized logit model with unigram tokens performed best in terms of the highest human-machine score agreement. The unigram sLDA and generalized logit model showed the highest classification accuracy of 0.702 and a $QW - \kappa$ score of $.88$.

A third study in this chapter proposed a five-step framework to simultaneously analyze and interpret both the mixed-format data and the textual response data. An empirical dataset from an English

Language Arts assessment was used. Results suggested the use of a constrained bi-factor model for the mixed-format response data and a five-topic model for the CR answers. A correlation analysis between the topic proportions and constrained bi-factor model scaled ability estimates indicated that use of some topics were modestly related to a higher ability score and use of other topics were modestly related to a lower ability score. These results may suggest that use of different topics in the model could be used as additional indicators of writing proficiency in estimating examinees' ability. Future research might be considered focusing on how to use these process data to improve the latent ability estimates.

Reference

Athreya, K. B., Doss, H., & Sethuraman, J. (1996). On the convergence of the markov chain simulation method. *The Annals of Statistics*, *24*(1), 69–100.

Attali, Y. (2004). Exploring the feedback and revision features of criterion. *Journal of Second Language Writing*, *14*, 191–205.

Auer, E. M., Mersy, G., Marin, S., Blaik, J., & Landers, R. N. (2022). Using machine learning to model trace behavioral data from a game-based assessment. *International Journal of Selection and Assessment*, *30*(1), 82–102. https://doi.org/10.1111/ijsa.12363

Baldwin, P., Yaneva, V., Mee, J., Clauser, B. E., & Ha, L. A. (2021). Using natural language processing to predict item response times and improve test construction. *Journal of Educational Measurement*, *58*(1), 4–30. https://doi.org/10.1111/jedm.12264

Beebe, N. L., Maddox, L. A., Liu, L., & Sun, M. (2013). Sceadan: Using concatenated n-gram vectors for improved file and data type classification. *IEEE Transactions on Information Forensics and Security*, *8*(9), 1519–1530.

Behrens, J. T., DiCerbo, K. E., & Foltz, P. W. (2019). Assessment of complex performances in digital environments. *The Annals of the American Academy of Political and Social Science*, *683*(1), 217–232.

Bejar, I. I., Mislevy, R. J., & Zhang, M. (2016). Automated scoring with validity in mind. *The wiley handbook of cognition and assessment: Frameworks, Methodologies, and applications*, 226–246.

Bentler, P. M. (2009). Alpha, dimension-free, and model-based internal consistency reliability. *Psychometrika*, *74*(1), 137–143.

Bianchi, F. M., Scardapane, S., Løkse, S., & Jenssen, R. (2020). Reservoir computing approaches for representation and classification of multivariate time series. *IEEE transactions on neural networks and learning systems*, *32*(5), 2169–2179.

Bickel, G., Nord, M., Price, C., Hamilton, W., & Cook, J. (2000). Guide to measuring household food security.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, *3*, 993–1022.

Bompas, S., Georgeot, B., & Guéry-Odelin, D. (2020). Accuracy of neural networks for the simulation of chaotic dynamics: Precision of training data vs precision of the algorithm. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *30*(11), 113118.

Buenaño-Fernandez, D., González, M., Gil, D., & Luján-Mora, S. (2020). Text mining of open-ended questions in self-assessment of university teachers: An LDA topic modeling approach. *IEEE Access*, *8*, 35318–35330. https://doi.org/10.1109/ACCESS.2020.2974983

Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: Understanding AIC and BIC in model selection. *Sociological methods & research*, *33*(2), 261–304.

Cai, C. (2019). Automatic essay scoring with recurrent neural network. *Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications*, 1–7. https://doi.org/10.1145/3318265.3318296

Cardozo-Gaibisso, L., Kim, S., Buxton, C., & Cohen, A. (2020). Thinking beyond the score: Multidimensional analysis of student performance to inform the next generation of science assessments. *Journal of Research in Science Teaching*, *57*(6), 856–878.

Chalmers, R. P. (2012). Mirt: A multidimensional item response theory package for the r environment. *Journal of statistical Software*, *48*, 1–29.

Chapelle, C. A., & Chung, Y.-R. (2010). The promise of NLP and speech processing technologies in language assessment. *Language Testing*, *27*(3), 301–315.

Chen, Y., Yu, B., Zhang, X., & Yu, Y. (2016). Topic modeling for evaluating students' reflective writing: A case study of pre-service teachers' journals. *Proceedings of the sixth international conference on learning analytics & knowledge*, 1–5.

Choi, H.-J., Kim, S., Cohen, A. S., Templin, J., & Copur-Gencturk, Y. (2021). Integrating a statistical topic model and a diagnostic classification model for analyzing items in a mixed format assessment. *Frontiers in Psychology*, 3997.

Choi, H.-J., Kwak, M., Kim, S., Xiong, J., Cohen, A. S., & Bottge, B. A. (2017). An application of a topic model to two educational assessments. *The Annual Meeting of the Psychometric Society*, 449–459.

Chouikhi, N., Ammar, B., Hussain, A., & Alimi, A. M. (2019). Bi-level multi-objective evolution of a multi-layered echo-state network autoencoder for data representations. *Neurocomputing*, *341*, 195–211.

Chouikhi, N., Ammar, B., Rokbani, N., & Alimi, A. M. (2017). PSO-based analysis of echo state network parameters for time series forecasting. *Applied Soft Computing*, *55*, 211–225.

Clifton Jr, C., Ferreira, F., Henderson, J. M., Inhoff, A. W., Liversedge, S. P., Reichle, E. D., & Schotter, E. R. (2016). Eye movements in reading and information processing: Keith rayner's 40 year legacy. *Journal of Memory and Language*, *86*, 1–19.

Co-operation, O. f. E., & Development. (2012). *Literacy, numeracy and problem solving in technology-rich environments: Framework for the OECD survey of adult skills*. OECD Publishing Paris.

Cui, Y., Chu, M.-W., & Chen, F. (2019). Analyzing student process data in game-based assessments with bayesian knowledge tracing and dynamic bayesian networks. *11*(1), 21.

Cui, Y., Gierl, M., & Guo, Q. (2016). Statistical classification for cognitive diagnostic assessment: An artificial neural network approach. *Educational Psychology*, *36*(6), 1065–1082.

Dewi, C., & Chen, R.-C. (2019). Random forest and support vector machine on features selection for regression analysis. *Int. J. Innov. Comput. Inf. Control*, *15*(6), 2027–2037.

Dey, A., Jenamani, M., & Thakkar, J. J. (2018). Senti-n-gram: An n-gram lexicon for sentiment analysis. *Expert Systems with Applications*, *103*, 92–105.

Dunteman, G. H. (1989). *Principal components analysis*. Sage.

Dzikovska, M. O., Nielsen, R., & Brew, C. (2012). Towards effective tutorial feedback for explanation questions: A dataset and baselines. *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 200–210.

Eisner, E. W. (1999). The uses and limits of performance assessment. *Phi Delta Kappan*, *80*(9), 658.

El Aouifi, H., Es-Saady, Y., El Hajji, M., Mimis, M., & Douzi, H. (2021). Toward student classification in educational video courses using knowledge tracing. In M. Fakir, M. Baslam, & R. El Ayachi (Eds.),

Business intelligence (pp. 73–82). Springer International Publishing. https://doi.org/10.1007/978-3-030-76508-8_6

Engelhard Jr, G. (2013). *Invariant measurement: Using rasch models in the social, behavioral, and health sciences*. Routledge.

Ercikan, K., Guo, H., & He, Q. (2020). Use of response process data to inform group comparisons and fairness research. *Educational Assessment*, *25*(3), 179–197.

Ercikan, K., Julian, M. W., Burket, G. R., Weber, M. M., & Link, V. (1998). Calibration and scoring of tests with multiple-choice and constructed-response item types. *Journal of Educational Measurement*, *35*(2), 137–154.

Ercikan, K., & Pellegrino, J. W. (2017). *Validation of score meaning for the next generation of assessments: The use of response processes*. Taylor & Francis.

Fabrigar, L. R., & Wegener, D. T. (2011). *Exploratory factor analysis*. Oxford University Press.

Fleiss, J. L., & Cohen, J. (1973). The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, *33*(3), 613–619.

Flor, M., & Hao, J. (2021). Text mining and automated scoring. In A. A. von Davier, R. J. Mislevy, & J. Hao (Eds.), *Computational psychometrics: New methodologies for a new generation of digital learning and assessment: With examples in r and python* (pp. 245–262). Springer International Publishing. https://doi.org/10.1007/978-3-030-74394-9_14

Franklin, S. B., Gibson, D. J., Robertson, P. A., Pohlmann, J. T., & Fralish, J. S. (1995). Parallel analysis: A method for determining significant principal components. *Journal of Vegetation Science*, *6*(1), 99–106. https://doi.org/10.2307/3236261

Gallicchio, C. (2018). Short-term memory of deep rnn. *arXiv preprint arXiv:1802.00748*.

Garro, B. A., & Vázquez, R. A. (2015). Designing artificial neural networks using particle swarm optimization algorithms. *Computational intelligence and neuroscience*, *2015*.

Gauthier, D. J., Bollt, E., Griffith, A., & Barbosa, W. A. (2021). Next generation reservoir computing. *Nature communications*, *12*(1), 1–8.

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

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National academy of Sciences*, *101*, 5228–5235.

Grimmer, J. (2010). A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, *18*(1), 1–35.

Gudise, V. G., & Venayagamoorthy, G. K. (2003). Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*, 110–117.

Guo, H., Rios, J. A., Haberman, S., Liu, O. L., Wang, J., & Paek, I. (2016). A new procedure for detection of students' rapid guessing responses using response time. *Applied Measurement in Education*, *29*(3), 173–183.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, *46*(1), 389–422.

Han, Y., & Wilson, M. (2022). Analyzing student response processes to evaluate success on a technology-based problem-solving task. *Applied Measurement in Education*, 1–13.

Hendrickson, A., Patterson, B., & Ewing, M. (2010). Developing form assembly specifications for exams with multiple choice and constructed response items: Balancing reliability and validity concerns. *College Board*.

Hogan, T. P., & Murphy, G. (2007). Recommendations for preparing and scoring constructed-response items: What the experts say. *Applied Measurement in Education*, *20*(4), 427–441.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika*, *30*(2), 179–185.

Huang, M.-L., Hung, Y.-H., Lee, W. M., Li, R.-K., & Jiang, B.-R. (2014). SVM-RFE based feature selection and taguchi parameters optimization for multiclass SVM classifier. *The Scientific World Journal*, *2014*.

Jaeger, H. (2001). The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, *148*(34), 13.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Jirak, D., Tietz, S., Ali, H., & Wermter, S. (2020). Echo state networks and long short-term memory for continuous gesture recognition: A comparative study. *Cognitive Computation*, 1–13.

Juang, C.-F. (2004). A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *34*(2), 997–1006.

Kamath, U., Liu, J., & Whitaker, J. (2019). *Deep learning for NLP and speech recognition* (Vol. 84). Springer.

Kaplan, D. (2008). An overview of markov chain methods for the study of stage-sequential developmental processes. *Developmental psychology*, *44*(2), 457.

Kaplan, R. M. (1992). Scoring natural language free-response items-a practical approach. *proceedings of the 33rd Annual Conference of the Military Testing Association*, 514–518.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, *4*, 1942–1948.

Kim, S., Kwak, M., Cardozo-Gaibisso, L., Buxton, C., & Cohen, A. S. (2017). Statistical and qualitative analyses of students' answers to a constructed response test of science inquiry knowledge. *Journal of Writing Analytics*, *1*.

Kim, Y. (2009). *Combining constructed response items and multiple choice items using a hierarchical rater model*. Columbia University.

Knierim, J. J. (2014). Information processing in neural networks. *From molecules to networks* (pp. 563–589). Elsevier.

Kozma, R. (2009). Transforming education: Assessing and teaching 21st century skills. *The transition to computer-based assessment*, *13*.

Kuechler, W. L., & Simkin, M. G. (2010). Why is performance on multiple-choice tests and constructed-response tests not more closely related? theory and an empirical test. *Decision Sciences Journal of Innovative Education*, *8*(1), 55–73.

Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. *Twenty-ninth AAAI conference on artificial intelligence*.

Landauer, T. K., Laham, D., Rehder, B., & Schreiner, M. E. (1997). How well can passage meaning be derived without using word order? a comparison of latent semantic analysis and humans. *Proceedings of the 19th annual meeting of the Cognitive Science Society*, 412–417.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *biometrics*, 159–174.

Leighton, J. (2017). Collecting and analyzing verbal response process data in the service of interpretive and validity arguments. *Validation of score meaning using examinee response processes for the next generation of assessments*, 25–37.

Li, C. H., & Park, S. C. (2009). An efficient document classification model using an improved back propagation neural network and singular value decomposition. *Expert Systems with Applications*, *36*(2), 3208–3215.

Li, F.-J., & Li, Y. (2017). Effects of the minimal singular value on the performance of echo state networks. *2017 36th Chinese Control Conference (CCC)*, 3905–3909.

Li, Y., & Li, F. (2019). PSO-based growing echo state network. *Applied Soft Computing*, *85*, 105774.

Liang, G., On, B.-W., Jeong, D., Kim, H.-C., & Choi, G. S. (2018). Automated essay scoring: A siamese bidirectional LSTM neural network architecture. *Symmetry*, *10*(12), 682.

Liu, T., & Israel, M. (2022). Uncovering students' problem-solving processes in game-based learning environments. *Computers & Education*, *182*, 104462. https://doi.org/10.1016/j.compedu.2022.104462

Livingston, S. A. (2009). Constructed-response test questions: Why we use them; how we score them. r&d connections. number 11. *Educational Testing Service*.

Longman, R. S., Cota, A. A., Holden, R. R., & Fekken, G. C. (1989). A regression equation for the parallel analysis criterion in principal components analysis: Mean and 95th percentile eigenvalues. *Multivariate behavioral research*, *24*(1), 59–69.

Luong, M.-T., Le, Q. V., Sutskever, I., Vinyals, O., & Kaiser, L. (2015). Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Ma, Q., Shen, L., Chen, W., Wang, J., Wei, J., & Yu, Z. (2016). Functional echo state network for time series classification. *Information Sciences*, *373*, 1–20.

Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, *14*(11), 2531–2560.

Masters, G. N. (1982). A rasch model for partial credit scoring. *Psychometrika*, *47*(2), 149–174.

Mcauliffe, J., & Blei, D. (2007). Supervised topic models. *Advances in neural information processing systems*, *20*.

Md Desa, Z. N. D. (2012). *Bi-factor multidimensional item response theory modeling for subscores estimation, reliability, and classification* (Doctoral dissertation). University of Kansas.

Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. *Design and Applications*, *5*, 64–67.

Merchant, K., & Pande, Y. (2018). Nlp based latent semantic analysis for legal text summarization. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1803–1807.

Meyer, J. P. (2010). A mixture rasch model with item response time components. *Applied Psychological Measurement*, *34*(7), 521–538.

Mislevy, R. J. (2019). Advances in measurement and cognition. *The Annals of the American Academy of Political and Social Science*, *683*(1), 164–182.

Mislevy, R. J., Corrigan, S., Oranje, A., DiCerbo, K., Bauer, M. I., von Davier, A., & John, M. (2016). Psychometrics and game-based assessment. *Technology and testing: Improving educational and psychological measurement*, 23–48.

Montavon, G., Samek, W., & Müller, K.-R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, *73*, 1–15.

Myerson, J., Green, L., & Warusawitharana, M. (2001). Area under the curve as a measure of discounting. *Journal of the experimental analysis of behavior*, *76*(2), 235–243.

Nam, J., Kim, J., Loza Mencía, E., Gurevych, I., & Fürnkranz, J. (2014). Large-scale multi-label text classification—revisiting neural networks. *Joint european conference on machine learning and knowledge discovery in databases*, 437–452.

Nebel, S., & Ninaus, M. (2019). New perspectives on game-based assessment with process data and physiological signals. In D. Ifenthaler & Y. J. Kim (Eds.), *Game-based assessment revisited* (pp. 141–161). Springer International Publishing. https://doi.org/10.1007/978-3-030-15569-8_8

Nichols, P. (2005). *Evidence for the interpretation and use of scores from an automated essay scorer*. PEM Research Report 05.

Nickerson, R. S. (1989). New directions in educational assessment. *Educational Researcher*, *18*(9), 3–7.

Noble, W. S. (2006). What is a support vector machine? *Nature biotechnology*, *24*(12), 1565–1567.

Onan, A., Korukoglu, S., & Bulut, H. (2016). LDA-based topic modelling in text sentiment classification: An empirical analysis. *Int. J. Comput. Linguistics Appl.*, *7*(1), 101–119.

Oranje, A., Gorin, J., Jia, Y., & Kerr, D. (2017). Collecting, analyzing, and interpreting response time, eye-tracking, and log data. *Validation of score meaning for the next generation of assessments*, 39–51.

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *International conference on machine learning*, 1310–1318.

Qasem, S. N., & Mohammadzadeh, A. (2021). A deep learned type-2 fuzzy neural network: Singular value decomposition approach. *Applied Soft Computing*, *105*, 107244.

Rakotomamonjy, A. (2003). Variable selection using SVM-based criteria. *Journal of machine learning research*, *3*, 1357–1370.

Rakotomamonjy, A. (2004). Optimizing area under roc curve with SVMs. *ROCAI*, 71–80.

Ramesh, A., Goldwasser, D., Huang, B., Daumé III, H., & Getoor, L. (2014). Understanding MOOC discussion forums using seeded LDA. *Proceedings of the ninth workshop on innovative use of NLP for building educational applications*, 28–33.

Rasch, G. (1966). An individualistic approach to item analysis. *Readings in mathematical social science*, 89–108.

Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests.* ERIC.

Roberts, M. E., Stewart, B. M., Tingley, D., & Airoldi, E. M. (2013). The structural topic model and applied social science. *Advances in neural information processing systems workshop on topic models: computation, application, and evaluation*, *4*, 1–20.

Rojas, R. (2013). *Neural networks: A systematic introduction*. Springer Science & Business Media.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Sanz, H., Valim, C., Vegas, E., Oller, J. M., & Reverter, F. (2018). SVM-RFE: Selection and visualization of the most relevant features through non-linear kernels. *BMC bioinformatics*, *19*(1), 1–18.

Schaetti, N. (2019). Behaviors of reservoir computing models for textual documents classification. *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–7.

Sebrechts, M. M., Bennett, R. E., & Rock, D. A. (1991). Agreement between expert-system and human raters' scores on complex constructed-response quantitative items. *ETS Research Report Series*, *1991*(1), 856–862.

Shermis, M. D. (2014). State-of-the-art automated essay scoring: Competition, results, and future directions from a united states demonstration. *Assessing Writing*, *20*, 53–76.

Shevlin, M., & Miles, J. N. V. (1998). Effects of sample size, model specification and factor loadings on the GFI in confirmatory factor analysis. *Personality and Individual Differences*, *25*(1), 85–90. https://doi.org/10.1016/S0191-8869(98)00055-5

Southavilay, V., Yacef, K., Reimann, P., & Calvo, R. A. (2013). Analysis of collaborative writing processes using revision maps and probabilistic topic models. *Proceedings of the third international conference on learning analytics and knowledge*, 38–47.

Strauss, T., Wustlich, W., & Labahn, R. (2012). Design strategies for weight matrices of echo state networks. *Neural computation*, *24*(12), 3246–3276.

Stucky, B. D., & Edelen, M. O. (2014). Using hierarchical IRT models to create unidimensional measures from multidimensional data. *Handbook of item response theory modeling*, 201–224.

Sun, L., Jin, B., Yang, H., Tong, J., Liu, C., & Xiong, H. (2019). Unsupervised EEG feature extraction based on echo state network. *Information Sciences*, *475*, 1–17.

Swygert, K. A., McLeod, L. D., & Thissen, D. (2001). Factor analysis for items or testlets scored in more than two categories. *Test scoring* (pp. 229–262). Routledge.

Takens, F. (1981). Detecting strange attractors in turbulence. *Dynamical systems and turbulence, warwick 1980* (pp. 366–381). Springer.

Tanaka, V. T., Engelhard, G., & Rabbitt, M. P. (2020). Using a bifactor model to measure food insecurity in households with children. *Journal of Family and Economic Issues*, *41*(3), 492–504.

Tang, S., Peterson, J. C., & Pardos, Z. A. (2016). Deep neural networks and how they apply to sequential education data. *Proceedings of the third (2016) acm conference on learning@ scale*, 321–324.

Tang, X., Wang, Z., He, Q., Liu, J., & Ying, Z. (2020). Latent feature extraction for process data via multidimensional scaling. *Psychometrika*, *85*(2), 378–397.

Tang, X., Wang, Z., Liu, J., & Ying, Z. (2021). An exploratory analysis of the latent structure of process data via action sequence autoencoders. *British Journal of Mathematical and Statistical Psychology*, *74*(1), 1–33.

Tatsuoka, K. K. (1991). Item construction and psychometric models appropriate for constructed responses. *ETS Research Report Series*, *1991*(2), i–38.

Tong, Z., & Tanaka, G. (2018). Reservoir computing with untrained convolutional neural networks for image recognition. *2018 24th International Conference on Pattern Recognition (ICPR)*, 1289–1294. https://doi.org/10.1109/ICPR.2018.8545471

Tripathy, A., Agrawal, A., & Rath, S. K. (2016). Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications*, *57*, 117–126.

Uto, M., & Ueno, M. (2018). Empirical comparison of item response theory models with rater's parameters. *Heliyon*, *4*(5), e00622.

van der Linden, W. J. (2008). Using response times for item selection in adaptive testing. *Journal of Educational and Behavioral Statistics*, *33*(1), 5–20.

van der Linden, W. J. (2011). Test design and speededness. *Journal of Educational Measurement*, *48*(1), 44–60.

Verstraeten, D., Schrauwen, B., d'Haene, M., & Stroobandt, D. (2007). An experimental unification of reservoir computing methods. *Neural networks*, *20*(3), 391–403.

Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., & Koumoutsakos, P. (2020). Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, *126*, 191–217.

von Davier, A. A. (2017). Computational psychometrics in support of collaborative educational assessments: Computational psychometrics. *Journal of Educational Measurement*, *54*(1), 3–11. https://doi.org/10.1111/jedm.12129

Wall, M. E., Rechtsteiner, A., & Rocha, L. M. (2003). Singular value decomposition and principal component analysis. *A practical approach to microarray data analysis* (pp. 91–109). Springer.

Wallach, H. M., Murray, I., Salakhutdinov, R., & Mimno, D. (2009). Evaluation methods for topic models. *Proceedings of the 26th annual international conference on machine learning*, 1105–1112.

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Computing*, *22*(2), 387–408.

Wang, H., Wang, C., Lv, B., & Pan, X. (2015). Improved variable importance measure of random forest via combining of proximity measure and support vector machine for stable feature selection. *J Inform Comput Sci*, *12*(8), 3241–52.

Wang, S., Xiao, H., & Cohen, A. (2021). Adaptive weight estimation of latent ability: Application to computerized adaptive testing with response revision. *Journal of Educational and Behavioral Statistics*, *46*(5), 560–591.

Wang, T., & Hanson, B. A. (2005). Development and calibration of an item response model that incorporates response time. *Applied Psychological Measurement*, *29*(5), 323–339.

Wei, P., Lu, Z., & Song, J. (2015). Variable importance analysis: A comprehensive review. *Reliability Engineering & System Safety*, *142*, 399–432.

Weingart, L. R., Prietula, M. J., Hyder, E. B., & Genovese, C. R. (1999). Knowledge and the sequential processes of negotiation: A markov chain analysis of response-in-kind. *Journal of Experimental Social Psychology*, *35*(4), 366–393.

Williamson, D. M., Xi, X., & Breyer, F. J. (2012). A framework for evaluation and use of automated scoring. *Educational measurement: issues and practice*, *31*(1), 2–13.

Wilson, M., Gochyyev, P., & Scalise, K. (2017). Modeling data from collaborative assessments: Learning in digital interactive social networks. *Journal of Educational Measurement*, *54*(1), 85–102.

Wise, S. L., & DeMars, C. E. (2006). An application of item response time: The effort-moderated IRT model. *Journal of Educational Measurement*, *43*(1), 19–38.

Wise, S. L., & Kong, X. (2005). Response time effort: A new measure of examinee motivation in computer-based tests. *Applied Measurement in Education*, *18*(2), 163–183.

Wright, B. D., & Masters, G. N. (1982). *Rating scale analysis*. MESA press.

Wyffels, F., & Schrauwen, B. (2010). A comparative study of reservoir computing strategies for monthly time series prediction. *Neurocomputing*, *73*(10), 1958–1964. https://doi.org/10.1016/j.neucom.2010.01.016

Xing, W., Lee, H.-S., & Shibani, A. (2020). Identifying patterns in students' scientific argumentation: Content analysis through text mining using latent dirichlet allocation. *Educational Technology Research and Development*, *68*(5), 2185–2214. https://doi.org/10.1007/s11423-020-09761-w

Xiong, J., Choi, H.-J., Kim, S., Kwak, M., & Cohen, A. S. (2019). Topic modeling of constructed-response answers on social study assessments. *The Annual Meeting of the Psychometric Society*, 263–274.

Xu, H., Fang, G., & Ying, Z. (2020). A latent topic model with markov transition for process data. *British Journal of Mathematical and Statistical Psychology*, *73*(3), 474–505.

Yang, T.-I., Torget, A., & Mihalcea, R. (2011). Topic modeling on historical newspapers. *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, 96–104.

Yuan, J., Xiao, Y., & Liu, H. (2019). Assessment of collaborative problem solving based on process stream data: A new paradigm for extracting indicators and modeling dyad data. *Frontiers in psychology*, *10*, 369.

Zhan, P., Jiao, H., & Liao, D. (2018). Cognitive diagnosis modelling incorporating item response times. *British Journal of Mathematical and Statistical Psychology*, *71*(2), 262–286.

Appendix A

Code

A.1 R code

A.1.1 Sequence log data simulation studies

```
#generate two groups' classification data

#library

library(markovchain)

library(tsne)

library(corrplot)

library(ggplot2)

library(caret)

library(factoextra)

library(glmnet)
```

```
rm(list=ls())

all=10

person=3000

L=2000

ACTIONS=c(letters,LETTERS)

P1=matrix(runif((all-1)^2,-15,15),nrow = all-1, byrow = TRUE)

U=P1

for (i in 1:dim(P1)[1]) {

  U[i,]=exp(P1[i,])/sum(exp(P1[i,]))

}

U_1=cbind(rep(0,all-1),U)

U1=rbind(U_1,c(rep(0,all-1),1))

statesNames <- ACTIONS[1:all]

rownames(U1)<-statesNames

colnames(U1)<-statesNames


#simulate the sequences

out<-NULL

total<-person/2

for (i in 1:total) {

  out[[i]] <- markovchainSequence(n = L, markovchain =
```

```
                                    new("markovchain", states = statesNames,
                                     transitionMatrix = U1),
                                    t0 = "a", include.t0 = T)
}
outs=out
#sequence 1
for (i in 1:length(outs)) {
  a=NULL
  b=NULL
  for (j in 1:length(outs[[i]])) {
    if (outs[[i]][j]==statesNames[all]){
      a=c(a,j)
    }
  }
  b=a[-1]
  outs[[i]]=outs[[i]][-b]
}


P2=matrix(runif((all-1)^2,-15,15),nrow = all-1, byrow = TRUE)
U_2=P2
for (i in 1:dim(P2)[1]) {
  U_2[i,]=exp(P2[i,])/sum(exp(P2[i,]))
```

```
}
U_3=cbind(rep(0,all-1),U_2)
U2=rbind(U_3,c(rep(0,all-1),1))
#generate matrix
statesNames <- ACTIONS[1:all]
rownames(U2)<-statesNames
colnames(U2)<-statesNames
out2<-NULL
total2<-person/2
for (i in 1:total2) {
  out2[[i]] <- markovchainSequence(n = L, markovchain =
                                                  new("markovchain", states = statesNames,
                                                   transitionMatrix = U2),
                                   t0 = "a", include.t0 = T)
}
outs2=out2
for (i in 1:length(outs2)) {
  a=NULL
  b=NULL
  for (j in 1:length(outs2[[i]])) {
    if (outs2[[i]][j]==statesNames[all]){
      a=c(a,j)
```

```
    }
  }
  b=a[-1]
  outs2[[i]]=outs2[[i]][-b]
}
#data frame
data1=outs
for (i in 1:length(outs)) {
  data1[[i]]=paste(outs[[i]],collapse =",")
}
data1_2<-data.frame(action=matrix(unlist(data1), ncol=1, byrow=TRUE),label="0")
data2=outs2
for (i in 1:length(outs2)) {
  data2[[i]]=paste(outs2[[i]],collapse =",")
}
data2_2<-data.frame(action=matrix(unlist(data2), ncol=1, byrow=TRUE),label="1")
#combine and shuffle dataframes
df <- rbind(data1_2,data2_2)
rows <- sample(nrow(df))
df2 <- df[rows, ]
df2$id=1:person
write.csv(df2,"JX\\simulation1.csv",row.names = F)
```

```
#############################################################################
#generate sequence with latent information merged for each chain
rm(list=ls())
all=10
person=3000
L=100
ACTIONS=c(letters,LETTERS)
actionNames <- ACTIONS[1:all]
theta=rnorm(person,0,1)


##generate transition matrix P and U
P=matrix(runif((all-1)^2,-15,15),nrow = all-1, byrow = TRUE)
U1=NULL
for (j in 1:person) {
  U=matrix(rep(0,(all-1)^2),nrow = all-1, byrow = TRUE)
  for (i in 1:dim(U)[1]) {
    U[i,]=exp(theta[j]*P[i,])/sum(exp(theta[j]*P[i,]))
  }
  U_1=cbind(rep(0,all-1),U)
  U1[[j]]=rbind(U_1,c(rep(0,all-1),1))
  rownames(U1[[j]])<-actionNames
```

```
  colnames(U1[[j]])<-actionNames
}
#simulate the sequences
out<-NULL
for (i in 1:person) {
  out[[i]] <- markovchainSequence(n = L, markovchain =
                                  new("markovchain", states = actionNames,
                                    transitionMatrix = U1[[i]]),
                                  t0 = "a", include.t0 = T)
}
outs=out
for (i in 1:length(outs)) {
  a=NULL
  b=NULL
  for (j in 1:length(outs[[i]])) {
    if (outs[[i]][j]==actionNames[all]){
      a=c(a,j)
    }
  }
  b=a[-1]
  outs[[i]]=outs[[i]][-b]
}
```

```
#data frame
data1=outs
for (i in 1:length(outs)) {
  data1[[i]]=paste(outs[[i]],collapse =",")
}
data1_2<-data.frame(action=matrix(unlist(data1), ncol=1, byrow=TRUE),theta=theta)

#combine and shuffle dataframes
data1_2$id=1:person
write.csv(data1_2,"JX\\simulation2.csv",row.names = F)

#############################################################################
#generate responses together
z=55 #item numbers
b_mc=rnorm(z,0,1)
#generating data pars framework
g_p=shape_df(par.dc=list(a=rep(1,z),b=b_mc,g=NULL),
             item.id=c(paste(rep("item", z),
                             seq(1,z,1), sep = " ")),
             cats=rep(2,z),
             model=rep("1PLM",z))
```

```
sularesponse<-data.frame(response=simdat(x=g_p, theta=theta, D=1))

#read into extracted features

feature=read.csv("extracted.csv",header=T)

x.data=as.matrix(cbind(sularesponse,feature))


#ridge regression

lambdas <- 10^seq(2, -3, by = -.1)

ridge_reg = glmnet(x.data, theta, nlambda = 25,

                alpha = 0.5, family = 'gaussian', lambda = lambdas)


cv_ridge <- cv.glmnet(x.data, theta, alpha = 0, lambda = lambdas)

optimal_lambda <- cv_ridge$lambda.min

predictions <- predict(ridge_reg, s = optimal_lambda, newx = x.data)

pile=as.matrix(coef(ridge_reg))


#linear model

baseline=sularesponse

baseline$y=theta

lm1=lm(y~.,data=baseline)

summary(lm1)

predictions2 <- predict(lm1,newx = sularesponse)
```

```
eval_results(theta, predictions, x.data)

eval_results(theta, predictions2, sularesponse)
```

A.1.2 Plot

```
library(ggplot2)

#PCA on the raw features

theta_pca<- prcomp(feature)

#Graph of individuals

fviz_pca_ind(theta_pca, geom = c("point"),

             #col.ind = "contrib",

             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")

)

#personal PCAs

res.ind <- get_pca_ind(theta_pca)

#plot of PCA by group

theta_plot <- data.frame(x = res.ind$coord[,1], y = res.ind$coord[,2],

                         label =df2$label)

ggplot(theta_plot) +xlab("PCA1") + ylab("PCA2")+

  geom_point(aes(x=x, y=y,color=label),size=2)+ theme_bw()+
```

```
  scale_colour_discrete("Levels")
```

A.1.3 LDA

```
#LDA

rm(list=ls())

library(tm)

library(slam)

library(lda)

library(psych)

library(car)


data1 = read.csv(file= 'JX\\SS_Economics_Cumulative_Assesslet_noID.csv',

                 header=T, sep=",", fill=T)

data2 = data1[,c(1,6,8,109,110,111)]

data3 = data2


colnames(data3)[4] <- "ER"

colnames(data3)[5] <- "ERScore"

colnames(data3)[6] <- "ERfeedback"


data4=data3
```

```
data4$text <- paste(data4$ER)

ScoreofER<- data4$ERScore

data4$Score<- paste(ScoreofER)


data5=data4[,c(1,3,7,8)]

data6 = data5[as.character(data5$text)!=" ",] #delete blank records

na.list = substr(data6$text,1,2) == "NA" #delete missing data

data7 = data6[na.list==F,]

text1<-data7$text

id<-data7$stuID

score<-data7$Score

length(text1)

length(score)

length(id)


text1 <- gsub("^[[:space:]]+", "", text1)

text2 <- gsub("[[:space:]]+$", "", text1)

text3 =sub("\\.", " ", text2) #remove periods

text4 = gsub("\\'s", "", text3) #remove "'s"

text5 = gsub("[[:punct:]]", " ", text4) #remove punctuation characters

text6 = gsub("[[:digit:]]", " ", text5) #remove digits

text7 = tolower(text6) #to lower case
```

```
head(text7)

#change sample size

max<-length(text7)

sample_size<-max

start_size<-max-sample_size+1


text7<-text7[start_size:max]

length(text7)




doc.list <- strsplit(text7, "[[:space:]]+")

doc.unlist<-unlist(doc.list)

#delete stop words and words that apprear less than 5 times

term.table = table(doc.unlist)

del  = term.table < 5

new.term.table <- term.table[!del]

vocab <- names(new.term.table)

length(vocab)

#write.table(vocab, file=JX/3narra.txt")

#get document length

d.length = unlist(lapply(doc.list,length))
```

```
new.doc.list=list()
s=0
for(l in 1:length(d.length)){
  e = s + d.length[l]
  new.doc.list[[l]] = doc.unlist[(s+1):e]
  s = e
}

# now put the documents into the format required by the lda package:
get.terms <- function(x) {
  index <- match(x, vocab)
  index <- index[!is.na(index)]
  rbind(as.integer(index - 1), as.integer(rep(1, length(index))))
}
documents <- lapply(new.doc.list, get.terms)
documents[1]

# Compute some statistics related to the data set before preprocessing:
D <- length(documents)  # number of documents
W <- length(vocab)  # number of terms in the vocab
# number of tokens per document
```

```
doc.length <- sapply(documents, function(x) sum(x[2, ]))
N <- sum(doc.length)  # total number of tokens in the data
 # frequencies of terms in the corpus
term.frequency <- as.integer(new.term.table)
N/D


#load stemming list
source("JX/economics_Stemming_lda.R")
d.length = unlist(lapply(doc.list,length))


new.doc.list=list()
s=0
for(l in 1:length(d.length)){
  e = s + d.length[l]
  new.doc.list[[l]] = doc.unlist[(s+1):e]
  s = e
}



# now put the documents into the format required by the lda package:
new.doc.unlist<-unlist(new.doc.list)
term.table = table(new.doc.unlist)
```

```
del  = term.table < 5
new.term.table <- term.table[!del]
vocab <- names(new.term.table)


# now put the documents into the format required by the lda package:
get.terms <- function(x) {
  index <- match(x, vocab)
  index <- index[!is.na(index)]
  rbind(as.integer(index - 1), as.integer(rep(1, length(index))))
}
documents.new <- lapply(new.doc.list, get.terms)


#delete documents that has length less than 10
del.doc = which(sapply(documents.new,length)<10)
id_doc_new<-id[-del.doc]
documents.new = documents.new[-del.doc]
score_new<-score[-del.doc]


length(id_doc_new)
length(documents.new)
length(score_new)
```

```
assign<-function(x){vocab[x[1,]+1]}

text7<-lapply(documents.new, assign)

text7<-lapply(text7, toString)

text7 <- gsub("^[[:space:]]+", "", text7)

text7 <- gsub("[[:space:]]+$", "", text7)

text7 =sub("\\.", " ", text7) #remove periods

text7 = gsub("\\'s", "", text7) #remove "'s"

text7 = gsub("[[:punct:]]", " ", text7) #remove punctuation characters

text7 = gsub("[[:digit:]]", " ", text7) #remove digits

text7 <- gsub("^[[:space:]]+", "", text7)

text7 = tolower(text7) #to lower case



tm0_total<-Corpus(VectorSource(text7))


pstP_dtm0_total<-

            DocumentTermMatrix(tm0_total, control=list(stemming=F,

            minWordLength = 1))

#select words that have more than 2 frequency

#two0_total<-findFreqTerms(pstP_dtm0_total,5) #

pstP_two.total<-pstP_dtm0_total
```

```
rownames(pstP_two.total)<-id_doc_new

term_tfidf <-tapply(pstP_two.total$v/row_sums(pstP_two.total)[pstP_two.total$i],

pstP_two.total$j, mean)*log2(nDocs(pstP_two.total)/col_sums(pstP_two.total > 0))

summary(term_tfidf)

quantile(term_tfidf,0.05)

medi<-unlist(summary(term_tfidf)[2])

a<-sort(term_tfidf)

barplot(a)



#stop words

#cv<-as.numeric(term_tfidf[order(term_tfidf)[30]])

q1=c("next", "not","their","this","only","one","much", "can",

"yet","for","and","are","that","what","him","with","but","out"

,"his","who","from","will","they","also","which",

"other", "you","still","our",

"all","how","than", "two","after","many","have",

"both","there","just","now", "every",

     "into","its","when","while","then","about","yes"

     )
```

```
pstP_two2.total<-pstP_two.total[,setdiff(pstP_two.total$dimnames$Terms, q1)]

rownames(pstP_two2.total)<-id_doc_new

pstP_two3.total <- pstP_two2.total[row_sums(pstP_two2.total) > 0,]

pstP_two3.total <- pstP_two3.total[col_sums(pstP_two3.total) > 0,]

W.total<-pstP_two3.total$ncol




id.new.2<-rownames(pstP_two3.total)

#stopword list

stvoca.list<-q1

stvoca.list



####Corpus Statistics after preprocessing###

W.total<-pstP_two3.total$ncol # number of terms in the vocab

D1 <- pstP_two3.total$nrow  # number of documents

N1 <- sum(pstP_two3.total$v)  # total number of tokens in the data

N1/D1 # average length of document


head(text7)
```

```
####################################

############Estimation##############

####################################


library(topicmodels)


# Setting estimation conditions


iter<-10000

keep<-1

burnin<-5000


#set up library


# generate numerous topic models with different numbers of topics

sequ <- seq(2, 10, 1)

alpha<-30/sequ

#alpha<-1/sequ

#delta<-300/W.total

delta<-1.8

SEED<-3423

test <- lapply(sequ, function(k) LDA(pstP_two3.total,
```

```
          k = k, method = "Gibbs",control=list(alpha=alpha,
           delta=delta, seed=SEED,
           burnin=burnin, iter = iter, keep = keep) ))
#train <- lapply(sequ, function(k) LDA(pstP_two3.training,
k = k, method = "Gibbs",
control=list(alpha=alpha, delta=delta, seed=SEED,
burnin=burnin, iter = iter, keep = keep) ))
#test<- lapply(train, function(l) LDA(pstP_two.test3, model=l,
k = sequ, method = "Gibbs",
control=list(alpha=alpha, delta=delta, seed=SEED,
burnin=burnin, iter = iter, keep = keep)))

# extract logliks from each topic
logLiks_many <- lapply(test,
                function(q) q@logLiks[-c(1:(burnin/keep))])

hm_many2 <- sapply(logLiks_many, function(h) harmonic.mean(h))

average<-function(v){mean(unlist(logLiks_many[v]))}
t<-sequ-1
dbar<--2*sapply(t, function(g) average(g))
```

```
num.par<-sequ*D1+sequ*W.total

dhat<--2*hm_many2

Pd<-dbar-(dhat)

aicc.penalty<-2*(num.par*(num.par+1)/(D1-num.par-1))

dic<-dbar+2*Pd

bic<--2*hm_many2+num.par*log(D1)

aic<--2*hm_many2+num.par

aicc<-aic+aicc.penalty

ssa_bic<--2*hm_many2+num.par*log((D1+2)/24)




plot(sequ,dic,type="o",

        main="Model Selection by Deviance Information Criterion",

      col="red",xlab="number of topics", ylab="DIC")



# inspect(likelihood & perplexity)


plot(sequ,dic, type="l")

plot(sequ,bic, type="l")
```

```
plot(sequ,aic, type="l")

plot(sequ,aicc, type="l")

plot(sequ,ssa_bic, type="l")


#Extracting


k<- 3 #number of topics

alpha<-30/k #dirichlet prior

#delta<-200/W.total #dirichlet prior

#alpha<-1/k #dirichlet prior


SEED<-7845652

m <-LDA(pstP_two3.total, k=k,

 method = "Gibbs", control=list(alpha=alpha,

       delta=delta, seed=SEED, burnin=burnin, iter=20000))

pstP_two3.total


n1<-table(m@z)[1]

n2<-table(m@z)[2]

n3<-table(m@z)[3]
```

```
##simple one

Topic_k<-topics(m,3)

Terms_k<-terms(m,30)

term.prob<-posterior(m, pstP_two3.total)$terms


data_post<-m@gamma # topic-document distribution

write.table(data_post, "JX\\mydata_post4.txt", sep="\t")


#term.prob<-posterior(m, pstP_two.total)$terms




#Support of the phi

list.m.t1<-order(term.prob[1,], decreasing=TRUE)[1:30]

list.m.t2<-order(term.prob[2,], decreasing=TRUE)[1:30]

list.m.t3<-order(term.prob[3,], decreasing=TRUE)[1:30]




#Posterior distribution of phi

m.t1<-term.prob[1,list.m.t1]

m.t2<-term.prob[2,list.m.t2]

m.t3<-term.prob[3,list.m.t3]
```

```
data1<-round(as.data.frame(m.t1),3)

data2<-round(as.data.frame(m.t2),3)

data3<-round(as.data.frame(m.t3),3)



#list document by topics


topic_pro<-m@gamma

final<-as.data.frame(cbind(id_doc_new,score_new))

id.label<-as.numeric(id.new.2)

small.rep<-length(id.label)

score_final_vec<-matrix(rep(NA, small.rep*2), nrow=small.rep)

for (i in 1:small.rep){

  score_final<-final[which(final$id_doc_new==id.label[i]),]

  score_final_vec[i,]<-matrix(as.numeric(score_final),ncol=2)

}


final_0<-cbind(id.new.2,topic_pro, score_final_vec[,2])

head(final_0)


final_0[id.new.2==407]
```

```
#topic 1

final_0[order(final_0[,2], decreasing=T),][1:5]

#topic 2

final_0[order(final_0[,3], decreasing=T),][1:5]

#topic 3

final_0[order(final_0[,4], decreasing=T),][1:5]
```

```
topic_pro<-m@gamma

score_final<-score[as.numeric(id.new.2)]

final_0<-cbind(id.new.2,topic_pro, score_final)

head(final_0)


final_0[id.new.2==2]
```

```
cor.used.topic<-3
logit<-function(x){log(x/(1-x))}
transform.topic.portion<-sapply(as.numeric(final_0[,cor.used.topic+1]),logit)
cor(transform.topic.portion,as.numeric(final_0[,5]),
      use="pairwise.complete.obs")

cor.used.topic<-2
logit<-function(x){log(x/(1-x))}
transform.topic.portion<-sapply(as.numeric(final_0[,cor.used.topic+1]),logit)
cor(transform.topic.portion,as.numeric(final_0[,5]),
      use="pairwise.complete.obs")

cor.used.topic<-1
logit<-function(x){log(x/(1-x))}
transform.topic.portion<-sapply(as.numeric(final_0[,cor.used.topic+1]),logit)
cor(transform.topic.portion,as.numeric(final_0[,5]),
     use="pairwise.complete.obs")
```

A.1.4 Supervised LDA

```
rm(list=ls())
library(tm)
library(slam)
```

```
library(lda)

library(psych)

library(car)

library(stringr)

library(ltm)

library(psych)

library(QuantPsyc)

library(janitor)

library(xlsx)


#input raw file

dat0 = read.csv(file='JX\\SS_U.S._History_Cumulative_Assesslet_noID.csv',

     header=T, sep=",", fill=T)

names(dat0)




#create new file

data00 = dat0[,c(1,109,110)]

dat2.1.0 = data00

colnames(dat2.1.0)[1] <- "id"

colnames(dat2.1.0)[2] <- "ERresponse"

colnames(dat2.1.0)[3] <- "score"
```

```
dim(dat2.1.0)


# choose ER response
dat2.1.1= dat2.1.0[(dat2.1.0$ERresponse)!="",] #delete blank records
dim(dat2.1.1)
na.list = substr(dat2.1.1$ERresponse,1,2) == "NA" #delete missing data
dat2.1.1 = dat2.1.1[na.list==F,]
dat2.1 = dat2.1.1




# dat 2
text1<-dat2.1$ERresponse
id<-dat2.1$id


dim(dat2.1)


#check
length(text1)
length(id)
#processing
text1 <- gsub("^[[:space:]]+", "", text1)
text2 <- gsub("[[:space:]]+$", "", text1)
```

```
text3 =sub("\\.", " ", text2) #remove periods

text4 = gsub("\\'s", "", text3) #remove "'s"

text5 = gsub("[[:punct:]]", " ", text4)

text6 = gsub("[[:digit:]]", " ", text5)

text7 = tolower(text6) #to lower case

head(text7)

head(dat2.1)

#sample size

max<-length(text7)

sample_size<-max

start_size<-max-sample_size+1


dat2.1<-dat2.1[start_size:max,]

text7<-text7[start_size:max]


length(text7)

dim(dat2.1)


#sampling documents

#set.seed(15423)

#text7<-sample(text7,1500, replace=F)
```

```
doc.list <- strsplit(text7, "[[:space:]]+")

doc.unlist<-unlist(doc.list)

#delete stop words and words that apprear less than 5 times

term.table = table(doc.unlist)

del  = term.table < 5

new.term.table <- term.table[!del]

vocab <- names(new.term.table)

length(vocab)

#write.table(vocab, file="JX/3narra.txt")

#get document length

d.length = unlist(lapply(doc.list,length))


new.doc.list=list()

s=0

for(l in 1:length(d.length)){

  e = s + d.length[l]

  new.doc.list[[l]] = doc.unlist[(s+1):e]

  s = e

}


# now put the documents into the format required by the lda package:

get.terms <- function(x) {
```

```
  index <- match(x, vocab)
  index <- index[!is.na(index)]
  rbind(as.integer(index - 1), as.integer(rep(1, length(index))))
}
documents <- lapply(new.doc.list, get.terms)
documents[1]


# Compute some statistics related to the data set before preprocessing:
D <- length(documents)  # number of documents
W <- length(vocab)  # number of terms in the vocab
doc.length <- sapply(documents, function(x) sum(x[2, ]))
N <- sum(doc.length)  # total number of tokens in the data
term.frequency <- as.integer(new.term.table)
N/D




#load stemming list
source("JX/history_Stemming_lda.R")
d.length = unlist(lapply(doc.list,length))
new.doc.list=list()
s=0
```

```
for(l in 1:length(d.length)){
  e = s + d.length[l]
  new.doc.list[[l]] = doc.unlist[(s+1):e]
  s = e
}

documents.new <- lapply(new.doc.list, get.terms)
documents.new[1]

assign<-function(x){vocab[x[1,]+1]}
text7<-lapply(documents.new, assign)
text7<-lapply(text7, toString)
text7 <- gsub("^[[:space:]]+", "", text7)
text7 <- gsub("[[:space:]]+$", "", text7) # remove whitespace
text7 =sub("\\.", " ", text7) #remove periods
text7 = gsub("\\'s", "", text7) #remove "'s"
text7 = gsub("[[:punct:]]", " ", text7) #remove punctuation characters
text7 = gsub("[[:digit:]]", " ", text7) #remove digits
text7 <- gsub("^[[:space:]]+", "", text7)
text7 = tolower(text7) #to lower case
```

```
tm<-Corpus(VectorSource(text7))



#Remove stopwords

tm3<-tm_map(tm, removeNumbers)

source("JX/sub_stopwords.R")


#check;

lapply(tm3[1:1], as.character)

all_dtm<-DocumentTermMatrix(tm3,

        control=list(stemming=F, minWordLength = 2, stopwords=stop_words))

v<-all_dtm$dimnames$Terms



#select words that have more than 5 frequency

thr<-findFreqTerms(all_dtm,5) #

all_thr<-all_dtm[,thr]
```

```
# now put the documents into the format required by the lda package:
get.terms <- function(x) {
  index <- match(x, v)
  index <- index[!is.na(index)]
  rbind(as.integer(index - 1), as.integer(rep(1, length(index))))
}
documents.new <- lapply(new.doc.list, get.terms)




#####Statistics after preprocessing######
W.total<-all_thr$ncol # number of terms in the vocab
D1 <- all_thr$nrow  # number of documents
N1 <- sum(all_thr$v)  # total number of tokens in the data
N1/D1 # average length of document




################################
##########topic model###########
################################
```

```
#determine the number of topics

K <- 4

G <- 5000 # the number of iterations

alpha <- 30/K #0.06/K

eta <- 0.01 #1/W


# Fit SLDA

# ER score

annotations<-dat2.1$score

annotations[is.na(annotations)] <- 0

dim(dat2.1)[1]

length(annotations)

length(documents.new)

variacne<-1/var(annotations,na.rm=T)

params<-c(0,0,0,0)

table(annotations)

annotations<-as.integer(annotations)



#only positive documents can be used

documents.new1=documents.new[lapply(documents.new,length)>0]

min(sapply(documents.new1, length))
```

```
del.doc.list<-id[lapply(documents.new,length)==0]

id.new<-id[lapply(documents.new,length)>0]

annotations.new<-annotations[lapply(documents.new,length)>0]



length(del.doc.list)

length(id)

length(id.new)

length(annotations)

length(annotations.new)

length(documents.new)

length(documents.new1)


#slda

num.e<-500

num.m<-20

t1 <- Sys.time()

fit5_slda <- slda.em(documents= documents.new1,

                           K = K, vocab = v,

                    num.e.iterations=num.e, num.m.iterations=num.m,

                          alpha = alpha,

                    eta = eta, annotations=annotations.new,
```

```
                        params=params, variance=variacne,
                    logistic = FALSE, lambda = 10, regularise = FALSE,
                     method = "sLDA", trace = 0L, MaxNWts=3000)
t2 <- Sys.time()
t2 - t1



# model summary
fit5_slda$model
fit5_slda$coefs
fit5_slda$topic_sums
fit5_slda


topic.prop.docs<-fit5_slda$document_sums
#get indivisual's topic proportion
for (i in 1: length(colSums(fit5_slda$document_sums))){
topic.prop.docs[,i]<-fit5_slda$document_sums[,i]/colSums(fit5_slda$document_sums)[i]
}
#check
sum(topic.prop.docs[,2])
#gain the transpose matrix
smatrix=topic.prop.docs
```

```
tsmatrix=t(smatrix)


# top words
top.words <- top.topic.words(fit5_slda$topics, 30, by.score = F)
top.words


#A function to organize the results
topics = function(fit,K){
  top.words <- top.topic.words(fit5_slda$topics, 30, by.score = F)
  dim(fit5_slda$topics)
  tot = apply(fit5_slda$topics,1,sum)
  p.word = fit5_slda$topics
  results = list()
  for(k in 1:K){
    p.word[k,] = fit5_slda$topics[k,]/tot[k]
    results[[k]] = round(as.data.frame(sort(p.word[k,],
                    decreasing=T)[1:30]),3)
    results[[K+k]] = as.data.frame(sort(fit5_slda$topics[k, ],
                    decreasing = TRUE)[1:30])
  }
  return(results)
}
```

```
#topics

A=topics(fit5_slda,4)[1]

B=topics(fit5_slda,4)[2]

C=topics(fit5_slda,4)[3]

D=topics(fit5_slda,4)[4]


write.csv(A, file = "JX/topic1.csv")

write.csv(B, file = "JX/topic2.csv")

write.csv(C, file = "JX/topic3.csv")

write.csv(D, file = "JX/topic4.csv")


top.topic.words(fit5_slda$topics, 30, by.score = F)







#call each of the score categories

#list document by topics

topic_pro<-tsmatrix

final<-cbind(id.new,topic_pro,annotations.new)

final_0=data.frame(final)
```

```
colnames(final_0)[6]<-"score"

colnames(final_0)[5]<-"topic4pro"

colnames(final_0)[4]<-"topic3pro"

colnames(final_0)[3]<-"topic2pro"

colnames(final_0)[2]<-"topic1pro"

colnames(final_0)[1]<-"id"

head(final_0)

dim(final_0)


#rank

#topic 1

topic1list=head(final_0[order(final_0$topic1,decreasing = T),],n=50)

#topic 2

topic2list=head(final_0[order(final_0$topic2,decreasing = T),],n=50)

#topic 3

topic3list=head(final_0[order(final_0$topic3,decreasing = T),],n=50)

#topic 4

topic4list=head(final_0[order(final_0$topic4,decreasing = T),],n=50)


#number of students in each score for top 50 studenst in each topic

table(topic1list[,6])

table(topic2list[,6])
```

```
table(topic3list[,6])

table(topic4list[,6])



data111 = t(data.frame(c(10,20,30)))

colnames(data111) = c("A","B", "C")

barplot(data111, main="Column Chart", xlab="Grades")



#call 7 score

final_0[final_0$score==7,]

#call 6 score

final_0[final_0$score==6,]

#call 5 score

final_0[final_0$score==5,]

#call 4 score

final_0[final_0$score==4,]

#call 3 score

final_0[final_0$score==3,]

#call 2 score

final_0[final_0$score==2,]

#call 1 score
```

```
final_0[final_0$score==1,]

#call 0 sccore

final_0[final_0$score==0,]




#check

final_0[,2:5]


fin=predict(fit5_slda$model,final_0[,2:5])

fina=final_0

fina$scorepre=fin

pred=fina[,c(1,6,7)]

#residual

pred$dif=pred$score-pred$scorepre

plot(pred$dif, main="Scatterplot between Raw Score and Predicted Score",

     xlab="Student ", ylab="Difference ")


#mean and stand deviation

mean(abs(pred$dif))

sd(abs(pred$dif))
```

```
#SSE

sum(pred$dif^2)

#SST

sum((pred$score-mean(pred$dif))^2)

#SSR

sum((pred$score-mean(pred$dif))^2)-sum(pred$dif^2)

#r^2

(sum((pred$score-mean(pred$dif))^2)-

           sum(pred$dif^2))/sum((pred$score-mean(pred$dif))^2)




predc=pred[c(2,3)]

predc_2=as.numeric(as.vector(as.matrix(predc)))

datapred<-data.frame(id=rep(c(1:409),2),

                  student=rep(c("A","B"),each=409),

                  value=c(predc_2)

)
```

A.2 Matlab

A.2.1 A simple ESN

Thanks to Dr. Mantas Lukoševičius.($https://mantas.info/code/simple_esn/$)

```
clear all

% seed random generator

tic

rand('state', sum(100*clock));


donTr=load('simulation_1.txt');

donTs=load('simulation_2.txt');

don=[donTr(:,2:end);donTs(:,2:end)];

don=don';

donT=[donTr(:,1);donTs(:,1)];


% unit counts (input, hidden, output)

IUC = 100; #change

HUC = 200; #change

OUC = 100; #change

IPP=don;

TPP=IPP;
```

```
probInp  = [  1.00 ];

rngInp   = [  1 ];

probRec  = [  0.01];

rngRec   = [ -0.6 ];

probBack = [ 0.0  ];

rngBack  = [0.0 ];

w_in = zeros(HUC, IUC, length(probInp));

w_rec = zeros(HUC, HUC, length(probRec));

%w_back = zeros(HUC, OUC, length(probBack));


for d=(1:length(probInp))

    w_in(:,:,d) = init_weights(w_in(:,:,d), probInp(d),rngInp(d));

end;


for d=(1:length(probRec))

    w_rec(:,:,d) = init_weights(w_rec(:,:,d), probRec(d),rngRec(d));

end;


% for d=(1:length(probBack))

%     w_back(:,:,d) = init_weights(w_back(:,:,d), probBack(d),rngBack(d));

% end;

SpecRad = max(abs(eig(w_rec(:,:,1))));
```

```
if SpecRad>0,
    w_rec = w_rec ./ SpecRad;
end
SpecRad;


x = zeros(HUC,size(TPP,2));
x(:,1) = rand(1,HUC);
w_out=rand(OUC,HUC);
for t=2:size(TPP,2),
%run without any learning/training in reservoir and readout unit
    x(:,t) = tanh(w_in*IPP(:,t) + w_rec*x(:,t-1));
end
% plot(x);
w_out = TPP(:,10:end)*pinv(x(:,10:end));
% w_in=w_out';
% for t=2:size(TPP,2),
%run without any learning/training in reservoir and readout unit
%
%     x(:,t) = tanh(w_in*IPP(:,t) + w_rec*x(:,t-1));
%
% end
```

```
IP=x(:,1:100)';

TP=donT(1:100,:);

IPT=x(:,101:end)';

TPT=donT(101:end,:);

svmStruct = fitcsvm(IP,TP);

Group = ClssificationSVM(svmStruct,IPT);

SVMStruct1 = fitcsvm(IP,TP);

Group1 = ClssificationSVM(SVMStruct1,IPT);

RD=0;

for(z=1:size(TPT,1));

if (Group1(z,:) == TPT(z,:))

    RD=RD+1;

end

end

  precision= RD/size(TPT,1)

 figure;

plot(x)

title('Hidden neurons activations: new data representation')

xlabel('hidden neurons')

ylabel('Activation')
```