1-1-2020

Discriminative factorization models for student behavioral pattern detection and classification

Mehrdad Mirzaei
University at Albany, State University of New York, mirzaei_mehrdad@yahoo.com

DISCRIMINATIVE FACTORIZATION MODELS FOR STUDENT BEHAVIORAL PATTERN DETECTION AND CLASSIFICATION

by

Mehrdad Mirzaei

A Dissertation

Submitted to the University at Albany, State University of New York

in Partial Fulfillment of

the Requirements for the Degree of

Doctor of Philosophy

College of Engineering and Applied Sciences

Department of Computer Science

Summer 2020

To my Family and Friends

ABSTRACT

The goal of this dissertation is to examine factors such as how a student chooses to engage with the online platform and time spent on individual tasks and draw conclusions to improve the efficiency of the students and efficacy of online learning tools. Student activities and decision-making while functioning in a computer-based learning environment are utilized to guide students with effective patterns in studying. In addition to the sequence of actions, we have considered the time spent on each activity in modeling to have a more accurate representation of students' behavior in studying. Using sequential pattern mining methods, we find students' patterns of behavior in studying. By analyzing the correlation between the most frequent patterns and students' performance, we will derive effective patterns of interaction in an online learning environment. We propose a novel method using non-negative discriminative matrix factorization and pattern structures to find patterns that are common among students (learning traits) and patterns that are specific to a group of students with low or high performance (performance traits). We use these patterns to model the students' behavior and predict their performance.

ACKNOWLEDGMENT

First, I appreciate Dr. Shaghayegh Sahebi, my thesis advisor, for her advice, assistance, and support during my PhD program. I would also like to express my deepest appreciation to my committee members. Dr. Reza Feizy for his invaluable feedback in education field that helped me a lot in my research. Dr. Chelmis Charalampos for his great comments on my work and pointing to the thorough details. And Dr. Ming-Ching Chang for his willingness of serving as my committee member the last moment. I got significant feedback and comments from them to improve my dissertation.

I would also thank my family for their unconditional love that supported me thorough my studies and always encouraged me to finish my PhD.

Finally, I would like to thank my friends here and other countries who stayed connected with me and never let me down.

This dissertation is based on three papers written jointly with Dr. Shaghayegh Sahebi and Dr. Peter Brusilovsky. These papers are:

- Mehrdad Mirzaei, Shaghayegh Sahebi, and Peter Brusilovsky. "Annotated Examples and Parameterized Exercises: Analyzing Students' Behavior Patterns" In International Conference on Artificial Intelligence in Education, pp. 308-319. Springer, Cham, 2019.
- Mehrdad Mirzaei, Shaghayegh Sahebi, and Peter Brusilovsky. "Detecting Trait versus Performance Student Behavioral Patterns Using Discriminative Non-Negative Matrix Factorization" In The Thirty-Third International Flairs Conference. 2020.
- Mehrdad Mirzaei, Shaghayegh Sahebi, and Peter Brusilovsky. "SB-DNMF: A Structure Based Discriminative Non-negative Matrix Factorization Model for Detecting Inefficient Learning Behaviors" (Not published yet)

The proposal is accepted to the AIED 2019 Doctoral Consortium:

- Mehrdad Mirzaei and Shaghayegh Sahebi. "Modeling Students' Behavior Using Sequential Patterns to Predict Their Performance" In International Conference on Artificial Intelligence in Education, pp. 350-353. Springer, Cham, 2019.

I, Mehrdad Mirzaei declare that this thesis titeld, "Discriminative Factorization Models for Student Behavioral Pattern Detection and Classification" and the work presented in it are my own.

CONTENTS

ABSTRACT iii

ACKNOWLEDGMENT iv

LIST OF FIGURES ix

1. Introduction 1

1.1 Motivation 1

1.2 Research Questions 2

1.3 Contribution 2

1.3.1 Annotated Examples and Parameterized Exercises: Analyzing Students' Behavior Patterns 3

1.3.2 Detecting Trait vs. Performance Student Behavioral Patterns Using Discriminative Non-negative Matrix Factorization 3

1.3.3 SB-DNMF: A Structure Based Discriminative Non-negative Matrix Factorization Model for Detecting Inefficient Learning Behaviors 3

1.3.4 Predicting Student Performance using Sequential Patterns 4

2. Related Work 5

2.1 Student Behavior Analysis 5

2.2 Student Sequence Modeling 6

2.3 Student Performance Prediction 8

2.4 Matrix Factorization 9

3. Datasets 12

3.1 Mastery Grids 12

3.2 OLI Psychology 15

3.3 OLI Statistics 17

4. Annotated Examples and Parameterized Exercises: Analyzing Students' Behavior Patterns 19
4.1 Introduction 19
4.2 Modeling Student Behavior 20
4.2.1 Coding Student Behavior 20
4.2.2 Sequential Pattern Mining 21
4.2.3 Building Pattern Vectors 23
4.3 Behavior Stability Analysis 23
4.3.1 Randomized Analysis 23
4.3.2 Longitudinal Analysis 24
4.3.3 Complexity Analysis 25
4.4 Behavior Cluster Analysis 25
4.4.1 Pattern Analysis 25
4.4.2 Evaluation 28
4.4.3 Performance Analysis 30
4.5 Conclusion 35
5. Detecting Trait vs. Performance Student Behavioral Patterns Using Discriminative Non-negative Matrix Factorization 39
5.1 Introduction 39
5.2 Discriminative Learning of Student Behavior 40
5.3 Experiments 43
5.3.1 Hyper-parameter Tuning 43
5.3.2 Clustering Patterns 43
5.3.3 Latent Factors Analysis 47
5.4 Conclusion 51
6. SB-DNMF: A Structure Based Discriminative Non-negative Matrix Factorization Model for Detecting Inefficient Learning Behaviors 52
6.1 Introduction 52

6.2 Modified Weighted Levenshtein (MWL) Distance Measure 53
6.3 Structure-Based Discriminative Non-negative Matrix Factorization (SB-DNMF) . . 55
6.3.1 SB-DNMF: the Model 55
6.3.2 Learning Model Parameters 57
6.4 Experiments 59
6.4.1 Evaluating SB-DNMF's Fit 59
6.4.2 Finding Trait vs. Performance Patterns 61
6.4.3 Evaluating SB-DNMF's Clustering 64
6.4.4 Detecting Performance Patterns within each Student Group 64
6.5 Conclusion 73
7. Matrix Factorization for Performance Prediction 83
7.1 Introduction 83
7.2 Performance Prediction: Problem Formulation 83
7.2.1 KNN on Original Space 84
7.2.2 Discriminative Latent Factors 84
7.2.3 Multi-View Model 86
7.2.4 Multi-View and Discriminative Latent Factors 87
7.2.5 Multi-View and Similarity 87
7.2.6 Multi-View, Similarity and Discriminative Latent Factors 88
7.3 Conclusion 94
8. Conclusion 95
8.1 Limitations 96
8.2 Discussion 96
8.3 Future Work 97
A. Appendix 98

LIST OF FIGURES

3.1 Progressor+ interface. The students are free to choose the topic and the learning material. 12

3.2 On the left, the student solves the problem incorrectly. On the right, he is able to try again the same problem with new parameters. 13

3.3 Mastery Grids 14

3.4 Mastery Grids 14

3.5 Grades Distribution in Mastery Grids 15

3.6 OLI Psychology 16

3.7 OLI Psychology 16

3.8 Grades Distribution In OLI Psychology 16

3.9 OLI Statistics 17

3.10 OLI Statistics 17

3.11 Grades Distribution In OLI Statistics 18

4.1 Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters in Mastery Grids. 26

4.2 Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Psychology. 27

4.3 Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Statistics. 28

4.4 Top 30 patterns and their frequencies in 3 clusters using generalisation. Patterns are ordered by the maximum difference of frequencies between three clusters in Mastery Grids. 29

4.5 Top 30 patterns and their frequencies in 3 clusters using generalisation. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Psychology. 29

4.6 Top 30 patterns and their frequencies in 3 clusters using generalisation. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Statistics. 30

4.7 High, medium, and low-performance (based on pre-test) student frequencies in three clusters generated based on student patterns in Mastery Grids. 30

4.8 High, medium, and low-performance (based on post-test) student frequencies in three clusters generated based on student patterns in Mastery Grids. 31

4.9 High, medium, and low-performance (based on learning gain) student frequencies in three clusters generated based on student patterns in Mastery Grids. 31

4.10 High, medium, and low-performance (based on pre-test) student frequencies in three clusters generated based on student patterns in OLI Psychology. 32

4.11 High, medium, and low-performance (based on post-test) student frequencies in three clusters generated based on student patterns in OLI Psychology. 32

4.12 High, medium, and low-performance (based on learning gain) student frequencies in three clusters generated based on student patterns in OLI Psychology. 33

4.13 High, medium, and low-performance (based on post-test) student frequencies in three clusters generated based on student patterns in OLI Statistics. 33

4.14 Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 1 in Mastery Grids 34

4.15 Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 2 in Mastery Grids 34

4.16 Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 3 in Mastery Grids 35

4.17 Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 1 in OLI Psychology 35

4.18 Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 2 in OLI Psychology 36

4.19 Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 3 in OLI Psychology 36

4.20 Patterns with significant difference of frequency for low performance and high performance (post-test score) students in Cluster 1 in OLI Statistics 37

4.21 Patterns with significant difference of frequency for low performance and high performance (post-test score) students in Cluster 2 in OLI Statistics 37

4.22 Patterns with significant difference of frequency for low performance and high performance (post-test score) students in Cluster 3 in OLI Statistics 38

5.1 Most frequent patterns are extracted from sequences by CM-SPAM. These patterns are rows of matrix X and students are columns. We split matrix X based on the performance of the students to X_1 and X_2. Then with discriminative non-negative matrix factorization, common and distinct latent factors are extracted. 42

5.2 Heatmap shows the distribution of latent factors in common and discriminative parts for Mastery Grids. 44

5.3 Heatmap shows the distribution of latent factors in common and discriminative parts for OLI Psychology 48

5.4 Heatmap shows the distribution of latent factors in common and discriminative parts for OLI Statistics 49

5.5 Latent factors for 6 clusters and respective patterns in Mastery Grids 50

5.6 Latent factors for 6 clusters and respective patterns for OLI Psychology 50

5.7 Latent factors for 6 clusters and respective patterns for OLI Statistics 51

6.1 Distances between attempts used in the Modified Weighted Levenshtein distance measure. The distance between any two attempts is the weight of shortest path in the graph. For example $d(s, F) = 1$ or $d(e, S) = 1$ (In OLI datasets, “E” and “e” are replaced with “H” and “h”) 54

6.2 The process of building the matrix $W_{c,d}$ for clustering the patterns. X_1 and X_2 are pattern-student matrices for low- and high- performance students. We decompose them and apply the similarity matrix S to enforce having similar patterns in each cluster. The matrix $W_{(1,2)c}$ is the average of $W_{1,c}$ and $W_{2,c}$. 58

6.3 Latent factors 0 to 9 represent common patterns, 11 to 20 represent low performance students' patterns and 22 to 29 represent high performance students' patterns in Mastery Grids. Patterns are clustered using Spectral clustering algorithm in 6 groups. In each cluster, one or more latent factors have higher weights. 65

6.4 Latent factors 0 to 5 represent common patterns, 7 to 19 represent low performance students' patterns and 21 to 33 represent high performance students' patterns in OLI Psychology. 66

6.5 Latent factors 0 to 3 represent common patterns, 5 to 19 represent low performance students' patterns and 21 to 35 represent high performance students' patterns in OLI Statistics. 67

6.6 Pattern clusters with NMF on Mastery Grids 68

6.7 Pattern clusters with NMF on OLI Psychology 69

6.8 Pattern clusters with NMF on OLI Statistics 70

6.9 $H_{1,d}$ is the student-latent factor matrix and represents low performance students. The rows are ordered by students' post-test scores. The latent factor 3 has a correlation with post-test score and its values decrease with higher scores. 74

6.10 $H_{2,d}$ is a student-latent factor matrix and represents high performance students. The rows are ordered by students' pre-test scores. The latent factor 4 has a correlation with pre-test scores and its values increase with higher scores. 75

6.11 $H_{1,d}$ is the student-latent factor matrix and represents low performance students. The rows are ordered by students' post-test scores. The latent factor 12 has a correlation with post-test score and its values increase with higher scores. 76

6.12 $H_{2,d}$ is the student-latent factor matrix and represents high performance students. The rows are ordered by students' pre-test scores. The latent factor 4 has a correlation with pre-test score and its values increase with higher scores. 77

6.13 $H_{2,d}$ is the student-latent factor matrix and represents high performance students. The rows are ordered by students' post-test scores. The latent factor 7 has a correlation with post-test score and its values decrease with higher scores. 78

6.14 $H_{2,d}$ is the student-latent factor matrix and represents high performance students. The rows are ordered by students' post-test scores. The latent factor 4 has a correlation with post-test score and its values decrease with higher scores. 79

7.1 Classification Accuracy on Mastery grids for 50% train and 50% test 89

7.2 Classification Accuracy on Mastery grids for 60% train and 40% test 89

7.3 Classification Accuracy on Mastery grids for 70% train and 30% test 89

7.4 Classification Accuracy on Mastery grids for 80% train and 20% test 90

7.5 Classification Accuracy on Mastery grids for 90% train and 10% test 90

7.6 Classification Accuracy on OLI Psychology for 50% train and 50% test 90

7.7 Classification Accuracy on OLI Psychology for 60% train and 40% test 91

7.8 Classification Accuracy on OLI Psychology for 70% train and 30% test 91

7.9 Classification Accuracy on OLI Psychology for 80% train and 20% test 91

7.10 Classification Accuracy on OLI Psychology for 90% train and 10% test 92

7.11 Classification Accuracy on OLI Statistics for 50% train and 50% test 92

7.12 Classification Accuracy on OLI Statistics for 60% train and 40% test 92

7.13 Classification Accuracy on OLI Statistics for 70% train and 30% test 93

7.14 Classification Accuracy on OLI Statistics for 80% train and 20% test 93

7.15 Classification Accuracy on OLI Statistics for 90% train and 10% test 93

CHAPTER 1
Introduction

The field of Educational Data Mining has been growing in recent years due to unprecedented amounts of student activity data from computer-based interactive learning environments. Data mining, Machine Learning, and Computational Modeling are used in educational data mining to analyze student behavior, predict outcomes, and evaluate student performance in the learning process to improve the overall quality of education and learning outcomes.

1.1 Motivation

Modeling students' behavior and predicting their performance are the main goals of this research. These are significant because both educators and students take advantage of the outcomes. By modeling the students' behavior, we can deduce patterns that are useful and encourage students to pursue them. In addition, we detect harmful patterns that students should avoid. Predicting students' performance in the early stages of the course matters, since educators can investigate if there are students at risk of failing and need assistance in their studies.

With the improvement of online learning systems, students are provided with more opportunities for learning. In modern learning systems, students are usually free to choose to access multiple learning material types. While some systems provide restrictions on the order of accessing learning content, in many systems, there are no predefined activity sequences and students are free to choose to work with any learning materials in any order. This choice provides students with more freedom to learn according to their own pace and background knowledge, and to repeat their past activities as they see fit. For example, students can skip some learning materials and work on the more advanced ones if they believe that they have already mastered the prerequisite concepts. Similarly, they can go back and repeat some learning materials. Despite the advantages, this freedom could lead to some inefficient and non-productive behavior. For example, past research on students' problem-solving behavior has found that students tend to practice the same set of concepts, well after mastering them, instead of moving to new concepts and more difficult problems [79, 34, 28].

In this dissertation, we aim to extract patterns from students' sequential behaviors, detect patterns that are associated with the students' performance, discover patterns that are common traits or performance traits, and predict students' performance using such patterns. Our findings are beneficial for students, educators, and decision-makers to improve students' learning behavior.

1.2 Research Questions

Ideally, a learning system should be able to detect inefficient behavior and guide the students towards efficient ones. To do this, the main challenge is to understand the relationship between students' behavior and performance. This challenge translates into these questions:

1. Could we discover stable student behavior patterns that could be recognized in-time to react? Are they persistent, or they happen at random?
2. Could we detect behavioral traits from performance traits and find behavioral patterns that are common and specific for students with different performance?
3. Could we recognize efficient and inefficient student behavioral patterns by associating them with their learning performance?
4. Could detecting efficient and inefficient patterns help us to better predict students' performance?

1.3 Contribution

The main contribution of this dissertation is designing a framework to answer the research questions. The methodology that we invented is able to find behavioral learning patterns that are stable and are specific for each student. We proposed a structure-based discriminative non-negative matrix factorization model to detect students' learning traits and performance traits from sequences of actions in an online learning environment. Using the latent factors, we detected the patterns that are efficient and inefficient. Our experiments on 3 real-world datasets demonstrated that we can find the association between students' behavior patterns and performance.

1.3.1 Annotated Examples and Parameterized Exercises: Analyzing Students' Behavior Patterns

Recent studies of student problem-solving behavior have shown stable behavior patterns within student groups. In this chapter, we study patterns of student behavior in a richer self-organized practice context where student worked with a combination of problems to solve and worked examples to study. We model student behavior in the form of vectors of micro-patterns and examine student behavior stability in various ways via these vectors. To discover and examine global behavior patterns associated with groups of students, we cluster students according to their behavior patterns and evaluate these clusters in accordance with student performance.

1.3.2 Detecting Trait vs. Performance Student Behavioral Patterns Using Discriminative Non-negative Matrix Factorization

Behavioral patterns can further be used to group the students into different clusters. However, as these clusters include both high- and low-performance students, the relation between the behavioral patterns and student performance is yet to be clarified. As many of these patterns are common between different student groups, discriminating between the patterns that are representative of students' performance versus the ones that are showcasing students' learning behavior traits is a difficult task. In this chapter, we study the relationship between students' learning behaviors and their performance, in a self-organized online learning system that allows them to freely practice with various problems and worked examples. We discover both the prevalent behavioral patterns in each group and the shared patterns across groups using discriminative non-negative matrix factorization. Our experiments show that we can successfully detect such common and specific patterns in students' behavior that can be further interpreted into student learning behavior trait patterns and performance patterns.

1.3.3 SB-DNMF: A Structure Based Discriminative Non-negative Matrix Factorization Model for Detecting Inefficient Learning Behaviors

In previous chapter, we demonstrated that many students, whether high-performing or low-performing, show some inefficient behavioral patterns in these online learning systems. Since the micro-patterns are in the form of strings and have structural similarities, we use this property to improve the model. In this chapter, we propose SB-DNMF, a structure-based discriminative non-negative matrix factorization model to distinguish between common and distinct learning behavior patterns of high- and low-learning gain students. Our proposed model can discover latent groups of students' behavioral micro-patterns while accounting for the structural similarities between these micro-patterns. To capture the structural similarities between the behavioral micro-patterns, we propose a weighted edit-distance measure that can consider different types of learning materials, attempt correctness in graded learning material, and attempt time duration. Our experiments on 3 real-world dataset demonstrated that we can find meaningful latent factors that are associated with students' learning gain and can cluster the behavioral patterns into high-performing, low-performing, and common groups. Also, we analyze these latent factors in each of the high- and low- learning gain student groups and show that they can further discriminate between students according to another performance measure: the pre- or post-test scores of students.

1.3.4 Predicting Student Performance using Sequential Patterns

Having common and discriminative students' behavioral patterns as features, can we predict the performance? In previous chapters, we demonstrated that there are learning traits that are common among students and performance-based traits that are specific for low-performance and high-performance students. Using these traits as features, we propose a novel discriminative classification model to predict if the student is a low-performer or a high-performer.

CHAPTER 2

Related Work

Online learning systems contain different learning materials such as problems, examples, quizzes and lecture videos. Previous studies have shown that solving problems and answering questions is very important for student learning. Other materials such as lecture videos are important but interactive activities will improve students' learning outcomes [46]. Learning theories have suggested that active learning is superior than passive form of learning [4, 20, 73]. So in this work, we focus on the problems that student solve and examples and hints that are feedback the students get from problems.

In the following, we review how student behaviors analysis, student sequence modeling, student performance prediction and matrix factorization are related in this context:

2.1 Student Behavior Analysis

Researchers in Educational Data Mining filed, study a variety of areas. A key area of EDM applications is studying student behavior. In this section, we highlight the importance of student behavior analysis and its applications.

Online educational systems collect increasing volumes of information from students' interactions with various kinds of learning content. The increase in information collection entailed to utilize such data to improve learning process. Students interactions with an online environment is known as students behavior. Student behavior analysis is of high importance in educational data mining for both students and educators. Intervention, gamification, performance, drop-out and retention prediction could be the outcome of these analysis [96, 48, 95, 19] and are challenges for the future of educational data mining [6]. While early works focused mostly on cumulative factors such as frequency of watching videos or using discussion forums [76], recent works attempted to build more complex models of student behavior and identify various kinds of behavior patterns to help students make progress and improve education outcomes. Some works in the literature focused on student trajectories [94, 41]. For example, Boubekki et al. [8] compared the navigation behavior of students in reading textbooks such as page-clicks or scrolls and discovered student clusters that were indicators of student performance. The data used in their work is fine-grained and is in the clickstream level. Our work is in a higher level and steps are solving a problem or reading an example rather than investigating the clicks and scrolls. Sawyer et al. [72] proposed a time-series representation of student problem-solving trajectories in a learning game. They used Euclidean distance and trajectory slope to measure students' distance with "expert paths", which was correlated with students' learning gain. Behavior activity logs in a programming course with historical student data such as characteristics and demographics are leveraged in [5] to identify at-risk students. They showed that personalized guidance can be recommended to students to perform better in exams. In our work, we use the learning material type and the amount of time spent on it, and no auxiliary data such as demographic and clickstream data is used. In this way, our work is different from such works.

Another application of student behavior analysis is to predict dropout in online open-access courses [11, 90, 3]. In [18], using logged data, researchers employed intervention strategies in an attempt to change students' behaviors, recommending productive behaviors and avoiding non-productive behaviors. Social and behavioral features are used in [27] to predict dropout and certification in MOOC. In [42], interaction patterns of students with video lectures, forum participation and assessments in MOOCs are used to classify the students and identify their engagement. Hew et al. in [31], modeled students behaviors with sentiment analysis and hierarchical linear modeling to predict the students satisfaction in MOOCs. They have shown that course instructor, content, assessment, and schedule are main factors of students' satisfaction. Students' learning patterns are explored in [35] to cluster students with similar behaviors in a blended learning course. The detected groups of the students are positive interactive group, stable learning group, positive teaching material group, and negative learning group. In [61], a transfer learning approach is presented that works based on transductive principal component analysis and correlation alignment loss term. Using clickstream data in a MOOC, they investigate the transferablity of dropout prediction across MOOCs.

2.2 Student Sequence Modeling

One approach that is extensively used to mine students' behavior is exploiting sequential pattern mining techniques. Sequential pattern mining problem is primarily introduced in [1] to find sequential patterns in a database of transaction sequences. The sequence mining algorithms are then used in genome searching, web logs, health data and informational retrieval, and was not limited to transaction databases [58]. Researchers also used these algorithms in educational data mining area. For example sequence mining is used in [37] to discover learning patterns in flipped classes and in [77] to mine patterns from the sequence of interactions with an application that students have in an online programming course. The interactions in these works are in the clickstream level and very fine-grained. In another work [56], authors extracted frequent action sequences in a collaborative learning environment to distinguish high achieving student from low achieving students. The analysis in this work is on interaction with resources on a tabletop. Students' actions on the tabletop are logged and coded into events to create sequences and frequent sequential patterns are extracted using n-grams. In [28], Guerra et al. modeled and analyzed the patterns that students work with parameterized exercises. In this work, micro-patterns are extracted using a sequential pattern mining algorithm and used to build student behavior profiles ("genomes"). Then, students with similar genomes are clustered into behavior groups. It is shown that sequential patterns that are extracted from interacting with problems are stable during the course. The results showed that there are distinctive patterns in each group, although, some patterns are common between the two groups. SCOVA is introduced in [21] which is a general method for evaluating alternative activity sequences. This method explores complex sequencing constraint, such as perquisite relationship and blocking to evaluate the activity sequences. Analyzing students' behavior working with multiple online platforms in [26] has shown meaningful patterns that are helpful for both instructors and students. The transitions between online platforms are considered as activities and N-gram is used to mine the sequences. In our work, we focus on interactions within one online platform, hence such approaches are different from ours.

Since student behavior is commonly considered as a sequence of students' actions or interactions with the system, various kind of sequence-oriented Markov models were explored for behavior analysis. For example, Hansen et al. [29] analyzed log data of an online education system, and modeled student behavior as interpretable Markov chains. The model compares action sequences across different lengths, focusing on the flow of actions. A two-layer Hidden Markov Model is used in [24] to automatically detect student behavior patterns from logged data of a MOOC platform. They have shown that the extracted patterns are meaningful and have a correlation with students' learning outcome. Authors in [78] discovers the learners' study patterns in MOOC assessment periods. Clustering the study patterns sequences, different activity profiles for students are captured. In a similar work in [7], students' study patterns are extracted from activity sequences. This work could be deployed during the course and support students in real-time. In such works, labeled students' activities are observed to generate latent states and the extracted patterns are relatively long sequences. So the order of using different learning materials are important. In our work, we focus on extracting short patterns (micro-patterns) from sequences and the learning materials are treated equally.

2.3 Student Performance Prediction

Previous researches have shown that students' behaviors can impact their performance since the behavior could be productive or non-productive. Among the explored topics, is the early prediction of student success or failure [11], which could be helpful to identify and support students-at-risk [90]. In prior works, researchers have used supervised learning approaches such as logistic regression [30, 43, 82, 36, 90, 85] and support vector machines [44, 85, 66] on a set of extracted features. The features include students demographic data, grades, type and time of their activities. In [85], regression is used for early prediction of students at risk from their demographic characteristics, number of examination attempts and final grade of each course. In another work [32] stereotypes are used for user modeling and personalization of students. Performance factors analysis based on logistic regression is used in this work to predict student problem-solving and learning behaviors. Eye-gaze activities in collaborative problem-solving tasks are studied using regression analysis in [13] to predict students roles in the team and scores. In another work, Xing et al. used genetic algorithms to predict students performance from their behaviors [91].

A group of researches have focused on the blended courses and found that transition among courses or different learning environments can impact students performance. Pattern mining and natural language processing models are used in [2] in a blended course to predict student performance from clickstream data. They discovered that assessment type make differences in interaction patterns. In [86, 26] authors used log data to predict performance in blended courses. It is shown that there are significant patterns that are associated with students' performance. A comparative study for early prediction of at-risk students is presented in [12] in a blending learning environment and the students performance is predicted in the first week of the course in some cases. Learning analytic and big data are applied in [55] to predicts students' academic performance in a blended course. Interaction with videos, practices, homework and quizzes are used in this work. Patterns in students' learning behavior are investigated in [23] through 10 online learning modules. Correlation of the detected patterns with overall course outcome is found in this work.

A group of researches used graphs related approaches to predict students' performance. In [80], domain knowledge alongside a knowledge graph representation with activity scope based on learning objectives is used to predict students performance in an online math learning environment. A predictive similarity policy is introduced in [69] to predict student performance on the next question. They also have evaluated the instructional policy accuracy. Chunqiao et al. applied Neural Networks to intervene the student and avoid failure or encourage them to pursue productive behaviors [18]. Boumi et. al used Hidden Markov Models to predict students' performance [9, 10]. The enrollment status is exploited to categorize students based on their performance in their work.

Some researches use students information from the past for performance prediction. The students' information from the past together with human-interpretable features are used as features for classification of the poor-performing students in [66]. The experiments in this work, specified the importance of the features to improve the prediction. in [92] a predictor is proposed to predict if the students complete the degrees or not given their current and past performance. Relevant courses are discovered in this work to construct the base predictors.

Another group of researches compares different classifiers to predict students performance. In a recent work [84], students at a high risk of dropping out are discovered and also their performance is predicted. The authors have concluded that adequate data from the interaction of the students with the learning environment ensure better analysis. A hybrid classification method is introduced in [67] that is a classification and visualization tool and predicts students' performance. These methods have extracted features that are specific for the used datasets.

2.4 Matrix Factorization

In [47] matrix factorization (MF) is introduced to improve recommendation systems and then it is used in tasks such as text mining [65] and document clustering [74, 93]. Discriminative Matrix Factorization (DMF), a variant of Matrix Factorization is also used for classification [49, 100, 97] and link prediction [38].

Recently, a group of researchers have used Matrix Factorization in EDM to model students' behavior and predict performance. For example in [92], a data-driven approach based on probabilistic matrix factorization is presented to discover course relevance. The latent factors are used in ensemble based predictors to predict students performance from their evolving performance states. In [60, 59], NMF is used to detect high-performance learners' browsing patterns from the collected log data to increase students' thinking skills. A wighted low-rank matrix factorization along with singular value decomposition-based initialization is used in [54] to predict students score in quizzes. The model predicts next quiz score from previous scores. In [98], a new MF method is proposed called graph regularized robust matrix factorization (GRMF). This method uses the side data of students and courses to predict the grades. In another work [40] Bayesian Probabilistic Matix Factorization (BPMF) is presented that uses students' background information for performance prediction. A Multi-Relational Factorization Model is presented in [83] that uses the relationship between students, tasks and skills. This model predicts the students' performance better than the models with students and tasks relationship only. In another research, a biased weighted multi relational matrix factorization is proposed in [62] to predict students' performance. The biases are student bias and the task bias that improved the prediction accuracy of the model. Another advanced way of representing students' behaviors are by using tensors. Sahebi et al. proposed a tensor-based method in [71] to model students' behavior based on the quizzes and predict their performance [71].

Another group of researches focused on utilizing Matrix Factorization to cluster students based on their behavior. For example, Gelman et al. [25] segmented students use of content and assessment, into weeks. Then, they used non-negative matrix factorization (NMF) to distill five basic behaviors of students ("Deep", "Consistent", "Bursty", "Performance", and "Response") and built vectors specifying how much each student shows each behavior. Lorenzen et al. used NMF to cluster students and track their behavioral changes over time [54]. NMF is used in another work [25] to provide a multi-dimensional view of user participation. The authors discovered basic student behaviors across the courses to assist the students improve their learning behaviors. In [81], non-negative matrix factorization is used to discover the topics from the co-occurrence relationships between extracted keywords in students' journals. The score for each word is based on the TF-IDF term weighting scheme. They could identify four significant impression topics from students' writing. Authors in [50] explored how behavioral data can be used to model students' engagement. They proposed a statistical method to factor the learner-question matrix to extract latent concept knowledge.

Another group of works used Additive Factor Models based methods to predict students' performance. Additive latent effect models are proposed in [68] to predict students' grades. The model incorporates factors such as student's academic level and course instructors for prediction. In [64], authors added collaborative features to a standard Additive Factor Model to predict student performances in groups. In another work [16] three different Additive Factors Analysis based model are compared to predict student performance. Students' response times and their knowledge components are used for prediction in this work.

A subset of MF is discriminative non-negative matrix factorization that is to find common and discriminative latent factors in different sets of data. For example joint discriminative non-negative matrix factorization has been used previously in [39] to discover common and distinctive topics in documents. Their topic modeling method simultaneously finds common and distinct topics from multiple datasets. A cluster-Driven non-negative matrix factorization called CD-NMF is introduced in [75] to discover common and distinct structural connectivity patterns between two different mental diseases. A semi-supervised NMF model is proposed in [87] to be generalizable when the dataset contains outliers and limited knowledge from domain experts. The model uses a structured normalization method to normalize the coefficients and incorporate the discriminative information. FacIt is a generic visual analytic system for tensor factorization presented in [88] that can be used to compare pair of common and discriminative patterns and their associated items. In [99], algorithm DICS based on NMF is presented and used as a classifier to discover discriminative and non-discriminative information from different views. The algorithm is designed to extract discriminative information from the common and the view-specific parts and it is illustrated that those are very helpful for improving students' learning performance. In [89], multi-way interactions are considered as behavior and common and discriminative patterns are discovered with a framework of iterative discriminant factorization. Hierarchical discriminant tensor factorization is used in [89] to discover patterns in a multi-level way and explore two performance groups.

Non-negative Matrix Factorization can be formulated as an optimization problem [52]. Iterative algorithms such as gradient descent are used effectively to solve matrix factorization problems and approximate low-rank matrices[15]. There exist numerical methods for solving MF such as Newton-Type approach [17]. Although approaches are different and not all methods work equally, there is not the best method to solve MF. However, for small matrices, the optimization works well and approximation of low-rank matrices satisfies.

CHAPTER 3

Datasets

In this work, we run the experiments on three different datasets. The datasets are Mastery Grids, OLI Psychology and OLI Statistics and are described in the following:

3.1 Mastery Grids

In our experiments, we use the student interaction data with learning materials in the "Introduction to object-oriented programming" course using *Progressor+* interface [33]. The system includes parameterized exercises and worked-out code examples as two different types of learning material. The learning materials in the practice system were grouped into topics. For each topic, multiple problems and examples are available. Although the order of topics was shown to the students, they could choose any topic, problem, or example to practice at any time in any preferred order. Interface of the system is demonstrated in Figure 3.1.

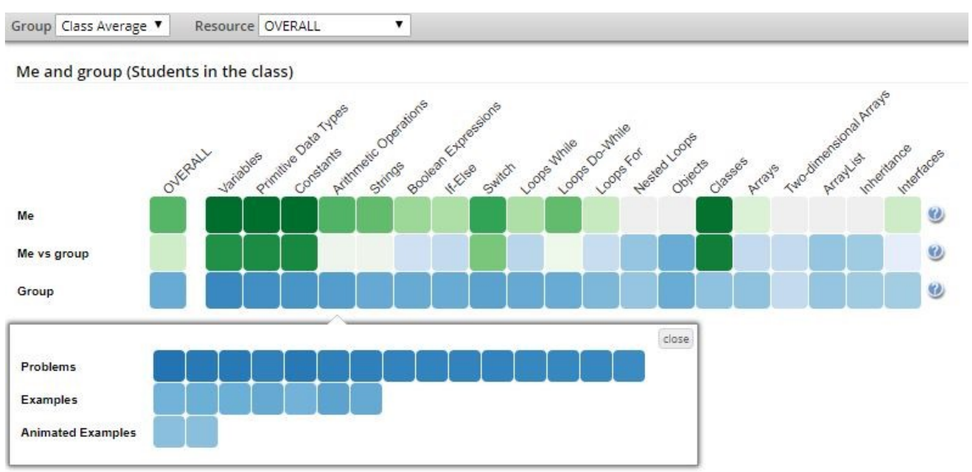


Figure 3.1: Progressor+ interface. The students are free to choose the topic and the learning material.

The parameterized exercises are small problems focused on program behavior prediction. Each exercise is a template with a parameter, which is generated randomly every time a student chooses to work on them. Consequently, students can use the same exercise template for practice multiple times with different parameters. A sample parameterized exercise is shown in 3.2.

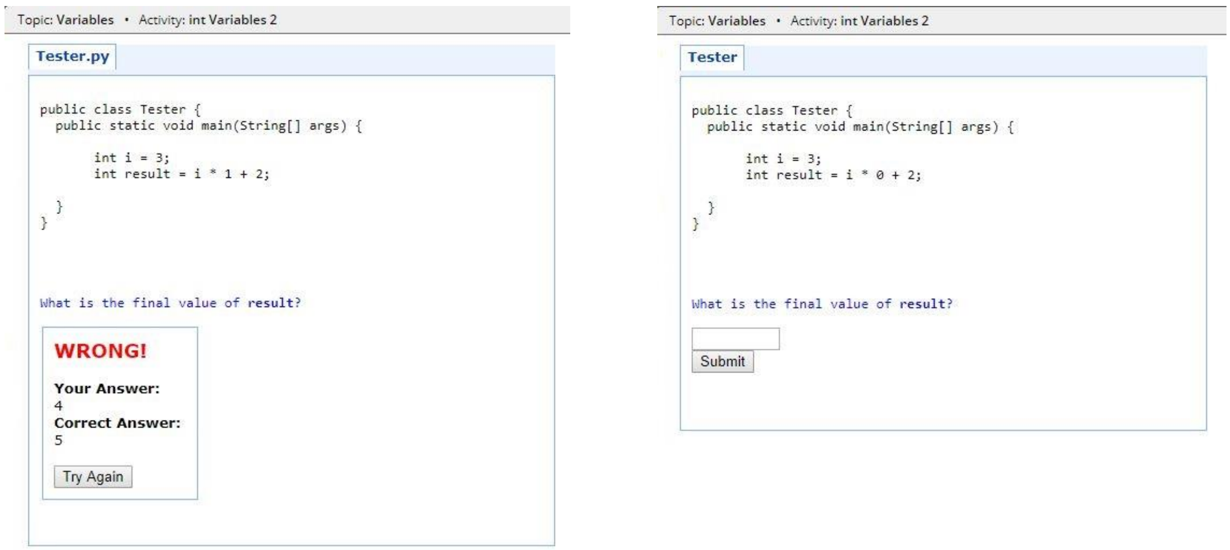
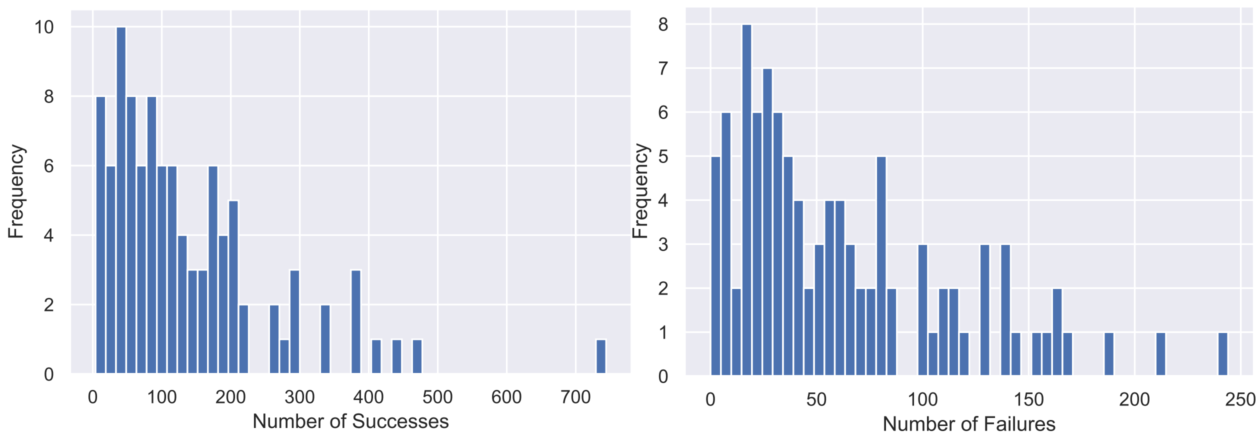


Figure 3.2: On the left, the student solves the problem incorrectly. On the right, he is able to try again the same problem with new parameters.

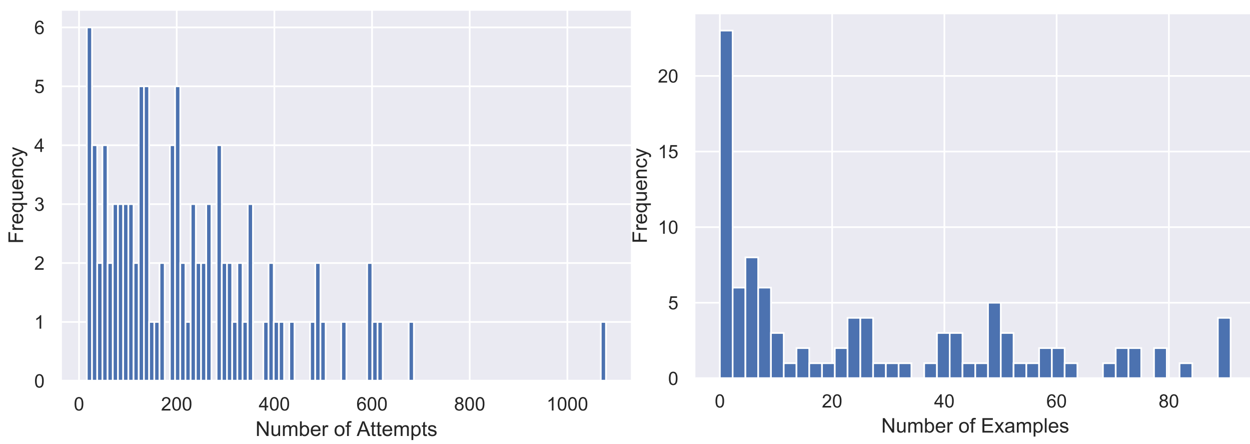
Worked-out annotated code examples are small complete programs annotated with short explanations for each line of code. Students can click on the lines of code in any order to read the explanation. The dataset includes three semesters of students activities in Java classes in a large public US university. After data cleaning, the dataset contains 83 students. There are 103 parameterized problems and 42 annotated examples in the dataset. The number of correct attempts to solve problems is 13796, the number of incorrect attempts is 6233, and the number of clicks on examples is 12713. Student sequence length in each session varies between 1 and 30, with an average of 2.33 activities. 61.2% of activities are on problems, and 38.8% are on examples. The average student success rate on problems is 68%. The histogram of number of attempts, success, failure and hint are shown in Figure 3.3 and Figure 3.4. These charts indicate that a few students have used the system very frequently. However, most of the students tend to have short attempts. Also, the number of examples the students read is typically low and doesn't follow the distribution of the problems solved.

In addition to the students behavior log, the dataset includes pre-test and post-test scores for each student. The pre- and post-tests included the same set of program behavior prediction questions administered at the beginning and at the end of each semester correspondingly. The minimum and maximum score in pre-test are 0 and 14 and in post-test are 5 and 24 respectively. To measure students' improvement over the course of the semester, normalized learning gain is



(a) Successes Distribution **(b)** Failures Distribution

Figure 3.3: Mastery Grids



(a) Attempts Distribution **(b)** Examples Distribution

Figure 3.4: Mastery Grids

calculated for each student as the normalized difference between post-test and pre-test:

$$\text{Normalized Learning Gain} = \frac{\text{post-test} - \text{pre-test}}{\max(\text{post-test}) - \min(\text{pre-test})}$$

The grades distribution for pre-test, post-test and learning gain are shown in Figure 3.5. The figures show that the students' pre-test score is low in comparison with post-test scores. It means that the majority of the students don't have the course knowledge before taking it, although a small number of students have high pre-test scores and are familiar with the concepts. But the post-test scores and learning gain have a different distribution and most of the students fall in the middle of the chart.

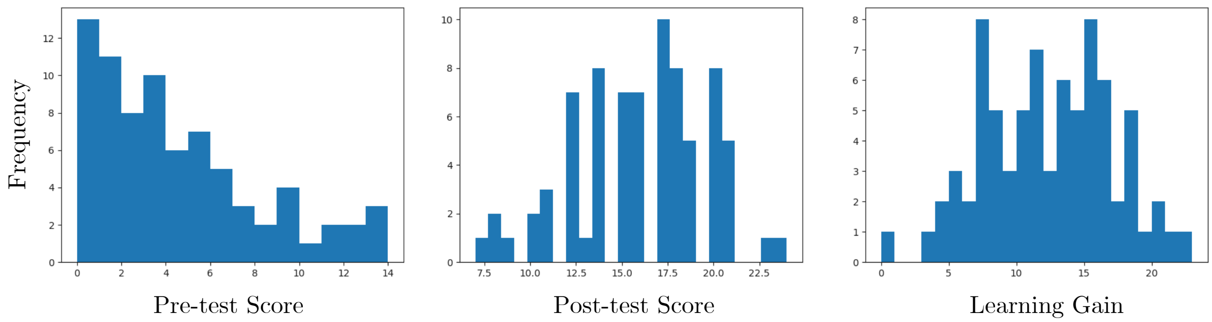


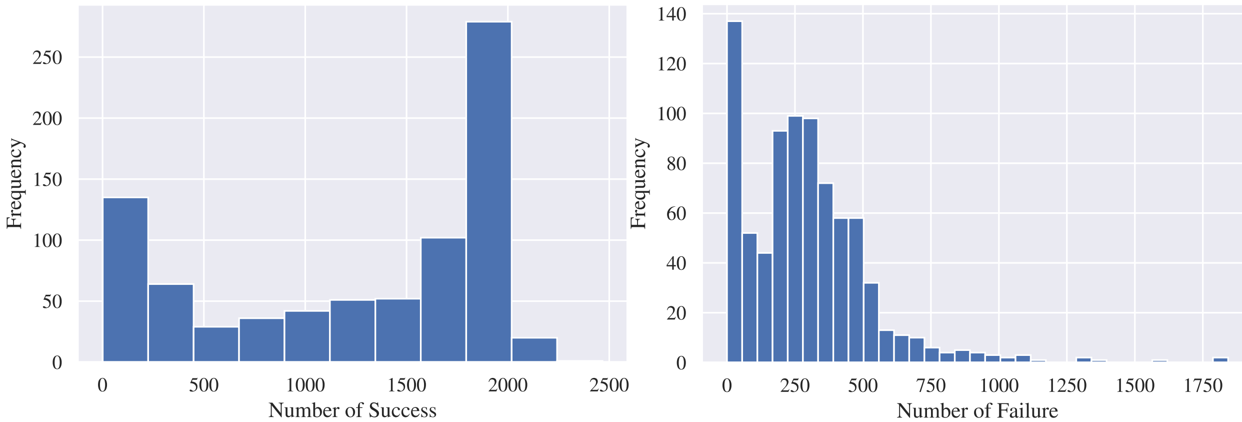
Figure 3.5: Grades Distribution in Mastery Grids

3.2 OLI Psychology

"Introduction to Psychology as a Science" is a 12-week introductory course that incorporated elements of the OLI[1] "Introduction to Psychology" learning environment. We used the "Psychology MOOC GT - Spring 2013" dataset accessed via DataShop[2] [45]. Course materials included a set of problems with hints that student were able to use hints or not in answering the problems. The dataset contains 811 students and 1998 problems. The number of correct attempts to solve problems is 1012761, the number of incorrect attempts is 237694, and the number of hints used is 4479. The students' pre- and post- test scores for this course are also available. The minimum and maximum score in pre-test are 0 and 19 and in post-test are 0 and 35 respectively. The normalized difference between post-test and pre-test score is calculated as learning gain. The histogram of number of attempts, success, failure and hint are shown in Figure 3.6 to Figure 3.7. The figures show that most of the time students have successful attempts and the number of failures is moderately low. Also, the number of hints used is way smaller than the number of problems solved.

The pre-test scores show that students know about the course and the course is not new for them unlike the students is the Mastery Grids. The grades distribution for pre-test, post-test and learning gain are shown in Figure 3.8. The post-scores demonstrate that students have high scores at the end of the course. It might be because of the initial knowledge that the students have before taking the course. In this way the learning gain is a decent measure to assess the students' performance. It makes the students' patterns to be biased toward problem solving as opposed to using hints.

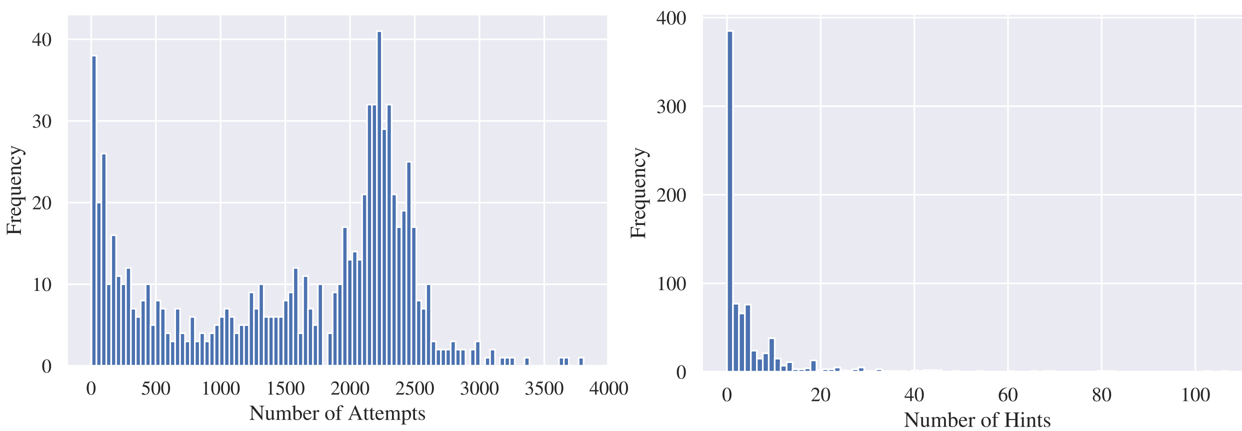
[1] Open Learning Initiative
[2] pslcdatashop.org



(a) Successes Distribution

(b) Failures Distribution

Figure 3.6: OLI Psychology



(a) Attempts Distribution

(b) Hints Distribution

Figure 3.7: OLI Psychology

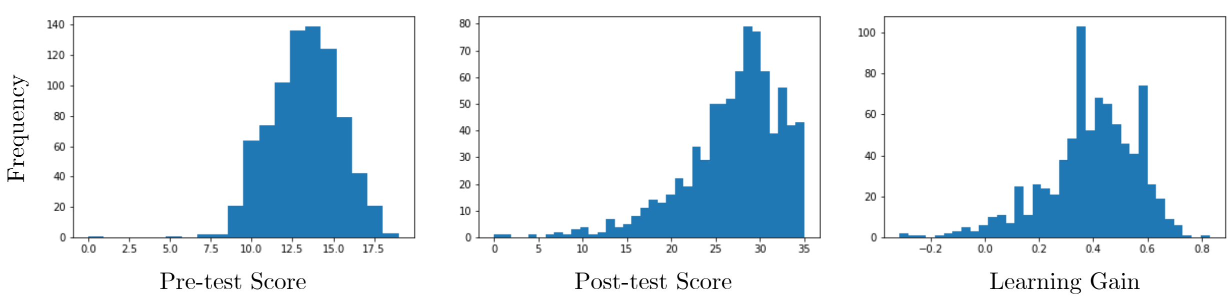
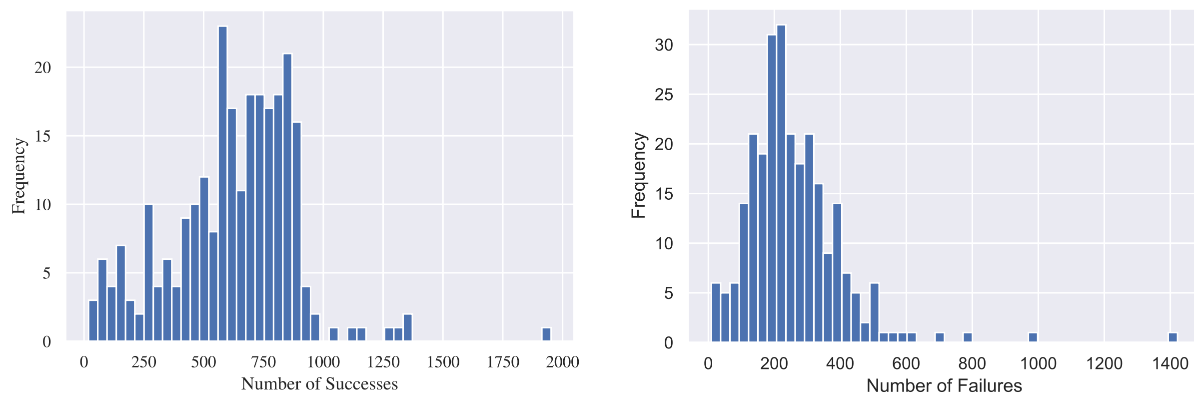
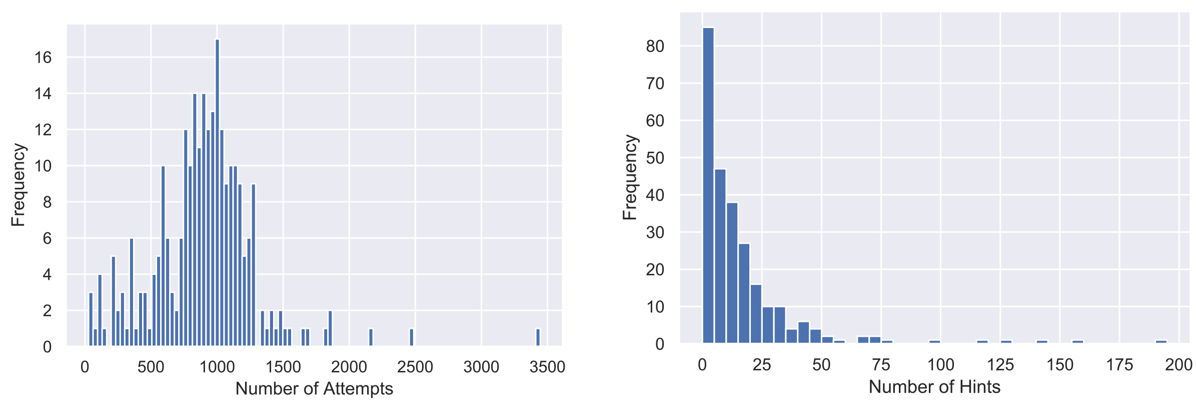


Figure 3.8: Grades Distribution In OLI Psychology



(a) Successes Distribution **(b)** Failures Distribution

Figure 3.9: OLI Statistics



(a) Attempts Distribution **(b)** Hints Distribution

Figure 3.10: OLI Statistics

3.3 OLI Statistics

The third dataset that we used is the "Statistics (2010-10-01 03_2014-12-30)" dataset accessed via DataShop. Course materials included a set of problems with hints that student were able to use hints or not in answering the problems. The dataset contains 261 students and 543 problems. The number of correct attempts to solve problems is 162510, the number of incorrect attempts is 67503, and the number of hints used is 4213. The histogram of number of attempts, success, failure and hint are shown in Figure 3.9 and Figure 3.10. The figures show that the number of successful and failed attempts are comparable but the number of hints used are lower relatively. So it is expected that the problem solving behavior is seen more frequently in the students' sequences rather than the hints.

The students' test scores (post-test) for this course is available and are from 1 to 5. Unfortunately, the pre-test scores were not available for the students in this course so we use the final scores

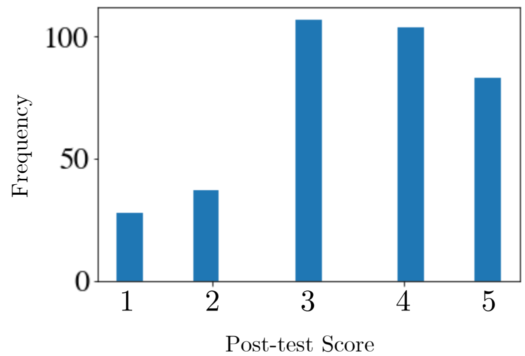


Figure 3.11: Grades Distribution In OLI Statistics

as the students' performance. The grades distribution for post-test is shown in Figure 3.11. The figure demonstrated that the number of students with low scores are low and most of the students have the score 3 or 4. So the the data is biased toward high-performance students.

CHAPTER 4

Annotated Examples and Parameterized Exercises: Analyzing Students' Behavior Patterns

4.1 Introduction

While past research on behavior patterns has mainly focused on problem-solving behavior, student behavior can get more complex as other types of learning materials are introduced. For example, consider a learning system that includes both reading materials and self-assessment problems. Here, a student can spend a significant amount of time persisting in reading on an advanced concept and failing in related problems, without having the prerequisites.

Prior research in the area of student problem-solving behavior indicated that stable patterns of student behavior do exist, however, theses patterns might not be directly related to their performance [28]. Instead, different patterns characterize students' individual ways to learn and approach a problem. To find stable patterns of student behavior, Guerra et al. in [28] built student "problem-solving genomes" from micro-patterns ("genes") and grouped the students based on their "genomes" into clusters. While students belonging to the same cluster tend to show the same behavior patterns, these clusters included both high and low performing students. However, there was an indication that within each cluster, the "genomes" could help to discover efficient and inefficient behaviors.

In this chapter, we attempt to apply the sequence mining-approach suggested in [28] to a more complex case, where students are working with two types of learning materials, and are repeating their attempts within the same topics. The two learning material types we focus on are: parameterized problems and annotated examples. The goal in this chapter is to answer the first research question in 1.2:

Could we discover stable student behavior patterns which could be recognized in-time to react? Are they persistent, or they happen at random?

4.2 Modeling Student Behavior

To extract micro-patterns from student logs, we code them into sequences and analyze them using a frequent pattern mining algorithm. We build macro-pattern vectors or *genome* as a representation of each individual student's behavior.

4.2.1 Coding Student Behavior

To discover behavioral patterns of students, we first label student attempts. Inspired by [28], we focus on two aspects when labeling student problem solving attempts: whether the student succeeds (or fails) in solving the problem, and whether the student spent shorter or longer time to answer a problem, compared to a median answering time. The median answering time is calculated separately for each problem, considering all attempts on it. The median split can be calculated within each students also. Since we are interested in capturing content access differences between students, and since time-spent variance among problems is larger than among students, we chose to split the data according to problem-answering medians. If a student solves a problem correctly in less time than the median, the attempt is labeled as 's' (short success). Likewise, if a student's successful attempt takes longer than the median it will be labeled as 'S' (long success). Similarly, if the student solves a problem incorrectly in a short time (vs. long time), her attempt will be labeled as 'f' (vs. 'F').

In addition to students' problem-solving, we code their example-reading behavior. Unlike the problem-solving attempts, working on annotated examples is not associated with correctness. Thus, we only measure the time spent by each student on each annotated example. To do this, we sum up all sequential student clicks on example lines of one annotated example, as the time spent on that example. We first calculate the median time spent on each annotated example by all students. For each example-reading activity of a student, if the time spent on the annotated example is less than its median, it is labeled 'e', otherwise as 'E'. In OLI datasets, there are hints instead of examples. Hint is a short explanation for each problem and students are able to use and read them. We follow the same approach to find short and long hints used and assign 'h' and 'H' to them accordingly.

The students continue to work in the system during the whole semester. To chunk the large sequence of student actions into smaller comparable sequences, we define a "session" as a consec-

	Pattern	Support		Pattern	Support		Pattern	Support
1	ss	0.1557	11	Fs_	0.0698	21	_FF	0.0499
2	Ss	0.1495	12	Ff	0.0698	22	_Sss	0.0461
3	ss_	0.1415	13	sss_	0.0686	23	FS_	0.0461
4	Fs	0.1184	14	Sss	0.0681	24	_Ss_	0.0455
5	_Ss	0.1000	15	_FS	0.0647	25	_ee	0.0443
6	_Fs	0.0925	16	ee	0.0609	26	fs_	0.0404
7	FS	0.0850	17	ee_	0.0567	27	ssss	0.0383
8	fs	0.0809	18	FF	0.0561	28	ff	0.0377
9	sss	0.0755	19	_Fs_	0.0554	29	Fss	0.0371
10	Ss_	0.0711	20	_Ff	0.0529	30	_FS_	0.0361

Table 4.1: Top 30 extracted patterns from Mastery Grids ordered by support

utive set of student activities within one topic. In other words, a session is a sequence of attempts on parameterized problems and annotated examples inside the same topic. An attempt on an example or problem from another topic starts another session. To indicate session borders within each student's sequence, we insert '_' between two consecutive sessions. For instance, the sequence '_ffSsee_' means that the student has a long success after two short failures, then a short success and finally is quickly examined two annotated examples within the same session.

4.2.2 Sequential Pattern Mining

To discover the most frequent micro patterns of student behavior, we use CM-SPAM [22] sequential pattern mining algorithm. This algorithm discovers patterns that appears frequently in various types of data, including sequences. It calculates the support of a sequential pattern which is the number of sequences that contain the pattern divided by the total number of the sequences. A frequent sequential pattern is a sequential pattern that has a support no less than a threshold which is called minimum support. We set the minimum support to 1% (i.e., we are interested in patterns that could be found in at least 1% of sequences) and require no gap between encoded attempts. Besides that, we only consider the patterns with more than one sequential attempts. The top 30 frequent patterns in Mastery Grids are illustrated in Table 4.1. Interestingly, we can see that the top frequent patterns are either problem-solving micro-patterns or example-reading micro-patterns. In other words, there are no mixed activity patterns (such as 'eF') among the top frequent ones. From this, we conclude that switching from one type of activity to another was considerably more rare than continuing with the same kind of activity.

For OLI Psychology and Statistics, we used a different method to rank the patterns. In this method, for each pattern, we measure how frequently a pattern occurs in a sequence as pattern fre-

	Pattern	Score		Pattern	Score		Pattern	Score
1	fS	66513	11	_Fs	39980	21	fFS_	14799
2	fS_	65463	12	_Fs_	39434	22	_fFS_	14669
3	_fS	64508	13	_fs	33734	23	Ff	14646
4	_fS_	63488	14	_fs_	33416	24	ff	13280
5	fs	51802	15	fF	23815	25	_Ff	12722
6	fs_	51428	16	_fF	22925	26	FF	11579
7	Fs	43622	17	_FS	19727	27	ffs	11346
8	Fs_	43024	18	_FS_	19501	28	ffs_	11248
9	FS	41150	19	fFS	15070	29	_ff	10346
10	FS_	40775	20	_fFS	14930	30	Ffs	9562

Table 4.2: Top 30 extracted patterns from OLI Psychology ordered by support

	Pattern	Score		Pattern	Score		Pattern	Score
1	sS	82568	11	fF	28511	21	sSS	17367
2	_sS	69503	12	_fS_	26458	22	ff	16461
3	sS_	58662	13	_fF	26007	23	sF	16350
4	_sS_	53324	14	ss	25726	24	sf	15411
5	fS	39667	15	FS_	25696	25	fFS	14433
6	FS	37891	16	SS_	23569	26	ss_	14057
7	SS	36812	17	SF	22417	27	_fFS	13556
8	_fS	34055	18	Fs	18839	28	sss	13290
9	fs_	29404	19	SSS	18399	29	fs_	12891
10	fs	29150	20	FF	18011	30	_ss	12656

Table 4.3: Top 30 extracted patterns from OLI Statistics ordered by support

quency (PF). Another measure finds the number of sequences that have a pattern as inverse support frequency (ISF). The score of each pattern is computed by $\frac{PF}{ISF}$. This method weigh down the frequent patterns while scale up the rare ones. The reason for using this method is the low number of hints used by the students so the patterns having hints are very rare among top extracted patterns. The top 30 frequent patterns and their scores in OLI Psychology and Statistics are illustrated in Table 4.2 and Table 4.3. We observe that the hints are still not among the top 30 frequent patterns in OLI datasets and conclude that students are not eager to use hints for solving problems.

The distribution of the attempts have affected the extracted patterns in each of the datasets. As the tables show, in Mastery Grids there are patterns of reading examples. However the patterns containing using hints are not among the top frequent patterns in OLI Psychology and OLI Statistics. The reason is that in these two datasets, number of hints used are very limited. The number successful attempts also impact the frequent patterns. For instance, in OLI Statistics the first 4 top extracted patterns ("sS", "_sS", "sS_" and "_sS_") have only successful attempts. One reason is the high number of successful attempts in comparison with failures in this dataset.

4.2.3 Building Pattern Vectors

The top frequent patterns found in Section 4.2.2 represent a variety of patterns used by all students. Each student could use each micro-pattern with different frequency or not at all. To model the behavior of an individual student, we build a behavioral pattern vector for each student. We use the top frequent patterns to build this vector. This pattern vector includes normalized frequencies of observing each of the top frequent patterns in the behavior log of the modeled student. To build it, we first count the number of times each frequent pattern occurred in the student's sequence. These *absolute* frequencies, however, could vary depending on the total length of student behavior sequence, i.e., how much the modeled student interacted with the system. To capture the relative importance of each micro-pattern regardless of total sequence length, we normalize the count vectors (i.e., the frequency of patterns are summing to one for each vector). These vectors represent the behavior of individual students and are used to discover macro-patterns by clustering student vectors.

4.3 Behavior Stability Analysis

Before establishing a relationship between students' micro-patterns and their performance, we should make sure that the patterns are representative of students' behavioral traits, and not other environmental factors. To do this, we analyze the *stability* of student patterns in three different setups: randomized, longitudinal, and complexity-based. In each of these setups, we split student sequences into two equal sets. Then, we independently build a pair of two pattern vectors for each student: one for each set. If our model of student behavior is stable, the vectors in each pair should be more similar to each other than to vectors form other pairs. Thus, in each of the setups, we test whether the students' behavior vector built from the first set is significantly more similar to their own behavior vector in the second set than to the behavior vectors of the rest of the students. To measure the similarity, we use Jensen-Shannon divergence [53].

4.3.1 Randomized Analysis

In the randomized analysis, our goal is to examine whether a student's pattern vector is stable across all sessions, if we split them randomly. It is to test if we can distinguish a student from other students according to their pattern vectors. To do this, we randomly split student sequences into two halves and build a pattern vector for each half. If a student's pattern vector from the first half is significantly more similar to her own pattern vector in the second half – compared to being similar to other students – then, we conclude that the student's patterns are stable and do not change randomly. To test the significance, we run paired sample t-test. The normality assumption is met. The results are shown in Table 4.4, 4.5 and 4.6. As we can see, the self-distance of two pattern halves (0.2082 on average) is significantly smaller than the distance to other students (0.4639) in Mastery Grids. In other two datasets, the average of self distances are also significantly smaller than the distance to others. We conclude that the pattern vectors are stable for students and is not depending on the course. It means that in each dataset that has a specific course, the students' patterns are stable during that course.

4.3.2 Longitudinal Analysis

Here, we are interested to see if the student patterns change as the semester advances. To study this, we split each student's activity sequence according to a mid-semester point: we build pattern vectors for the first half and the second half of the semester. Similar to randomized analysis, we compare the distance between halves within each student's vector and between student vectors. As shown in Tables 4.4, 4.5 and 4.6 we see that the distance between first half and second half of one student's pattern vector is significantly smaller than the distance to other students. Individual student behavior pattern changes slightly over the semester, yet this change is by far not sufficient to cross the difference from other students.

	Self distance		Distance to others			
	Mean	SE	Mean	SE	t Stat	P-value
Random split	0.2082	0.0207	0.4639	0.0105	-16.0279	<0.0001
First half/second half	0.2995	0.0211	0.5207	0.0113	-12.3501	<0.0001
Random Split (Easy)	0.3644	0.0258	0.5769	0.0110	-9.9099	<0.0001
Random Split (Medium)	0.3266	0.0246	0.5465	0.0092	-11.1404	<0.0001
Random Split (Hard)	0.4219	0.0266	0.5703	0.0106	-6.4266	<0.0001

Table 4.4: Comparing average of students' pattern vector distances with themselves vs. other students according to various splits in Mastery Grids

	Self distance		Distance to others			
	Mean	SE	Mean	SE	t Stat	P-value
Random split	0.0946	0.0038	0.1815	0.0024	-19.0818	<0.0001
First half/second half	0.1171	0.0043	0.1803	0.0026	-12.3925	<0.0001

Table 4.5: Comparing average of students' pattern vector distances with themselves vs. other students according to various splits in OLI Psychology

	Self distance		Distance to others			
	Mean	SE	Mean	SE	t Stat	P-value
Random split	0.1043	0.0065	0.1941	0.0051	-19.1653	<0.0001
First half/second half	0.0767	0.0047	0.1704	0.0046	-25.0777	<0.0001

Table 4.6: Comparing average of students' pattern vector distances with themselves vs. other students according to various splits in OLI Statistics

4.3.3 Complexity Analysis

Another factor that can affect students' behavior is activitiy complexity. Each learning material is labeled with "easy", "medium" or "hard" in Mastery Grids dataset. Accordingly, we build separate pattern vectors for each group of learning activities for each student. E.g., in each topic and session, we separate the "easy" problems and examples as one "easy" session. We assess pattern vectors stability by comparing the difference within a student (comparing according to complexity) and between students. Three rows at the bottom of Table 4.4 represent the distances and statistical tests that show student pattern vectors are stable across learning material complexities. For the other two datasets, the complexity of the problems was unavailable, so this analysis in exclusive for Mastery Grids.

The Stability analysis showed that the patterns are students' behavioral traits. The patterns are stable during the course and do not depend on the complexity of the problems. The stability is inspected in 3 different datasets to ensure that the patterns are specific for students.

4.4 Behavior Cluster Analysis

Having stable student pattern vectors, we aim to distinguish efficient patterns. To do this, we study if student behavioral patterns are associated with student performance. Namely, we would like to understand if students with similar behavior have a similar performance. First, we cluster the students based on their behavior patterns to have students with similar patterns together. Afterward, we analyze the patterns to recognize useful patterns in each cluster.

4.4.1 Pattern Analysis

We apply Spectral clustering [63] on student pattern vectors and group students in 3 clusters. The interpretation analysis showed that 3 clusters will provide the best results. To understand the student differences in each cluster, we compare their average pattern frequencies in the top 30

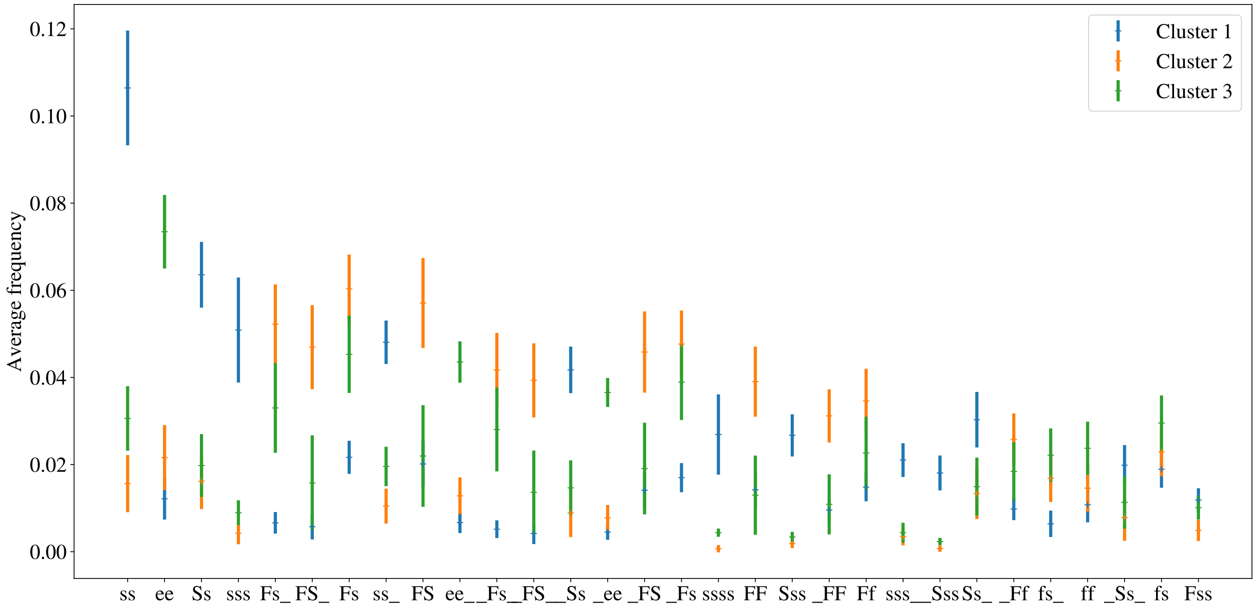


Figure 4.1: Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters in Mastery Grids.

micro-patterns.

Mastery Grids; Figure 4.1 shows these top 30 frequent patterns and their average frequencies in each cluster for Mastery Grids dataset. The error bars show 95% confidence interval for each micro-pattern. The micro-patterns are ordered based on the maximum frequency difference in three clusters. As we can see in the Figure 4.1, in cluster 1, patterns like 'ss', 'Ss' and 'sss' are significantly more frequent than the other two clusters. We can say, students in this cluster tend to repeat practicing an exercise within a topic even if they succeed in it. Significantly frequent patterns in cluster 2 are '_FS_', 'FS' and 'FS_' that demonstrate longer failures and longer successes afterwards. We can conclude that the students in this cluster tend to spend more time on solving a problem, and then succeed afterwards. They do not attempt the problems randomly and do not answer them by chance. Students in cluster 3 read more examples since patterns such as 'ee' and 'ee_' are frequent in this cluster. In [28] students were grouped into "confirmers" and "non-confirmers" according to their patterns. "confirmers" were the students who preferred to confirm their success by repeating it. "non-confirmers" were the ones who ended their session right after having a short success. Here, we see a "confirmers" type of pattern in cluster 1. However, in cluster 2 students are mostly "thinkers" rather than "non-confirmers". They fail and then succeed, but with thinking and spending time on the activity. Students in cluster 3 are mostly "readers". They tend to spend more time on reading the annotated examples. Therefore, we can recognize 3 types of behaviors

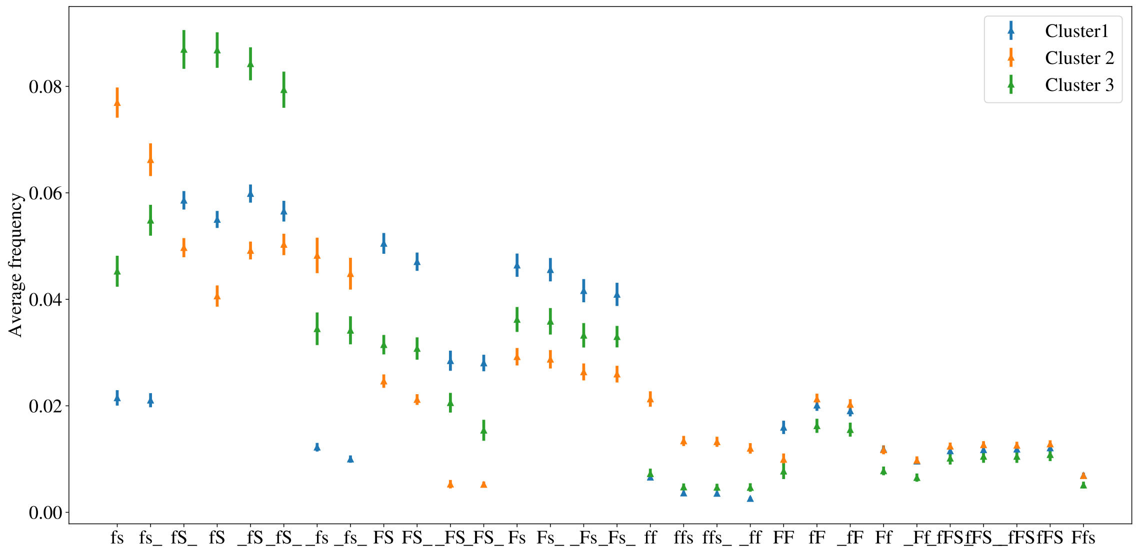


Figure 4.2: Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Psychology.

with the extracted patterns. We should mention that these labels are provided to distinguish the discovered clusters, rather than exactly describing student behaviors in them.

OLI Psychology; Figure 4.2 shows these top 30 frequent patterns and their average frequencies in each cluster for OLI Psychology. In cluster 1, patterns such as 'FS', 'FS_', 'Fs' and 'Fs_' are significantly more frequent than the other two clusters. The common behavior in this cluster is having long failures and then a success. In cluster 2, frequent patterns are 'fs', 'fs_', '_fs', '_fs_'. These students tend to spend a short time on solving problems and have a short failure after a short success. In cluster 3, patterns that are more frequent are 'fS_', 'fS', '_fS' and '_fS_'. Having a long successful attempt after a short failure is the common behavior in students in this cluster. We can recognize 3 types of different behaviors with the extracted patterns, however the behaviors in these cluster differ from previous dataset. So in each dataset, the pattern clusters are specific for that course and the same set of patterns are not essentially discovered in all datasets.

OLI Statistics; Figure 4.3 shows top 30 frequent patterns and their average frequencies in each cluster for OLI Statistics. In cluster 1, patterns such 'ff', 'fs', 'FF' and 'sf' are significantly more frequent than the other two clusters. The common behavior in this cluster is having failures(usually short). In cluster 2, frequent patterns are 'SS', 'sss' and 'SS_'. These students tend to have consecutive successful attempts. This cluster is analogous to "confirmers" as they successfully repeat their attempts. In cluster 3, patterns that are more frequent are 'sS', '_sS', 'sS_' and

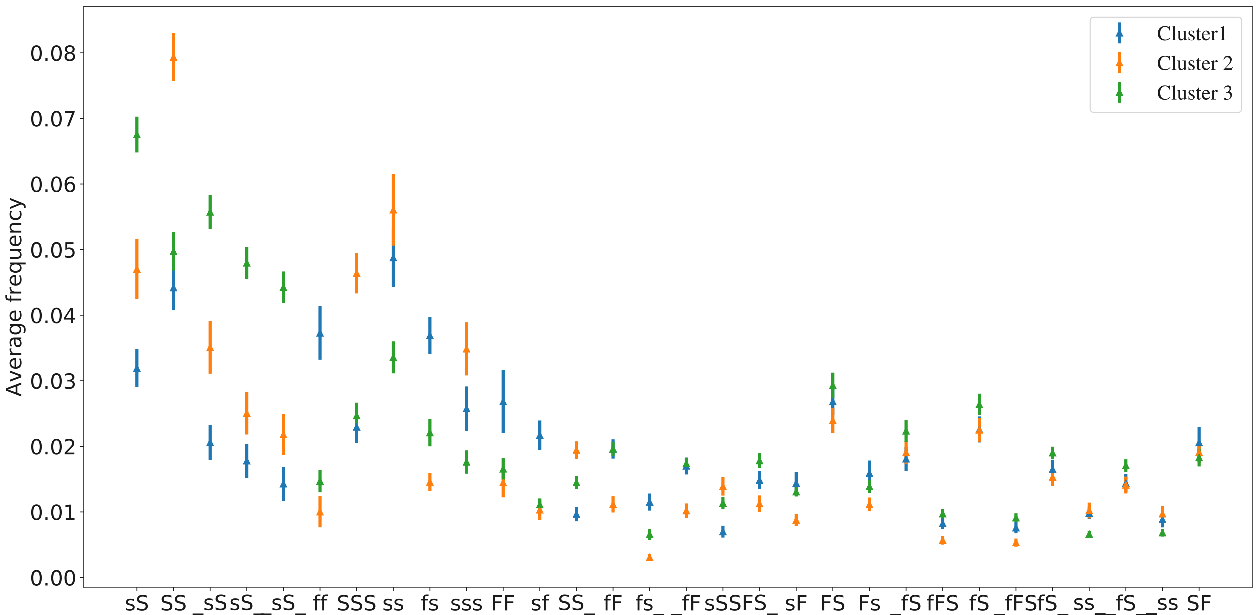


Figure 4.3: Top 30 patterns and their frequencies in 3 clusters. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Statistics.

'_sS'. A combination of short and long successful attempts are seen frequently in this cluster. They first answer a problem rapidly and right after they spend more time on the same problem.

The results show that students in each cluster have similar behaviors. However, the clusters such as "confirmers" or "thinkers" that we found in Mastery Grids are not in two other datasets. It implies that clustering patterns is meaningful with our approach, but the clusters are all inclusive and are dataset dependant.

4.4.2 Evaluation

In [56], Martinez et. al. proposed a method to compress the sequences. In that work, the regular expressions are used to compact the sequential attempts of the same type. So the quantifier "+" shows the repetition of an activity. We compare this method with ours and find the average frequency of the patterns with generalisation. The Figures 4.4 to 4.6 shows the results. The results show that using generalisation, information is lost and the patterns such as "s+" does not reflect how many times the attempts is repeated. As a result, for example in Mastery Grids "Ss" and "+S" are in different clusters while they represent a similar behavior.

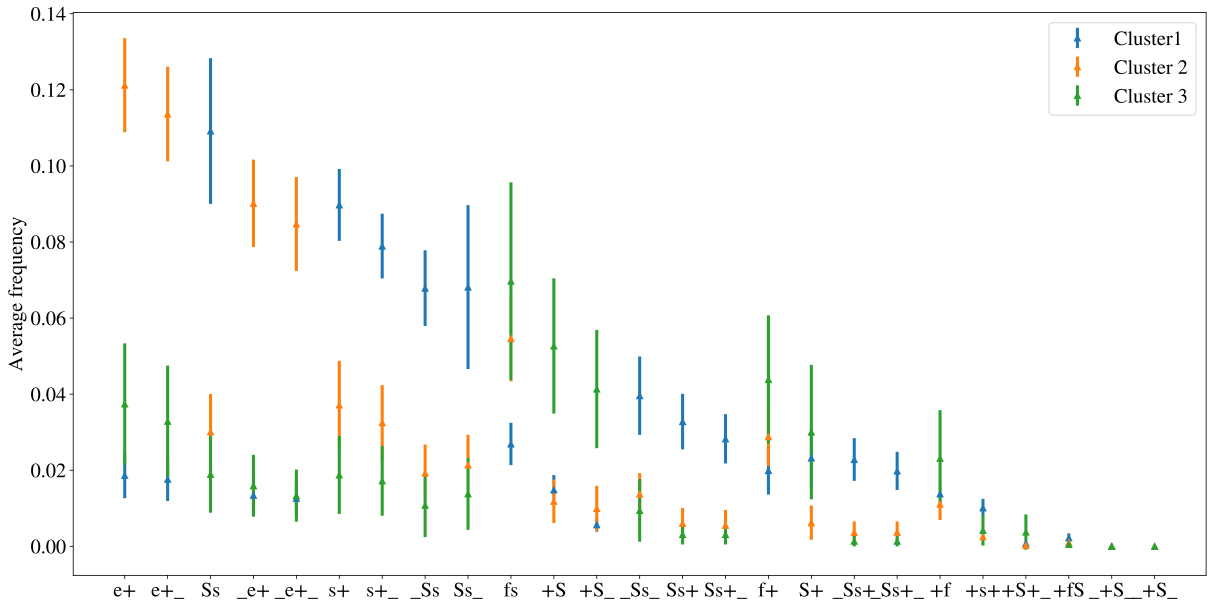


Figure 4.4: Top 30 patterns and their frequencies in 3 clusters using generalisation. Patterns are ordered by the maximum difference of frequencies between three clusters in Mastery Grids.

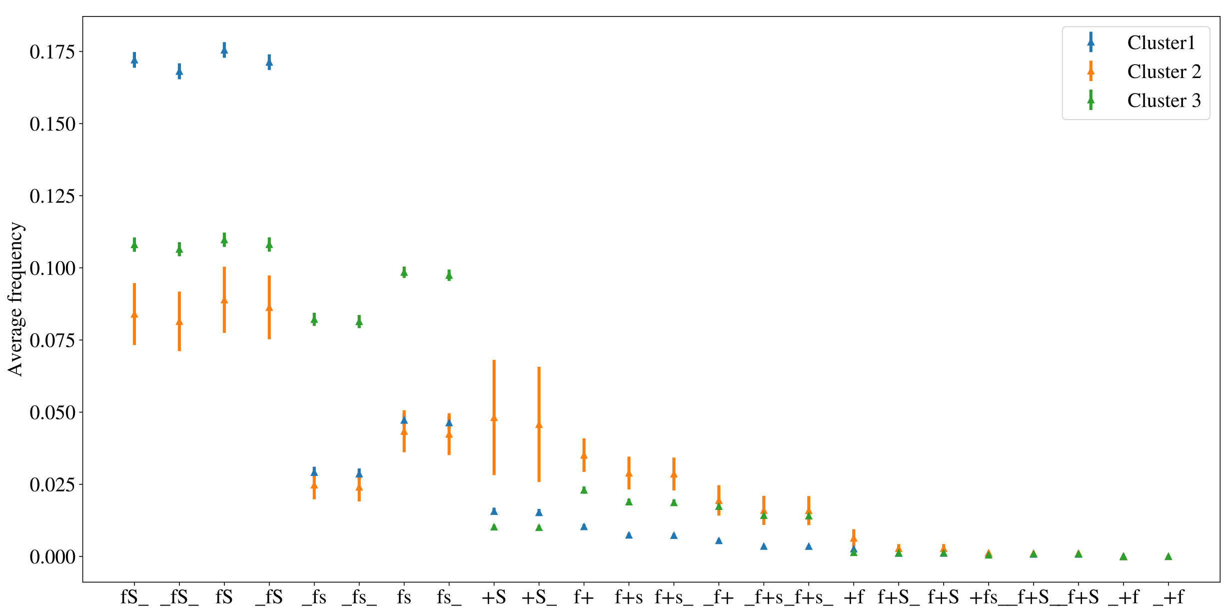


Figure 4.5: Top 30 patterns and their frequencies in 3 clusters using generalisation. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Psychology.

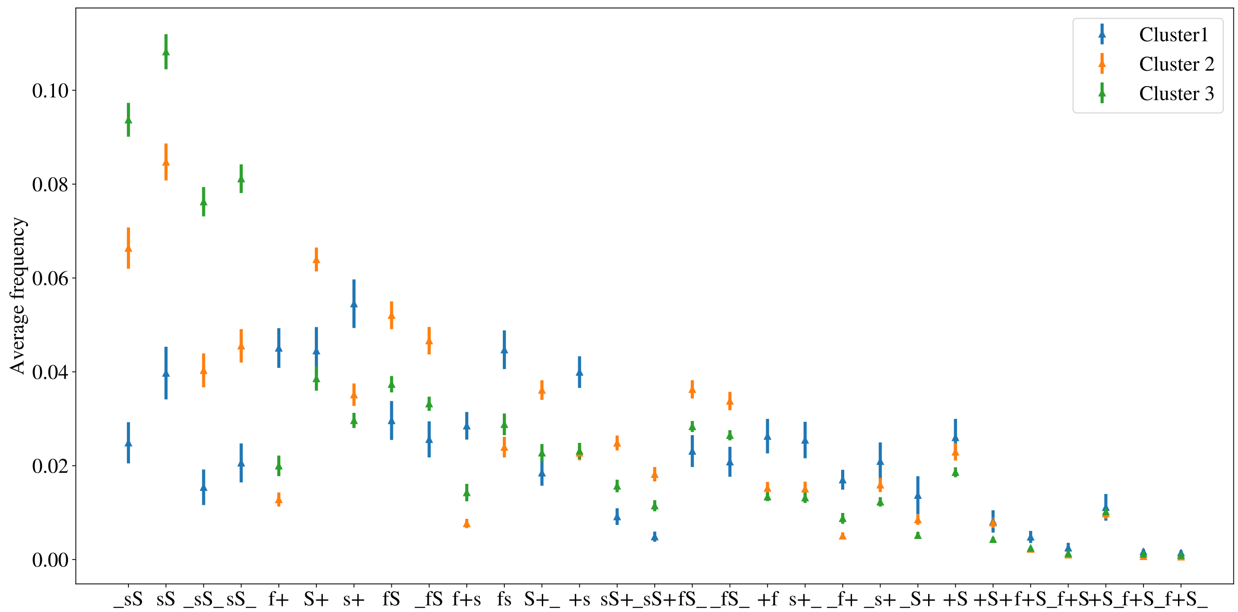


Figure 4.6: Top 30 patterns and their frequencies in 3 clusters using generalisation. Patterns are ordered by the maximum difference of frequencies between three clusters in OLI Statistics.

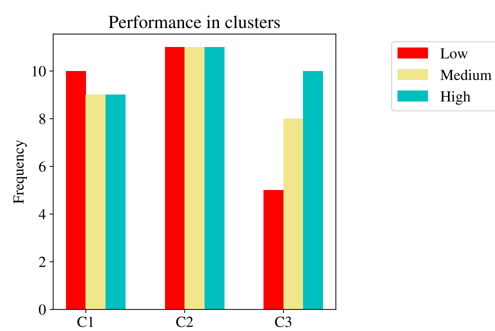


Figure 4.7: High, medium, and low-performance (based on pre-test) student frequencies in three clusters generated based on student patterns in Mastery Grids.

4.4.3 Performance Analysis

Here we examine the clusters to detect whether we can associate the macro-behavioral patterns represented by each cluster with students' learning performance, measured by normalized learning gain. Figures 4.7 to 4.9 show the number of students with low, medium and high performance (pre-test, post-test and learning gain) in each cluster for Mastery Grids dataset.

As we can see in the figures, the clusters do not show a significant difference in the number of high, medium, or low performance students. For example in cluster 2 the number of low-, medium- and high-performance students based on pre-test are exactly the same. The similar conclusion holds for pre-test and post-test performance of students. We can conclude that the

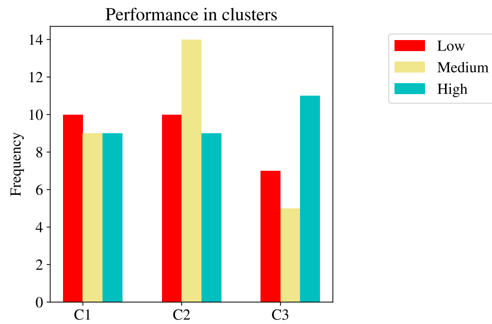


Figure 4.8: High, medium, and low-performance (based on post-test) student frequencies in three clusters generated based on student patterns in Mastery Grids.

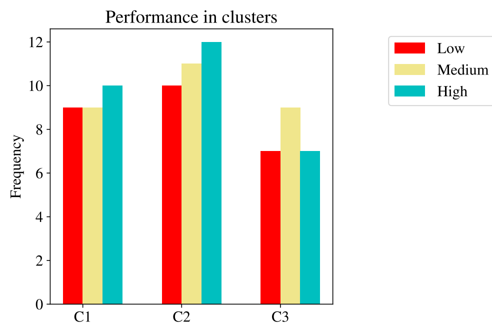


Figure 4.9: High, medium, and low-performance (based on learning gain) student frequencies in three clusters generated based on student patterns in Mastery Grids.

macro-patterns represented by the clusters are neither related to students' past performance nor to their course-level performance. In other words, the patterns do not separate weak students from strong ones. Instead, they represent students' different approaches to work with learning content. Within the group of students using the same approach, however, we can find both strong and week students. The results are similar to the observation in the original paper [28].

Figures 4.10 to 4.12 show the number of students with low, medium and high performance (pre-test, post-test and learning gain) in each cluster in OLI Psychology dataset. There is not a significant difference in the number of high, medium, or low performance students.

Figure 4.13 shows the number of students with low, medium and high performance (post-test) in each cluster in OLI Statistics dataset. The number of high performance students in cluster 2 and 3 are higher than cluster 1. These clusters have patterns with more successful attempts.

Next, we study the differences in behavioral micro-patterns of high and low-performance students within each cluster. By this, we hope to uncover the efficient and inefficient micro-patterns

Figure 4.10: High, medium, and low-performance (based on pre-test) student frequencies in three clusters generated based on student patterns in OLI Psychology.

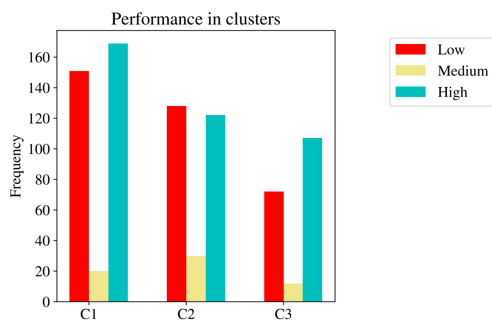


Figure 4.11: High, medium, and low-performance (based on post-test) student frequencies in three clusters generated based on student patterns in OLI Psychology.

that happens within students with the same studying traits. To achieve this, we examine the average frequencies of micro-patterns for low and high performance students in each cluster and select the ones with a significant difference.

Mastery Grids; The results are shown in Figures 4.14 to 4.16. As presented in Figure 4.14, in cluster 1 ("confirmers"), patterns such as 'fssss' and 'eE' are found to be significantly more in high performance students. On the other hand, patterns 'Fss_', 'Fs_', and '_ss' appear more in low performance students. According to this, we can conclude that, the "confirmer" group students do repeat their success. But their approach to this repetition determines their performance in the course: 1) repeat after an initial success ('_ss') is associated with weaker students; 2) more repetition after an initial failure ('fssss') is associated with stronger students, as short repetitions and quitting after failure ('Fss_' and 'Fs_') is associated with weaker students; and 3) repeat reading examples is associated with stronger students.

We can see that in cluster 2 ("thinkers", Figure 4.15), high performance students have patterns such as '_FF', 'FF', and 'Sss', while low performance ones have higher rate of patterns with short failure ('f') in them. This shows that high-performance "thinkers" think each time they try a

Figure 4.12: High, medium, and low-performance (based on learning gain) student frequencies in three clusters generated based on student patterns in OLI Psychology.

Figure 4.13: High, medium, and low-performance (based on post-test) student frequencies in three clusters generated based on student patterns in OLI Statistics.

problem, until it is sufficiently understood. In contrast, weaker students frequently try to guess and fail in solving problems. Interestingly, low-performance "thinkers" also have a high frequency of 'Fff' pattern. It can be concluded that they start with serious intentions, but then start to guess the answers.

For the "reader" students (cluster 3, Figure 4.16), we see longer attempts (e.g., 'EE', '_FS_', and 'FS') for high-performance students, compared to shorter attempts (e.g., 'ffs' and 'Fs') for low-performance ones. We can see that 1) high-performance students work with examples more carefully; 2) they do not rush after failure, but think and most always get it right; and 3) in contrast, low-performance students do not spend enough time on their attempts, whether it is a success or failure. In general, having patterns that include long attempts among high performance students and short attempts in low-performance ones demonstrate the impact of spending time on the performance.

OLI Psychology; In this dataset, cluster 1 shows that high-performance students use '_FS', '_FS_', 'FS' and 'FS_' more often. These patterns consist of long attempts either success or failure. The patterns that are more associated with the low-performing students have at least one short

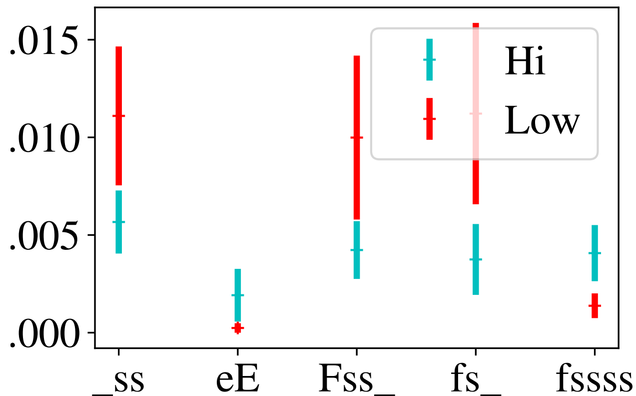


Figure 4.14: Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 1 in Mastery Grids

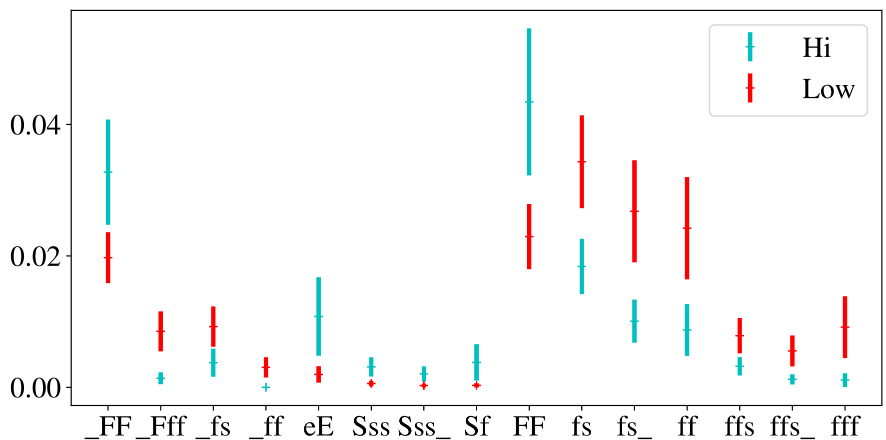


Figure 4.15: Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 2 in Mastery Grids

attempts such as 'Fs', 'Fs_', 'fF', 'ff' and 'ffs' In cluster 2, low-performance students use 'ff', 'ffs' and 'ffs_' more often. The patterns that contain longer attempts are used more frequently in these clusters. These results indicate that amount of time spent on activities has an association with students' performance. The results are shown in Figure 4.17 to 4.19.

OLI Statistics; In this dataset, cluster 1 shows that high-performance students use patterns such as '_sS_', 'fs', 'ss' and 'sss' more often and low-performance students use 'FS', 'SF' and 'FF'. Long attempts of the low-performers show that they struggle in solving problems and lack of knowledge could be the reason of it. In cluster 2, low-performance students use 'fS', 'FS', '_fS', 'fS_' and '_fS_' more often. These patterns start with a failure and then a success and the failures are usually short. The students rapidly try to solve a problem and make mistakes. High-performance students in cluster 3 use 'SS', 'ss', 'SS_', 'SSS' and 'sss'. The results are shown in Figure 4.20 to 4.22.

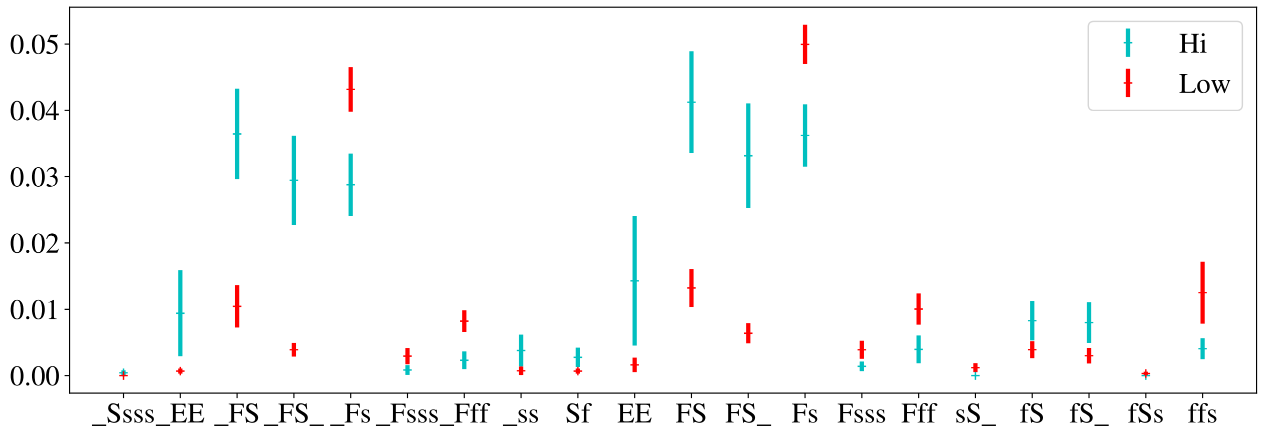


Figure 4.16: Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 3 in Mastery Grids

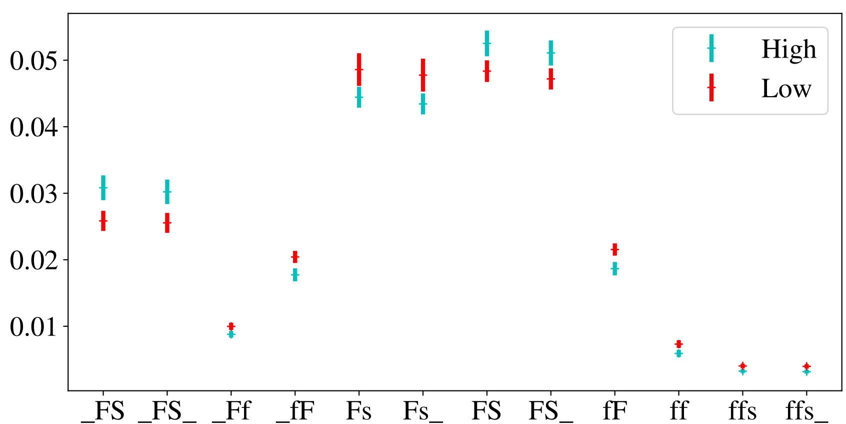


Figure 4.17: Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 1 in OLI Psychology

The performance analysis shows that within recognized clusters, the patterns are associated with students' performance in each cluster. It means that although there is not a direct relation between patterns and students' performance but to some extent the relation exists that needs to be investigated more.

4.5 Conclusion

In this chapter, we answered the first research question by analyzing students' behavior patterns while working with parameterized exercises and annotated examples. Using frequent pattern mining, we discovered frequent *micro-patterns* of student behavior and used them to construct *macro-pattern* behavior vectors for students. Using data driven approaches, we analyzed the sta-

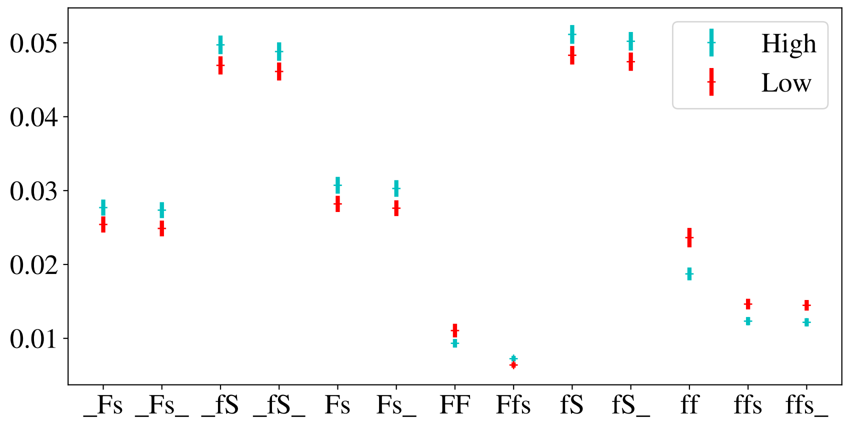


Figure 4.18: Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 2 in OLI Psychology

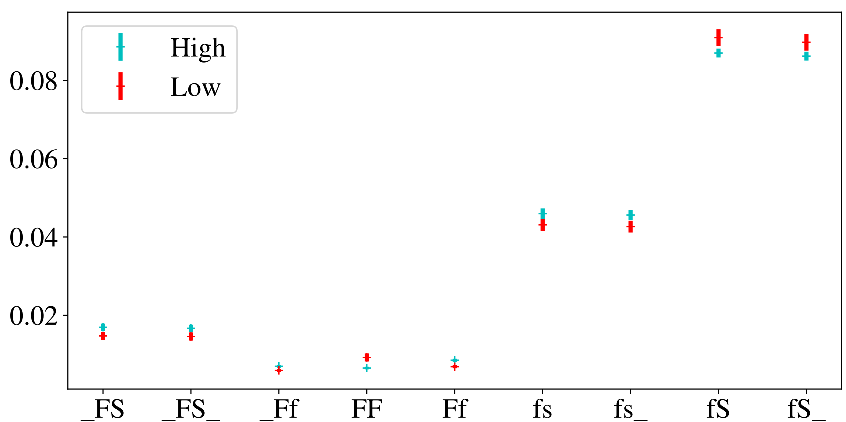


Figure 4.19: Patterns with significant difference of frequency for low performance and high performance (learning gain) students in Cluster 3 in OLI Psychology

bility of these macro-patterns and showed that these are results of students' behavioral traits. Clustering students according to these macro-patterns, we discovered three groups of students, which we nicknamed as "confirmers", "thinkers", and "readers". Among these groups, we identified students' efficient and inefficient micro-patterns by comparing frequent patterns of high and low-performing students. Our results suggested that for "confirmer" students, it is beneficial to encourage repetitions after they fail in solving a problem. But, repetitions after success is redundant and inefficient. For "thinkers", it is useful to encourage them to continue to think deeper each problem, even after failure. For "readers", working more carefully with examples and spending more time to think is beneficial. Being able to discover a few behavioral clusters that represent different ways of

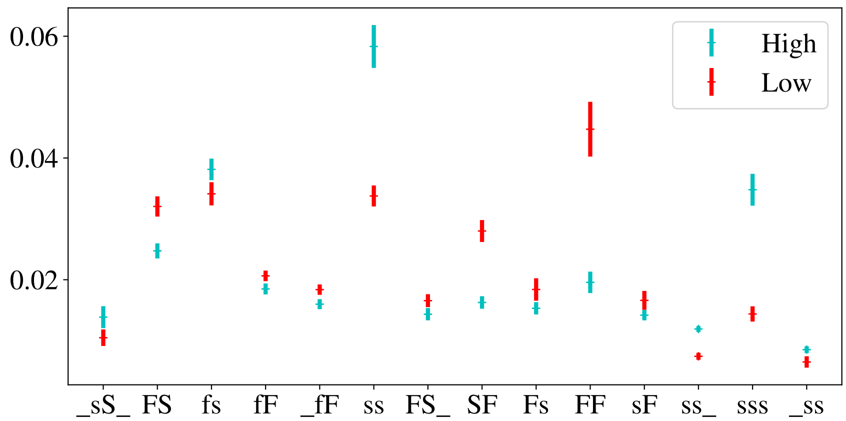


Figure 4.20: Patterns with significant difference of frequency for low performance and high performance (post-test score) students in Cluster 1 in OLI Statistics

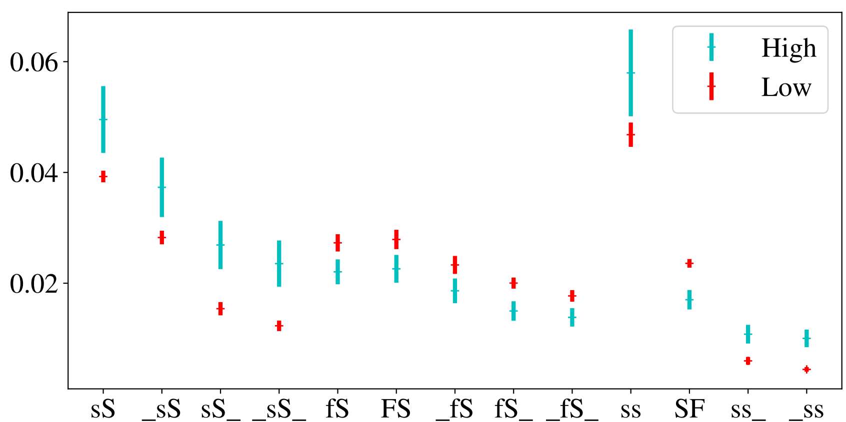


Figure 4.21: Patterns with significant difference of frequency for low performance and high performance (post-test score) students in Cluster 2 in OLI Statistics

learning is a promising step towards personalization: if learning behavior diversity among students is not that large, we can nudge different student groups towards the optimal behavior in different ways. In the future, these results can be extended to be used as encouragement or recommendation to help students of each group to take on more efficient behaviors.

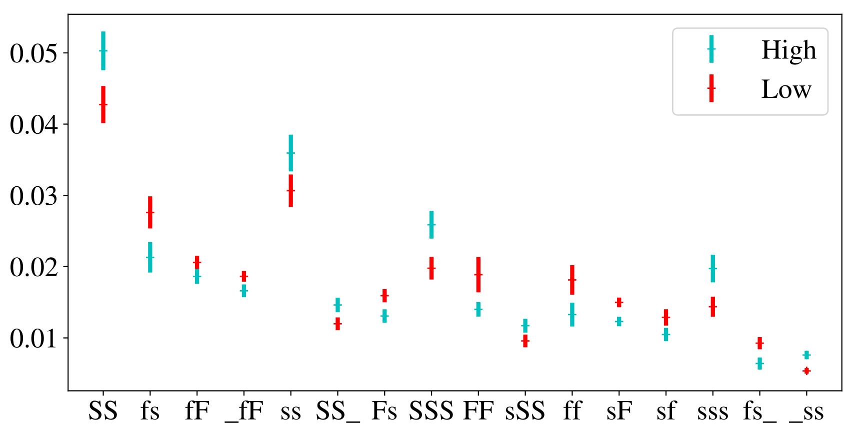


Figure 4.22: Patterns with significant difference of frequency for low performance and high performance (post-test score) students in Cluster 3 in OLI Statistics

CHAPTER 5

Detecting Trait vs. Performance Student Behavioral Patterns Using Discriminative Non-negative Matrix Factorization

5.1 Introduction

Recent studies on extracting behavioral patterns from students' activities sequences have shown that students follow stable behavioral patterns while working with these systems [28, 57, 26, 89]. In addition to learning patterns, some studies have discovered inefficient learning behaviors in student sequences. For example in previous chapter, we found that some students tend to repeat the same problems and concepts, even after mastering them, rather than moving on to learn new and more complex concepts. One may expect to see an association between these inefficient learning behaviors and low performance in students. However, the same studies showed that using all behavioral patterns, one cannot easily separate high- and low-performing students. Studying stability of these patterns during the time, suggested that many of them are representative of student behavioral traits, rather than student performance. Specifically, both high- and low-learners may demonstrate some inefficient behavioral patterns in their sequences.

In this context, a natural question is if we can differentiate between the trait behavioral patterns and the performance behavioral patterns. In other words, which of the behavioral patterns are associated with student behavioral traits, and which are indicators of students' high or low performance? Answering these questions will help to better detect inefficiencies in students' sequences while interacting with online learning systems, and guide them towards a more productive learning behavior. In this chapter, we mine the trait versus performance behavioral patterns in students by grouping the students according to their performance and discovering the latent factors that represent each group using discriminative non-negative matrix factorization. We experiment on three real-word dataset of sequences from students interacting with online learning systems, with different learning material types: problems, worked-examples and hints. Our experiments show the discriminative power of our method between different types of behavioral patterns. Also, by clustering these patterns according to their discovered latent factors, we reveal interesting associations between them. The goal in this chapter is to answer the second research question in 1.2:

Could we detect behavioral traits from performance traits and find behavioral patterns that are common and specific for students with different performance?

5.2 Discriminative Learning of Student Behavior

In this section, we describe how pattern vectors from previous chapters are used to model students' behavior. In summary, our framework follows the following steps:

1. Building student-pattern matrices.
2. Finding discriminative vs. common patterns between high- and low-performing students.

In section 4.2, we calculated the normalized frequency of each micro-pattern in their complete coded sequence for each student. Here we build a pattern-student matrix that represents all student behaviors by concatenating their normalized micro-pattern vectors. This matrix represents the weight of each pattern in students' pattern vectors.

Our main goal in this chapter is to distinguish between micro-patterns that can represent students' learning behavior traits and the ones that can be indicators of student performance. To measure the performance of student s, we use students' normalized learning gain. Our assumption is that the micro-patterns that are representative of learning behavior traits, are independent of student performances. As a result, they can be shared across both high- and low-performance students. On the other hand, we assume that the micro-patterns that discriminate high-performing students from the low-performing ones, can be predominantly seen in one of these two groups. According to these assumptions, we expect to see three sets of micro-patterns in high- and low-performance students' pattern vectors:

- Group i: A set that is common across the student groups, and has a similar importance in both groups' pattern vectors.
- Group ii: A set that is frequently seen in high-performance students' sequences, and not in low-performance ones'.
- Group iii: A a set that is specific to low-performance students.

To verify this distinction between different sets of patterns, we apply discriminative non-negative matrix factorization [39] that was proposed for discriminatory topic modeling in documents. To do this, we split the student-pattern matrix X, based on the students' performance to achieve matrix X_1 for low-performing students, and X_2 for low-performing ones and transpose them to have pattern-student matrices. Each column in these matrices represent micro-patterns of one student, and each row represent the presence of one micro-pattern in all students' sequences.

Using simple non-negative matrix factorization, each of these two matrices can be decomposed into multiplication of two lower-dimensional matrices W and H, with k latent factors. These latent factors can summarize the association between behavioral micro-patterns and students, using a shared latent space ($X_1 \approx W_1 H_1^T \quad X_2 \approx W_2 H_2^T$). To learn the W and H matrices, an optimization algorithm (such as gradient descent) can be used to minimize the following objective function, with respect to these parameters:

$$L = \left\| X_1 - W_1 H_1^T \right\|_F^2 + \left\| X_2 - W_2 H_2^T \right\|_F^2 \tag{5.1}$$

However, this factorization does not discriminate between common and distinctive patterns. To enforce our assumptions and further group the micro-patterns into the above-mentioned three sets, we use their latent representations. To find the micro-patterns that belong to group i, we restrict the discovered latent representations for some of the micro-patterns to be as similar as possible across the two groups of students. To find the micro-patterns that belong to groups ii and iii, we impose the discovered latent representations for other micro-patterns to be as different as possible across the two groups of students. To do so, we assume W and H can be split to two sub-matrices, each having either common or discriminative patterns, with k_c and k_d latent factors, respectively:

$$\begin{gathered} W_1 = [W_{1,c} \quad W_{1,d}], \quad W_2 = [W_{2,c} \quad W_{2,d}] \\ H_1 = \begin{bmatrix} H_{1,c} \\ H_{1,d} \end{bmatrix} \quad H_2 = \begin{bmatrix} H_{2,c} \\ H_{2,d} \end{bmatrix} \end{gathered} \tag{5.2}$$

Here $W_{1,c}$ and $W_{2,c}$ contain common patterns and $W_{1,d}$ and $W_{2,d}$ have distinct ones and $k = k_c + k_d$. To impose the similarity between common patterns (setting $W_{1,c} \approx W_{2,c}$) and dissimilarity between distinct patterns (setting $W_{1,d} \not\approx W_{2,d}$), we add two regularization terms, $f_c(.)$ and $f_d(.)$, to the objective function. $f_c(.)$ and $f_d(.)$ aim to penalize the difference between common patterns and the similarity between distinct patterns, respectively. For the difference between common patterns, the euclidean distance is used and for the similarity between distinct ones, the dot product between vectors. As a result, these two functions are defined as in Equation (5.3).

$$
\begin{aligned}
f_c(W_{1,c}, W_{2,c}) &= \left\| W_{1,c} - W_{2,c} \right\|_F^2 \\
f_d(W_{1,d}, W_{2,d}) &= \left\| W_{1,d}^T W_{2,d} \right\|_F^2
\end{aligned}
\tag{5.3}
$$

Eventually, considering regularization on W and H for generalizability purposes, we will minimize the objective function in Equation (5.4), with respect to W and H, and constraining them to be non-negative, using Gradient Descent (GD) algorithm. An illustration of our framework is presented in Figure 5.1.

Non-negative Matrix Factorization is appropriate for tasks that underlying factors could be interpreted as non-negative. In this decomposition, the components are meaningful. In the matrix that we decompose, each column represents a pattern and each element is the normalized frequency of the pattern. The NMF decomposes the matrix to components that could be considered as behaviors. These low dimensional matrices represents students' behavior. So the interpretation while having negative weight for a behavior is not possible.

$$
\begin{aligned}
L = \quad & \left\| X_1 - W_1 H_1^T \right\|_F^2 + \left\| X_2 - W_2 H_2^T \right\|_F^2 + \\
\alpha \left\| W_{1,c} - W_{2,c} \right\|_F^2 &+ \beta \left\| W_{1,d}^T W_{2,d} \right\|_F^2 + \gamma(\|W\|^2 + \|H\|^2)
\end{aligned}
\tag{5.4}
$$

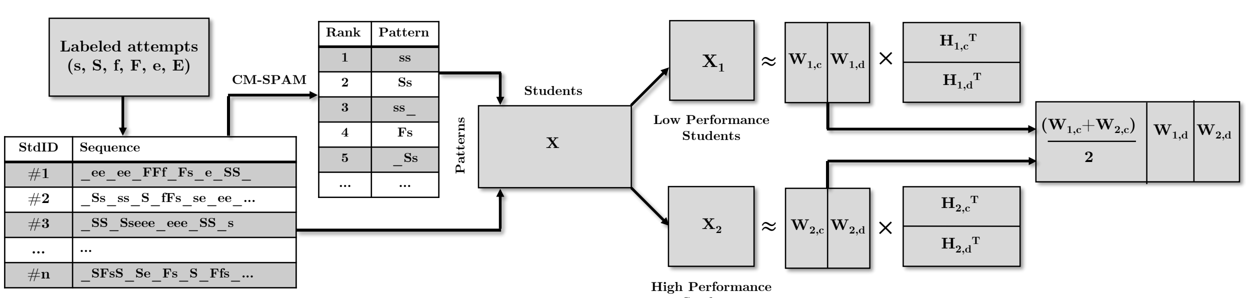


Figure 5.1: Most frequent patterns are extracted from sequences by CM-SPAM. These patterns are rows of matrix X and students are columns. We split matrix X based on the performance of the students to X_1 and X_2. Then with discriminative non-negative matrix factorization, common and distinct latent factors are extracted.

5.3 Experiments

5.3.1 Hyper-parameter Tuning

Using the GD algorithm and performing a grid-search to find the best number of common and distinct latent factors (K_c and K_d), we find each pattern's latent vectors. To evaluate the goodness of fit, we use the reconstruction error (Root Mean Square Error) on matrices X_1 and X_2. We vary K between 2 and 20 and for each K, we search over K_cs between 0 to K, such that $K_d = K - K_c$. The least reconstruction error happens when:

- **Mastery Grids:** $K = 15$, $K_c = 10$, and $K_d = 5$
- **OLI Psychology:** $K = 12$, $K_c = 3$, and $K_d = 9$
- **OLI Statistics:** $K = 13$, $K_c = 4$, and $K_d = 9$

5.3.2 Clustering Patterns

To discern the students' learning behavior trait and performance patterns, we cluster these patterns and peruse the latent factors that fall in each cluster. To do this, we build matrix $W_{c,d}$ from decomposed matrices $W_{1,c}$, $W_{2,c}$, $W_{1,d}$ and $W_{2,d}$. First we take the average of common latent factors in $W_{1,c}$ and $W_{2,c}$ to build a new matrix $W_{(1,2)c}$ and then construct matrix $W_{c,d}$ from concatenating $W_{(1,2)c}$, $W_{1,d}$ and $W_{2,d}$ as shown in Figure 5.1. We cluster the rows of the matrix $W_{c,d}$ according to the discovered latent factors, into different groups using the spectral clustering algorithm. We choose 6 clusters because the patterns are well separated in the clusters. The results of the clustering and latent discovery are in the following:

Mastery Grids; Figure 5.2 demonstrates the matrix $W_{c,d}$ arranged into the detected clusters. In this figure, the horizontal bars are dividing the patterns into the discovered clusters. The left 10 columns show an average of common latent factors in $W_{1,c}$ and $W_{2,c}$, the middle 5 are discriminative latent factors for low-performing students ($W_{1,d}$), and the last 5 are factors of high-performing students ($W_{2,d}$). The darker the color, the more a latent factor is weighted for each pattern. The results illustrate the division of patterns based on a combination of learning trait and performance factors. First, we see that trait vs. performance patterns are falling into separate clusters. For example, patterns containing reading examples (as a trait) such as 'ee_', and '_ee' fall into the same

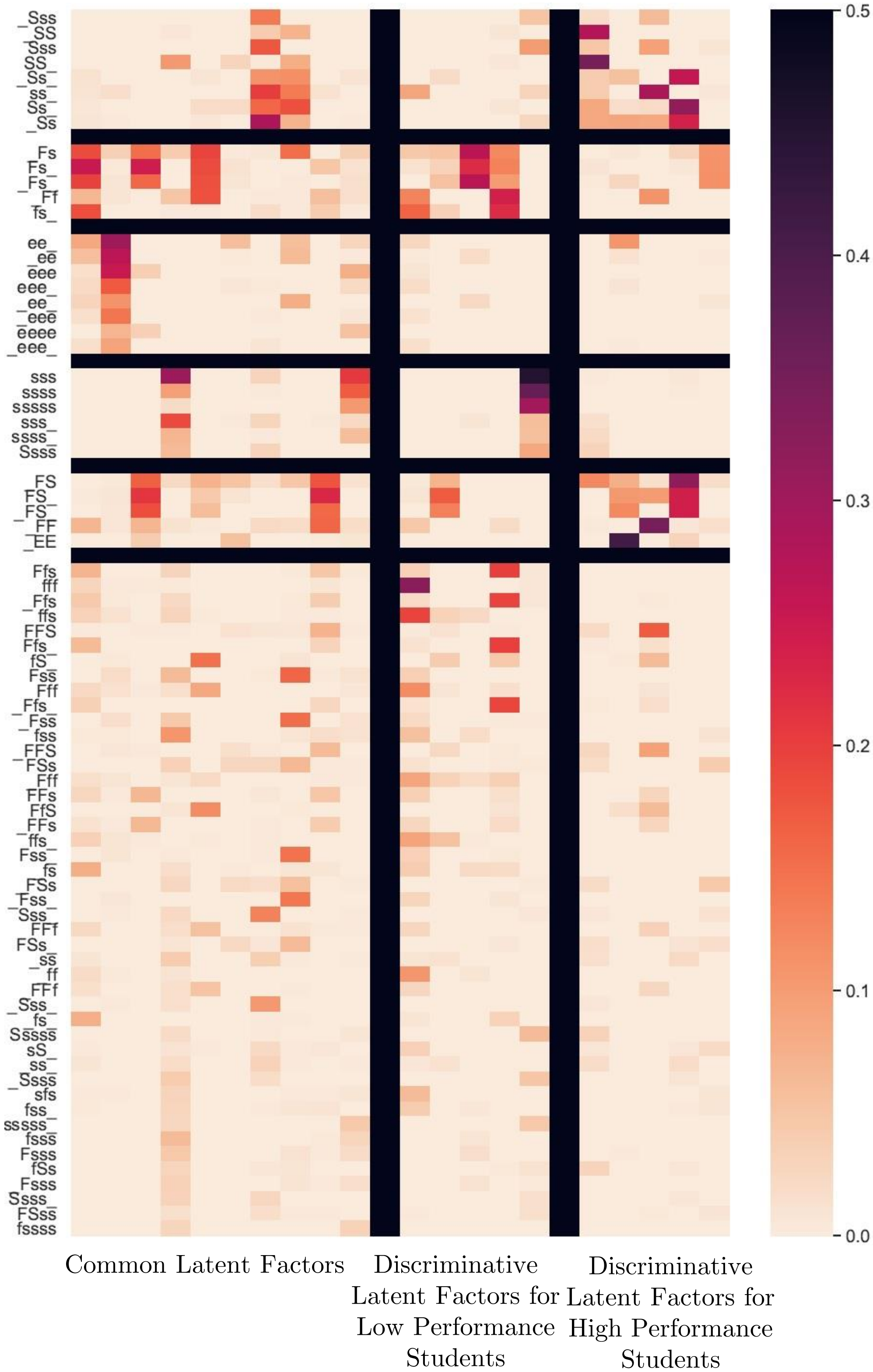


Figure 5.2: Heatmap shows the distribution of latent factors in common and discriminative parts for Mastery Grids.

cluster, and patterns with long successes (as a performance indicator) such as '_Sss', and '_SS' are together in another cluster. Second, we see that high- and low-performance patterns are falling into separate clusters. For example, patterns with long successes (as a high-performance indicator) such as '_SS' vs. patterns with short repetitive successes (as a low-performance indicator), such as 'sssss', belong to different clusters. Third, we observe a trait-related separation between different performance clusters. For example, both first group of patterns (with long successes, followed by a few short successes) and the fifth group of patterns (with long failures, mostly followed by long success) are indicators of high-performance students. However, the first one shows the students that would like to repeat their success a few times after spending the time to get a problem right. While the second one shows the group of students who will move on to other problem topics as soon as they have a long-thought success, after a long failure. This result is in accordance with grouping the students into "confirmers" and "non-confirmers" by Guerra et al. [28]. We see similar trait-based clusters within low-performance patterns: the second and fourth sets of patterns in Figure 5.2.

Looking at the heatmap, we can see that a big group of micro-patterns in the bottom rows have similar, and lower weights in common and distinctive latent factors. These are the patterns that happen in student sequences from any groups (so, associated with learning behavior trait), but are not very strong in showing the kind of learning trait. Another group of patterns that are common between students is the ones that show predominantly example-related activities (e.g., 'ee_', and '_ee' micro-patterns). For these patterns, we see lower discriminatory weights for the performance latent factors, but high weights for the common latent factors. This shows that not only these patterns are indicative of learning behavior traits, but they are also representing a specific kind of these traits: they show the group of students who are interested in studying the worked examples, more than others. This finding is in accordance with having "readers" vs. other student clusters in [57].

The rest of the patterns are performance patterns: if they have a high weight in low-performing latent factors, they will not have a high weight in high-performing latent factors, and vice versa. For example, the first group of patterns, mostly with long successful attempts repeated only once or twice with shorter successes, are having higher weights in high-performance factors, and very low weight in low-performance factors. This means that observing these sets of patterns in a student's behavior can be indicative of their high performance. On the other hand, the sets of patterns with many repeated successful, but short attempts, (like ‘sss’, and ‘ssssss’) are having high weights in low-performance factors and almost zero weights in high-performance factors. It means students that succeed in solving problems of the same topic repeatedly but do not take the time on them are more likely to be low-performing students.

OLI Psychology; The heatmap after clustering the patterns and discovering latent factors are shown in Figure 5.3. The left 3 columns show an average of common latent factors in $W_{1,c}$ and $W_{2,c}$, the middle 9 are discriminative latent factors for low-performing students ($W_{1,d}$), and the last 9 are factors of high-performing students ($W_{2,d}$). We observe that similar patterns fall into separate clusters. For example patterns such as ’fS’,’fS_’,’_fS’,’_fS_’ are in the first cluster. The patterns in this cluster resemble such that starts with a short failure and is followed by a long success. The patterns such as ’Fs’, ’Fs_’, ’_Fs’, ’Ffs_’ and ’Ffs’ are in the second cluster. These patterns are unlike the former patterns and Have a short success after a long failure. The other cluster that has such similarities is the fourth cluster. This cluster contains ’fs’, ’_fs’, ’fs_’ and ’_fs_’. The same order as in other mentioned clusters but with short attempts. In each of these clusters, similar traits are seen so the clustering is meaningful.

We can see that the fourth cluster of patterns has higher weights in low-performance factors. The patterns in this cluster are a short success after a short failure. It means that solving problems quickly without spending enough time is associated with low-performance students. And the second group of patterns has higher weights in high-performance factors. The patterns in this cluster are mainly showing success after a long failure. So spending enough time on the problems that students solve incorrectly, is specific for high-performance students. Low-performance students use this pattern barely and do not spend enough time on the problems that solve wrongly.

A big group of patterns falls into the third cluster. The corresponding latent factors in this cluster have lower weights in distinctive latent factors. Such patterns are seen in students’ sequences with low- and high-performances so are recognized as learning traits. The patterns that include using hints are in this cluster. So using hint is a learning trait and not a performance trait. Another cluster that is showing behavior traits is the first cluster that has high weights for the common latent factors. This finding shows that having a long success after a short failure is a learning trait and is not related to students’ performance.

OLI Statistics; The heatmap after clustering the patterns and discovering latent factors are shown in Figure 5.4. The left 4 columns show an average of common latent factors in $W_{1,c}$ and

$W_{2,c}$, the middle 9 are discriminative latent factors for low-performing students ($W_{1,d}$), and the last 9 are factors of high-performing students ($W_{2,d}$). Clusters 1, 2, 3 and 5 have higher weights in low-performance latent factors and lower weights in high-performance latent factors. On the other hand, the fourth cluster has higher weights in high-performance and lower weight in low-performance latent factors. This cluster has patterns with long successful attempts. It shows that spending more time on solving problems of the same topic is associated with high-performance students. We can discover that there are discriminative pattern clusters in this dataset, however, the patterns in each cluster are not very similar and are mixed up. So finding a trait specific for each cluster is not obvious.

5.3.3 Latent Factors Analysis

To analyze the clusters more and find the most discriminating patterns within each cluster, we find the average of the latent factor values in each cluster. These results are plotted in Figures 5.5, 5.6 and 5.7. The error bars represent the 95% confidence interval, showing if the weight of a latent factor in a cluster is significantly different from the weight of the same latent factor in other clusters.

Mastery Grids; We observe that the second common latent factor is the most prominent in cluster 3 (the example studying patterns). Cluster 4's (low-performance patterns indicating a short success after a long failure) most prominent latent factor is the fourth low-discriminative factor; and cluster 2's (low performing sequence of repeated short successes) most weighted factor is the last factor in the discriminative ones. These results show the discriminative power of latent factors, especially in indicating "example studying" behavior and finding low-performing patterns.

OLI Psychology; The third and fourth latent factors in low-dicriminative factors are the most significant in cluster 4: The patterns that include short success after a short failure. In cluster 2, the forth and eight latent factors in high-discriminative factors have significantly more weights than other latent factors: The patterns with a short success after a long failure.

OLI Statistics; In this dataset, recognizing latent factors that belong exclusively for one of the 3 parts is not apparent. As it is shown in Figure 5.7, the only latent factors that are significantly different from others are in cluster 6.

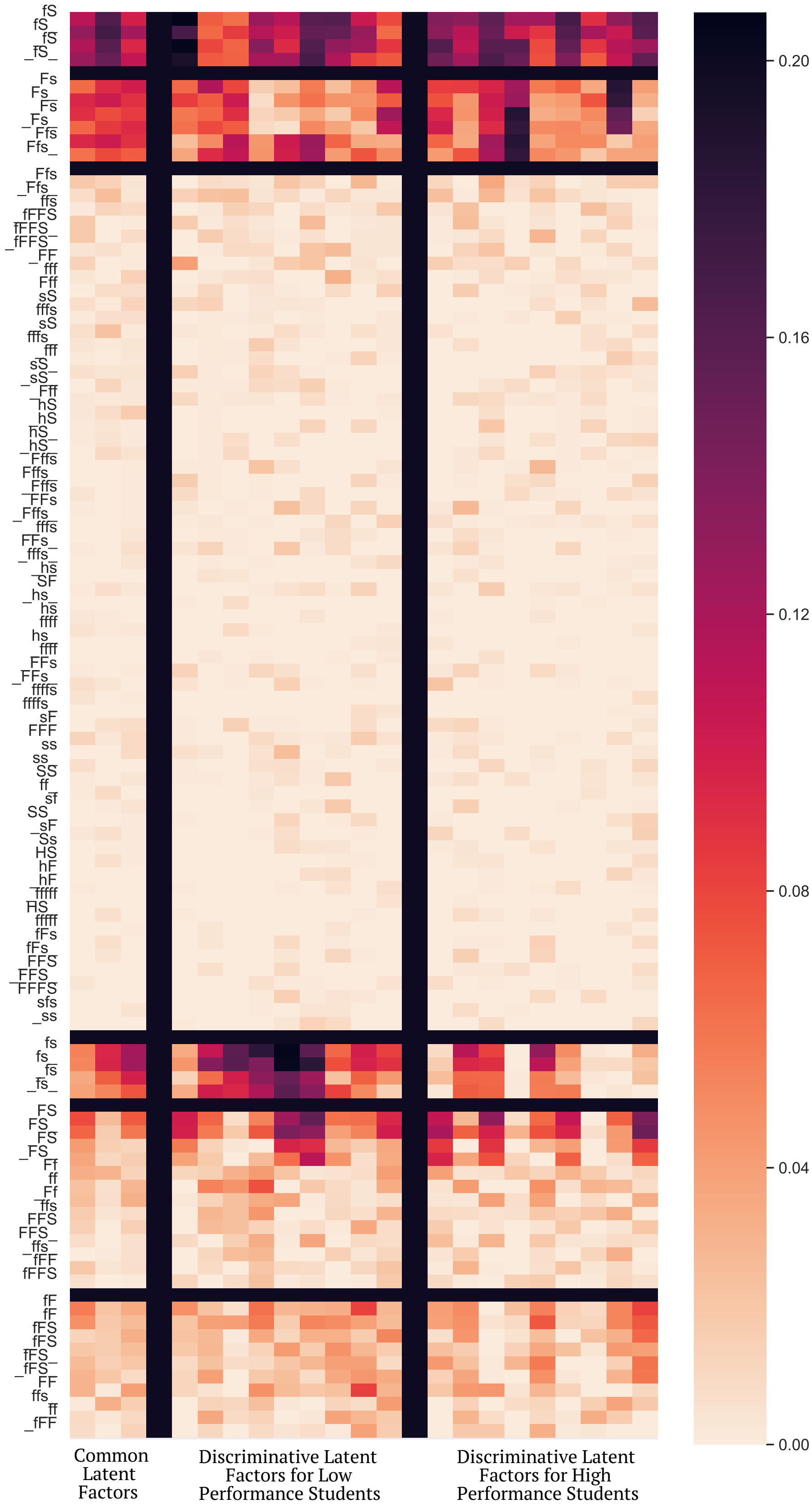


Figure 5.3: Heatmap shows the distribution of latent factors in common and discriminative parts for OLI Psychology

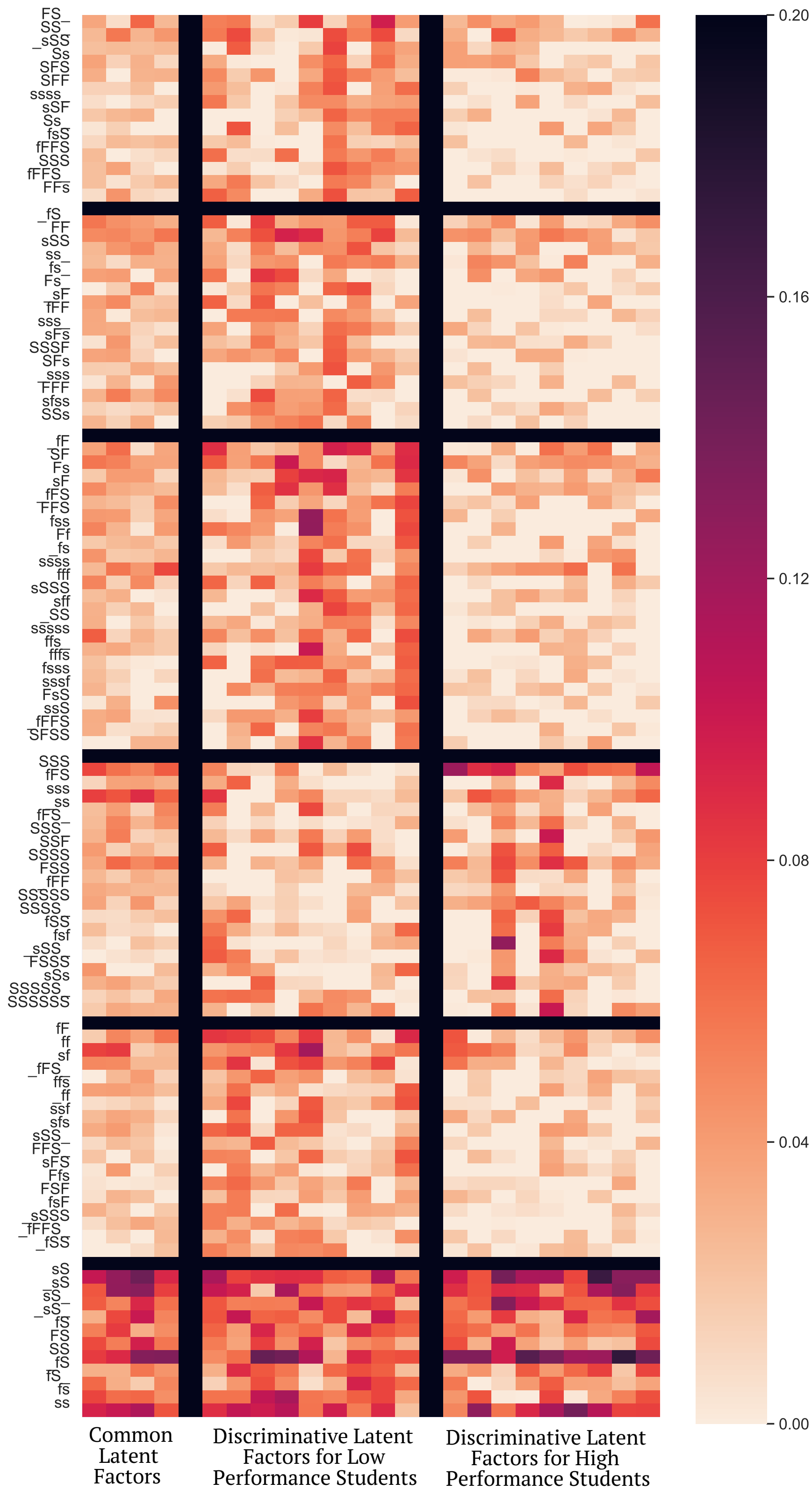


Figure 5.4: Heatmap shows the distribution of latent factors in common and discriminative parts for OLI Statistics

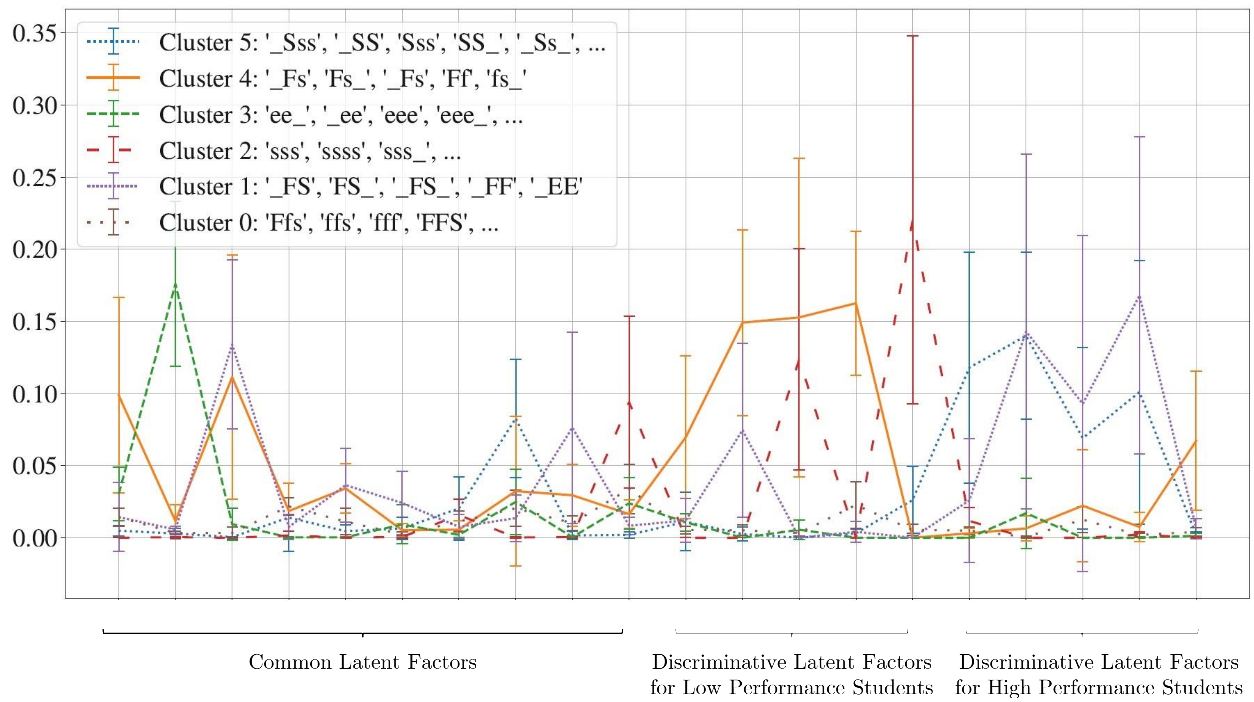


Figure 5.5: Latent factors for 6 clusters and respective patterns in Mastery Grids

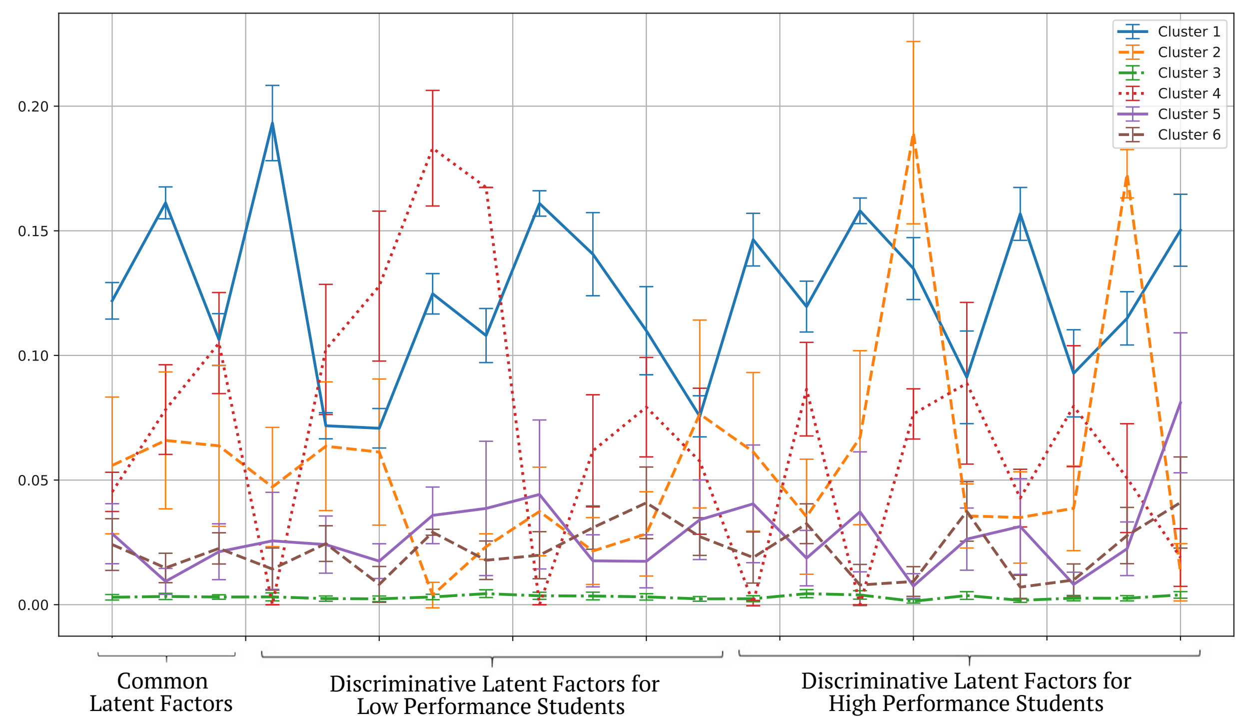


Figure 5.6: Latent factors for 6 clusters and respective patterns for OLI Psychology

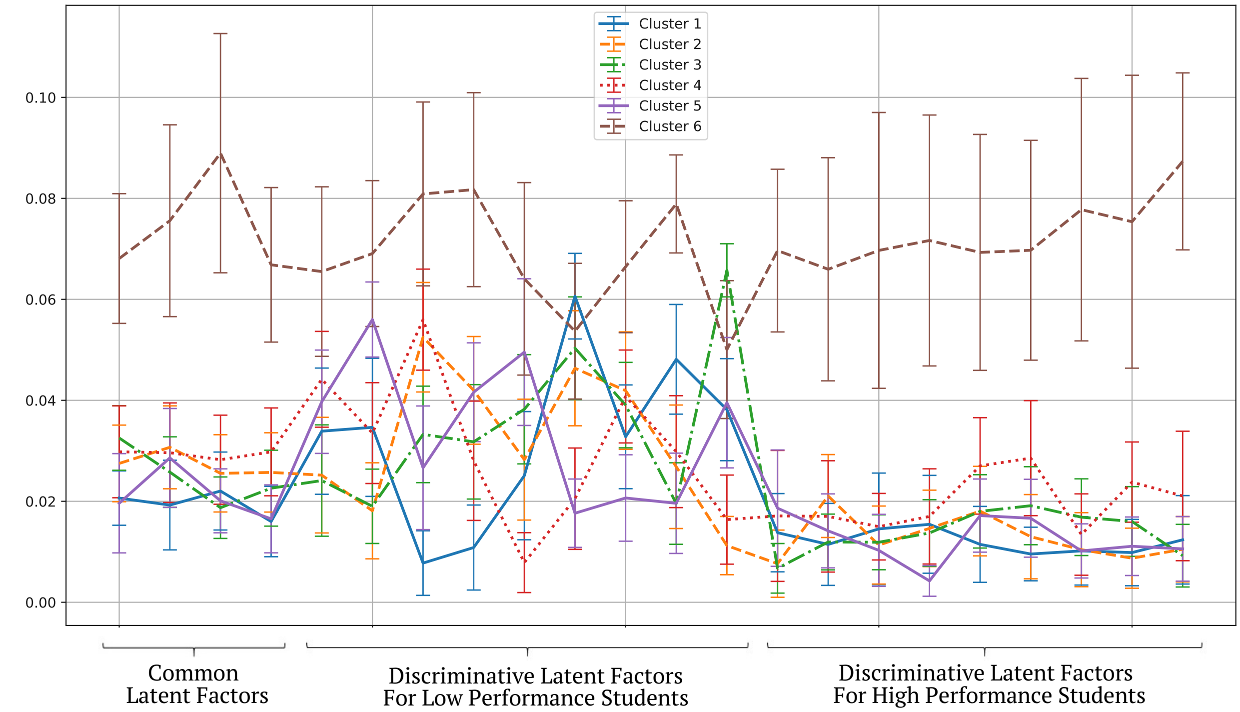


Figure 5.7: Latent factors for 6 clusters and respective patterns for OLI Statistics

5.4 Conclusion

In this chapter, we answered the second research question by proposing a framework to discriminate learning behavior trait patterns vs. performance-indicator patterns of students from student sequences in an online learning environment. In our analyses, we have shown that we can discover meaningful pattern clusters based on the latent factors that we find using discriminative non-negative matrix factorization. These patterns demonstrate that high-performance students either repeat their success if they have achieved it by spending a long time or try to reinforce what they have learned after a long failure by spending the time to get the problem right again. Low-performing students either hastily repeat their successful attempts over and over again without spending enough time or leave the problem with just one short success after a long failure, only not to learn from it.

CHAPTER 6

SB-DNMF: A Structure Based Discriminative Non-negative Matrix Factorization Model for Detecting Inefficient Learning Behaviors

6.1 Introduction

In previous chapters, we demonstrated that only a few performance-related factors can be observed in the studying patterns, after clustering the students according to their learning behavior traits. In this chapter, we propose a novel discriminative matrix factorization method (SB-DNMF) that takes into account the structures of the patterns to model students' common studying traits, while differentiating between the distinct behaviors of high- and low-performing students. Considering that there are meaningful structural similarities and differences between the sequential patterns, we introduce a pattern similarity measure to quantify these structures. Additionally, we impose a similarity-based constraint on SB-DNMF based on the proposed similarity measure such that it can adopt to such structural similarities. In our experiments on real-world datasets, we show that SB-DNMF:

1. Has a good fit to students' sequential pattern features.
2. Can distinguish between the patterns associated with student traits, high learning gain, and low learning gain.
3. Discovers meaningful factors that can be interpreted for both students and patterns.

Additionally, we unveil meaningful associations between the discovered factors and other performance measures of students. The goal in this chapter is to answer the third research question in 1.2:

Could we recognize efficient and inefficient student behavioral patterns by associating them with their learning performance?

6.2 Modified Weighted Levenshtein (MWL) Distance Measure

In this section, we propose a new distance measure to calculate structural similarity between two micro-patterns. We build this distance measure based on the Levenshtein distance [51]. Levenshtein distance measures the difference between two sequences and calculates how similar they are. This distance measure, counts the minimum number of insertions, deletions, or substitutions required to change a string sequence into another one. For two strings a and b, with respective lengths i and j, this measure can be calculated as:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if} \min(i,j) = 0, \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{6.1}$$

However, since the sequence labels in our micro-patterns meaningfully represent the duration and type of activities, we cannot weight all the label edits equally. For example, the micro-patterns "Fss" and "Sss" both have one unit Levenshtein distance with "sss". However, in our application, since in "Sss" (similar to "sss") the student continues to succeed in 3 consecutive attempts, it should be more similar to "sss", compared to "Fss". As a result, we modify the Levenshtein distance to adopt it to our application. To do this, we weigh the edit distances such that attempts of the same type or length are assumed more similar and have less distance between them. To define attempt similarities, we create a weight graph that is shown in Figure 6.1. Accordingly, we define the substitution edit distance of every two labels as the shortest path weight between them in this graph. Based on this graph, the attempts of the same type such as "S" and "s" (solving a problem successfully) or of the same length (short or long) such as "S" and "F" (spending a long time on solving a problem) have a shorter distance, compared to the ones that have both different types and lengths, such as "F" and "s". So substitutions required to transform a sequence into another one depends on the type and length of the activity.

Additionally, we add another modification to account for continuation of the same attempts vs. different attempts in micro-patterns of different lengths. For example, between micro-patterns "ssf" and "sss", the sequence "sss" should be more similar to "ss" as the student continues to have the same attempt type (short successes) in it. Accordingly, we add more weights to different

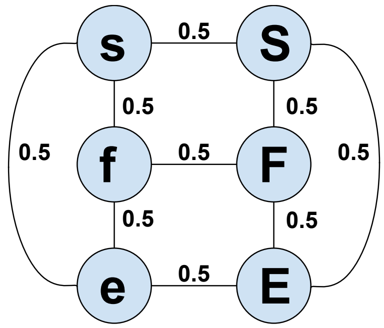


Figure 6.1: Distances between attempts used in the Modified Weighted Levenshtein distance measure. The distance between any two attempts is the weight of shortest path in the graph. For example $d(s, F) = 1$ or $d(e, S) = 1$ (In OLI datasets, "E" and "e" are replaced with "H" and "h")

	ss	fs	_Fs	Ss_	FS	ssss	ee	...	_FFf
ss	1.0	0.39	0.33	0.50	0.28	0.33	0.33		0.39
fs	0.39	1.0	0.61	0.56	0.44	0.50	0.56		0.39
_Fs	0.33	0.61	1.0	0.50	0.50	0.33	0.39	...	0.44
Ss_	0.50	0.56	0.50	1.0	0.67	0.61	0.39		0.39
FS	0.28	0.44	0.50	0.67	1.0	0.39	0.44		0.39
ssss	0.33	0.50	0.33	0.61	0.39	1.0	0.33		0.22
ee	0.33	0.56	0.39	0.39	0.44	0.33	1.0		0.28
⋮			⋮					⋱	
_FFf	0.39	0.39	0.44	0.39	0.39	0.22	0.28	...	1.0

Table 6.1: Similarity Matrix shows the similarity of each two patterns based on Modified Weighted Levenshtein distance measure in Mastery Grids.

consecutive attempts compared to the same ones. The added weight of insertion or deletion of the same consecutive attempt is 0.5, otherwise it is 1. After finding the distances, the similarity between two patterns p_i and p_j is calculated by their distances deduced from the maximum possible distance. We normalize the similarities to have the similarities between 0 and 1.

$$similarity(p_i, p_j) = \frac{(\max_{l,k:1\ldots m} distance(p_l, p_k)) - distance(p_i, p_j)}{(\max_{l,k:1\ldots m} distance(p_l, p_k)) - (\min_{l,k:1\ldots m} distance(p_l, p_k))} \tag{6.2}$$

Using MWL, we calculate the similarities between each two micro-patterns and build similarity matrix $S_{p\times p}$. A part of similarity matrix between micro-patterns in each dataset is illustrated in Table 6.1 to Table 6.3.

	fs	hS	Fffs	FFs	SF	hs	ffff	...	SS
fs	1.0	0.80	0.70	0.80	0.60	0.90	0.70		0.70
hS	0.80	1.0	0.50	0.60	0.70	0.90	0.50		0.80
Fffs	0.70	0.50	1.0	0.80	0.50	0.60	0.80	...	0.40
FFs	0.80	0.60	0.80	1.0	0.70	0.60	0.60		0.70
SF	0.60	0.70	0.50	0.70	1.0	0.60	0.50		0.90
hs	0.90	0.90	0.60	0.60	0.60	1.0	0.60		0.70
ffff	0.70	0.50	0.80	0.60	0.50	0.60	1.0		0.20
$\vdots$			$\vdots$					$\ddots$	
SS	0.70	0.80	0.40	0.70	0.90	0.70	0.20	...	1.0

Table 6.2: Similarity Matrix shows the similarity of each two patterns based on Modified Weighted Levenshtein distance measure in OLI Psychology.

6.3 Structure-Based Discriminative Non-negative Matrix Factorization (SB-DNMF)

We follow the assumption that the micro-patterns are of two –trait vs. performance– types (behavior assumption). Furthermore, we expect to see the trait micro-patterns in both low and high performance students (commonality assumption); while we anticipate some of the performance-related patterns to be stronger in low learning gain students vs. the high learning gain ones, and vice versa (discriminative assumption). Eventually, we assume that the presence of different micro-patterns in students' pattern vectors follow some hidden group-like arrangements that can be represented by a shared set of latent factors (shared latent assumption). For example, a latent factor can represent the example-reading activity in students. As a result, students who read many examples, as well as the example reading micro-patterns, can be grouped together.

6.3.1 SB-DNMF: the Model

Our goal is to detect these 3 sets of patterns from students' sequential attempts:

- Common patterns among all students
- Specific patterns for low-performance students

	sS	fS	FS	SS	SF	Fs	SSS	...	FFs
sS	1.0	0.92	0.83	0.92	0.83	0.75	0.83		0.67
fS	0.92	1.0	0.92	.83	.75	0.83	0.75		0.75
FS	0.83	0.92	1.0	0.92	0.83	0.92	0.83	...	0.83
SS	0.92	0.83	0.92	1.0	0.92	0.83	0.92		0.75
SF	0.83	0.75	0.83	0.92	1.0	0.75	0.83		0.75
Fs	0.75	0.83	0.92	0.83	0.75	1.0	0.75		0.92
SSS	0.83	0.75	0.83	0.92	0.83	0.75	1.0		0.75
$\vdots$			$\vdots$					$\ddots$	
_FFf	0.67	0.75	0.83	0.75	0.75	0.92	0.75	...	1.0

Table 6.3: Similarity Matrix shows the similarity of each two patterns based on Modified Weighted Levenshtein distance measure in OLI Statistics

- Specific patterns for high-performance students

We proposed model 6.3 in previous chapter that satisfies our assumptions. Here we propose another model that fulfils the similarity assumption as well.

$$\begin{aligned} L = \; & \|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2 + \\ & \|W_{1,c} - W_{2,c}\|_F^2 + \|W_{1,d}^T W_{2,d}\|_F^2 \end{aligned} \tag{6.3}$$

Integrating Pattern Structures. So far, we have included all assumptions, except the similarity assumption, in the model. As we have seen before, the extracted micro-patterns are mini-sequences of student activities. These mini-sequences have structural motifs that are important in our application. For example, consider the micro-pattern “sss” representing a student succeeding in solving problems in the same topic, fast, for three consecutive times. This micro-pattern is similar to “ssss” (four consecutive short successes) and different from “FF” (two consecutive long failures). Hence, we expect the latent factors of “sss” and “ssss” micro-patterns to be more similar than that of “FF”. However, our model so far is ignorant of such similarities and differences.

In this part, having the structural similarities between the micro-patterns in the similarity matrix $S_{p\times p}$, we augment our factorization model to impose these similarities. Particularly, we concatenate the matrices W_1 and W_2 to construct a pattern latent matrix W that includes both common and distinct pattern latent factors from both high- and low- performing data. Then, we calculate the matrix WW^T as the latent factor-based similarity between patterns. Our expectations is that this latent factor-based similarity matrix resembles the structural similarity between patterns. In other words, $S \sim \varepsilon WW^T$. Parameter ε here controls for the potential scaling effects.

We add this similarity as an additional Frobenius norm constraint to the objective function (6.3) to obtain a new objective function (6.4). Regularization terms on W and H are added to avoid over-fitting and α, β, γ, and δ are coefficients to control the importance of each term. To maintain interpretability, we impose non-negativity constraints on W and H.

$$\begin{aligned} L = \; & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \\ & \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\ & \delta\|S - \varepsilon WW^T\|_F^2 + (\|W\|^2 + \|H\|^2) \\ & \text{s.t. } W, H \geq 0 \end{aligned} \tag{6.4}$$

An illustration of our approach is presented in Figure 6.2.

6.3.2 Learning Model Parameters

We use the iterative Gradient Descent optimization algorithm, illustrated in Algorithm 1, to minimize the objective function (6.4) according to the following gradients:

$$\begin{aligned} \frac{\partial L}{\partial W_{1,c}} = & -2\gamma(X_1 - W_1 H_1^T)H_{1,c}^T + \\ 2\alpha(W_{1,c} - W_{2,c}) + 2W_{1,c} & - 2 \times 2\delta(S - \varepsilon WW^T)W_{1,c} \end{aligned} \tag{6.5}$$

$$\begin{aligned} \frac{\partial L}{\partial W_{2,c}} = & -2\gamma(X_2 - W_2 H_2^T)H_{2,c}^T - 2\alpha(W_{1,c} - W_{2,c}) + \\ & 2W_{2,c} - 2 \times 2\delta(S - \varepsilon WW^T)W_{2,c} \end{aligned} \tag{6.6}$$

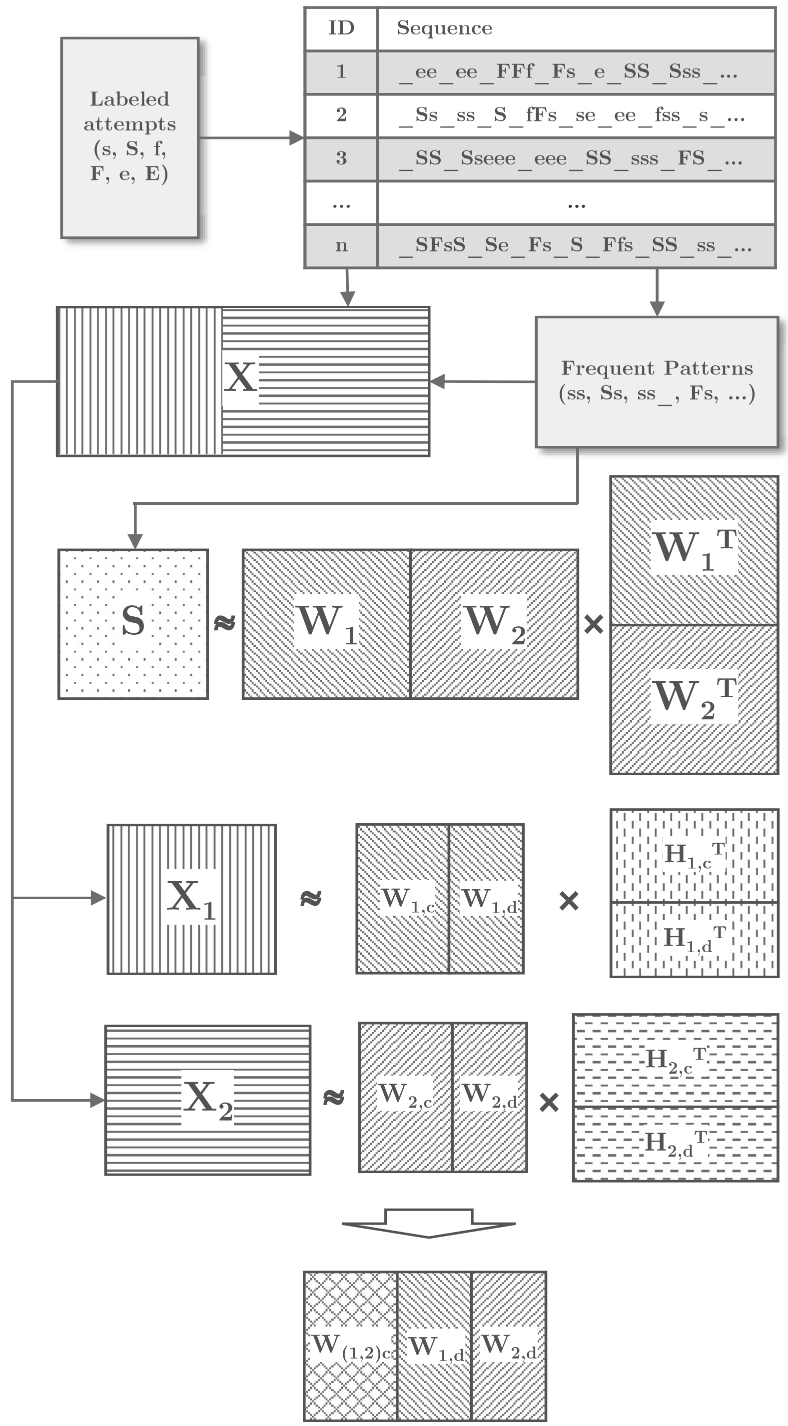


Figure 6.2: The process of building the matrix $W_{c,d}$ for clustering the patterns. X_1 and X_2 are pattern-student matrices for low- and high- performance students. We decompose them and apply the similarity matrix S to enforce having similar patterns in each cluster. The matrix $W_{(1,2)c}$ is the average of $W_{1,c}$ and $W_{2,c}$.

$$\begin{aligned}\frac{\partial L}{\partial W_{1,d}} = -2\gamma(X_1 - W_1 H_1^T) H_{1,d}^T + \\ 2\beta(W_{1,d}^T W_{2,d}) W_{2,d} + \\ 2W_{1,d} - 2 \times 2\lambda(S - \varepsilon W W^T) W_{1,d}\end{aligned} \tag{6.7}$$

$$\begin{aligned}\frac{\partial L}{\partial W_{2,d}} = -2\gamma(X_2 - W_2 H_2^T) H_{2,d}^T + \\ 2\beta(W_{1,d}^T W_{2,d}) W_{1,d} + \\ 2W_{2,d} - 2 \times 2\lambda(S - \varepsilon W W^T) W_{2,d}\end{aligned} \tag{6.8}$$

$$\frac{\partial L}{\partial H_1} = -2\gamma(X_1 - W1 H_1^T) W_1 + 2H_1 \tag{6.9}$$

$$\frac{\partial L}{\partial H_2} = -2\gamma(X_2 - W2 H_2^T) W_2 + 2H_2 \tag{6.10}$$

$$\frac{\partial L}{\partial \varepsilon} = (-2WW^T) : (S - \varepsilon W W^T) \tag{6.11}$$

6.4 Experiments

6.4.1 Evaluating SB-DNMF's Fit

To evaluate the model and ensure that we are discovering the correct parameters, we calculate the root mean square error (RMSE) and mean absolute error (MAE) of the observed pattern matrices versus the reconstructed ones. The lower these values are, the better the fit would be. To have a point of comparison, we compare SB-DNMF's fit with the following two baseline algorithms:

- NMF: the standard non-negative matrix factorization, which is similar to SB-DNMF without imposing $f_c(.)$ and $f_d(.)$ and the similarity constraint in (6.4);
- DNMF: the discriminative non-negative matrix factorization, which is similar to SB-DNMF without imposing the similarity constraint in (6.4).

The results, shown in Table 6.4, demonstrate that the reconstruction error of SB-DNMF is small, which means that it can accurately recover the data matrix. However, SB-DNMF's error is slightly higher than DNMF or NMF. This is expected because of the additional restrictions applied to the model to find common and discriminative patterns in the data. In other words, we lose a small reconstruction accuracy in exchange for finding meaningful patterns in the data.

Algorithm 1: SB-DNMF Learning algorithm

Input: Matrices X_1, X_2 and S, values K, K_c, K_d and parameters α , β , ρ;
Output: Matrices W_1, H_1, W_2, H_2 and ε;
Initialize W_1, H_1, W_2, H_2 and ε randomly;
for $i \leftarrow 1$ **to** $epoch$ **do**
  Split W_1 to $W_{1,c}$ and $W_{1,d}$
  Split W_2 to $W_{2,c}$ and $W_{2,d}$
  Split H_1 to $H_{1,d}$ and $H_{2,d}$
  Update $W_{1,c}$ according to (6.5)
    $W_{1,c} \leftarrow W_{1,c} - \frac{\partial L}{\partial W_{1,c}}$
  Update $W_{2,c}$ according to (6.6)
    $W_{2,c} \leftarrow W_{2,c} - \frac{\partial L}{\partial W_{2,c}}$
  Update $W_{1,d}$ according to (6.7)
    $W_{1,d} \leftarrow W_{1,d} - \frac{\partial L}{\partial W_{1,d}}$
  Update $W_{2,d}$ according to (6.8)
    $W_{2,d} \leftarrow W_{2,d} - \frac{\partial L}{\partial W_{2,d}}$
  Update H_1 according to (6.9)
    $H_1 \leftarrow H_1 - \frac{\partial L}{\partial H_1}$
  Update H_2 according to (6.10)
    $H_2 \leftarrow H_2 - \frac{\partial L}{\partial H_2}$
  Update ε according to (6.11)
    $\varepsilon = \varepsilon - \frac{\partial L}{\partial \varepsilon}$
  $W_1 \leftarrow$ Append $W_{1,c}$ and $W_{1,d}$
  $W_2 \leftarrow$ Append $W_{2,c}$ and $W_{2,d}$
end

		Mastery Grids		OLI Psychology		OLI Statistics	
		X1 Error	X2 Error	X1 Error	X2 Error	X1 Error	X2 Error
RMSE	NMF	0.0212	0.0199	0.0444	0.0439	0.0384	0.0301
	DNMF	0.0235	0.0224	0.0543	0.0555	0.0428	0.0424
	SBDNMF	0.0246	0.0236	0.0554	0.0541	0.0540	0.0521
MAE	NMF	0.0123	0.0122	0.0213	0.0311	0.0221	0.0177
	DNMF	0.0129	0.0127	0.0413	0.0444	0.0240	0.0227
	SBDNMF	0.0144	0.0139	0.0378	0.0408	0.0311	0.0293

Table 6.4: The reconstruction errors of X_1 and X_2 (RMSE and MAE) with NMF, DNMF and SB-DNMF for different datasets

Hyperparameter Tuning: The reported results are based on the best hyperparameters found according to hyperparameter tuning. We use grid-search on the combination of parameters α, β, δ, K, number of common latent factors K_c, and distinct ones K_d. Particularly, we vary α, β, and δ in the range $[0.1 \dots 0.9]$, and vary K between 2 and 30 and for each K, vary K_c between 0 and K whilst $K = K_c + K_d$. The best found combinations are:

- **Mastery Grids:** $k = 20$, $k_c = 12$, $k_d = 8$, $\alpha = 0.6$, $\beta = 0.1$, $\gamma = 1.3$ and $\delta = 0.9$.
- **OLI Psychology:** $k = 19$, $k_c = 6$, $k_d = 13$, $\alpha = 0.1$, $\beta = 0.9$, $\gamma = 1.0$ and $\delta = 0.9$.
- **OLI Statistics:** $k = 19$, $k_c = 4$, $k_d = 15$, $\alpha = 0.1$, $\beta = 0.5$, $\gamma = 1.4$ and $\delta = 0.8$.

6.4.2 Finding Trait vs. Performance Patterns

In this section we analyze the discovered pattern latent factors and cluster them into different groups, representing trait, high-performance, and low-performance ones. Specifically, we build a matrix $W_{c,d}$ for all patterns from the decomposed matrices to cluster the patterns as explained in 5.3.2 and shown in Figure 6.2. We perform Spectral clustering on this pattern-latent matrix and find 6 distinct clusters among them.

Mastery Grids; The heatmap in Figure 6.3 illustrates $W_{c,d}$ arranged into the discovered clusters. The rows represent micro-patterns and the columns are latent factors. The pattern clusters are separated with horizontal lines and sets of common vs. distinct latent factors are separated with vertical lines. The first $K_c = 12$ columns represent the common latent factors, the $K_d = 8$ columns in the middle are discriminative latent factors for low performance students ($W_{1,d}$), and the 8 columns on the right side are the latent factors of high performance students ($W_{2,d}$).

As we can see, the patterns in each cluster have meaningful similarities. For example, the patterns that end with a short successful attempt of solving a problem (“sss_”, “_Fs_”, “_Ss_”, “ssss_”) are in one cluster; the patterns containing short reading examples (“ee_”, “_ee”, “eee”, “eee_”) are in one cluster; and the patterns that start with long attempts are in another cluster (“_Ss”, “_Fs”, “_FF”, “_Sss”). This shows the effectiveness of the imposed structural similarity measure. Also, looking at different common and distinct latent factors in the heatmap, we see meaningful patterns among them. The color intensity shows that a pattern cluster belongs to each set of latent factors. For example, the third pattern cluster with multiple short successes in a row (“Sssss”, “fssss”, “sssss”, “Ssss”) have higher weight in the distinct low-performance latent factors and a low weight in high-performance and common factors. This means that the students that persist on solving the problems of the same topic quickly without spending enough time on them, tend to have lower learning gains. On the other hand, the patterns in the last cluster that start with long attempts (“_EE”, “_FF”, “_Ssss”, “_SS”) have higher weights in the distinct high-performance latent factors and lower weights in low-performance latent factors. It shows that the students who spend more time on the problems and examples, whether they fail in solving a problem or succeed in it, have a higher learning gain.

Furthermore, the patterns in the fourth cluster that are associated with spending a short time in repeatedly reading examples (“_eee_”, “eeee”, “eee_”, “_eee”) have higher weights in the common latent factors. This cluster has lower weight in low and high performance latent factors. It shows that reading examples repeatedly is a common activity between low and high performance students. Thus, it can be interpreted as a learning trait, rather than a performance trait. The first cluster (changing topics after at least one short success) shows a more mixed results. It has mostly common or low learning gain latent factors. However, some of the shorter patterns in that cluster, such as “_Fs_” are more prevalent in the high learning gain ones. Additionally, the second cluster (starting the topic with a long failure and struggling to getting it right) is mostly associated with low-performing factors, but can be seen in the common ones too.

OLI Psychology; The heatmap in Figure 6.4 illustrates $W_{c,d}$ arranged into the discovered clusters. The first $K_c = 6$ columns represent the common latent factors, the $K_d = 13$ columns in the middle are discriminative latent factors for low performance students $(W_{1,d})$, and the 13 columns on the right side are the latent factors of high performance students $(W_{2,d})$. The patterns in each cluster have meaningful similarities. For example, the patterns that ends with a short success after a short failure (“fs”, “fs_”,“_fs”) are in one cluster; the patterns containing multiple short failures (“fff”, “fffs”, “ffff”) are in one cluster; These patterns are similar to the patterns that we found in Chapter 4 for “confirmers” but in another way. “Confirmors“ solve a problem successfully and confirm their success, but this group shows the multiple failures in the same problem. the patterns with long failures (“FF”, “FFS”, “FFF”) are in one cluster; and patterns with long attempts (“fFS_”, “fFFS”, “FFS_”) are in another cluster. This cluster of patterns has the same behavior of “thinkers” that we found in Chapter 4.

The patterns in the third cluster that have multiple short failures and in the fifth cluster that have multiple long failures have higher weights in low performance latent factors. It shows that having multiple failures either short or long is associated with low-performance students. In other words, multiple failures in solving a problem is an inefficient pattern and students should learn the concepts before start solving the problems. Furthermore, the patterns in the fourth cluster that start with a short failure and followed by a short success ("fs", "fs_", "_fs", "_fs_") have relatively higher weights in the distinct low-performance latent factors and lower weights in high-performance latent factors. It means that students that solving a problem successfully after a failure is an inefficient behavior and students should either spend more time on solving the problems or practice more. The second cluster which include patterns with hints have higher weight in high performance latent factors. So high-performance students use hints more often. This is interesting because in Mastery Grids reading examples is a learning trait and not a performance trait. But in OLI Psychology, it shows that high-performance students use hints and solve the problems more carefully.

OLI Statistics; The heatmap in Figure 6.5 illustrates $W_{c,d}$ arranged into the discovered clusters. The first $K_c = 4$ columns represent the common latent factors, the $K_d = 15$ columns in the middle are discriminative latent factors for low performance students $(W_{1,d})$, and the 15 columns on the right side are the latent factors of high performance students $(W_{2,d})$.

The patterns in each cluster have meaningful similarities. For example the patterns of multiple long successful attempts ("SSS_", "SSSS", "sSSS", "SSSSS") are in one cluster; These are the "confirmers" and they confirm their successful attempts on solving problems. Long attempts are in one cluster ("FS", "SF", "FF", "FFS"); These are the "thinkers" that spend more time on solving problems. And the patterns with multiple failures ("fFS", "fFF", "fffs", "fFFS") are in another cluster. This behavior is also seen in the previous dataset that students have multiple failures in solving one problem.

The first cluster have higher weights in high performance latent factors and lower weights in low performance latent factors. This cluster contains relatively few number of successful attempts for one topic. It means that practicing the problems within a topic is associated with high-performance students. However, these attempts shouldn't take longer than 4 attempts. The fourth and sixth cluster have higher weights in low-performance latent factors and lower weights in high-performance latent factors. The fourth cluster consist of multiple long failures ("FF", "FFS", "FFS_", "SFF"). It means that the low-performance students struggle solving problems. One possibility is that they don't have enough knowledge about the topic and should use hints.

And the sixth cluster has patterns with two attempts ("fs", "sS_", "fS", "fF"). It demonstrates that low-performance students don't have enough practice and should solve more problems in a topic. Although some of the patterns are long attempts here, the number of problems solved are not enough for the students to master a topic. Another explanation is that students are divided by score, not learning gain. As a result, we don't know who has learned vs. if they already knew the concepts. So our conclusion here is not reliable.

6.4.3 Evaluating SB-DNMF's Clustering

In previous section, we have seen that SB-DNMF produces meaningful pattern clusters. To have an additional quantitative measure of how well SB-DNMF performs compared to simpler models in finding meaningful clusters, we use the Silhouette index (the higher is better) [70]. Particularly, we cluster the pattern latent vectors obtained by the baseline methods NMF and DNMF, and measure how consistent the achieved pattern clusters are. The achieved index scores are:

- Mastery Grids: $S_{NMF} = 0.0319$, $S_{DNMF} = 0.0286$ and $S_{SBDNMF} = 0.1267$.
- OLI Psychology: $S_{NMF} = 0.0506$, $S_{DNMF} = 0.0480$ and $S_{SBDNMF} = 0.2610$.
- OLI Statistics: $S_{NMF} = 0.0399$, $S_{DNMF} = 0.0.0633$ and $S_{SBDNMF} = 0.0863$.

SBDNMF has the highest Silhouette index, meaning that having a discriminative model and adding the structure-based similarity to it produces more coherent pattern clusters.

We perform NMF on the pattern vectors to compare with our proposed method. The pattern clusters are demonstrated in Figures 6.6 to 6.8. The heatmaps show that using NMF, the similar patterns will not fall into the same clusters. On the other hand, the NMF is not able to discriminate between common and specific patterns. But our proposed method using structure cluster the patterns properly and also is able to find discriminative and common patterns for low- and high-performance students.

6.4.4 Detecting Performance Patterns within each Student Group

Similar to pattern latent matrices, in this section we analyze the discovered student latent matrices. As the students are already divided into two groups according to their learning gains, the

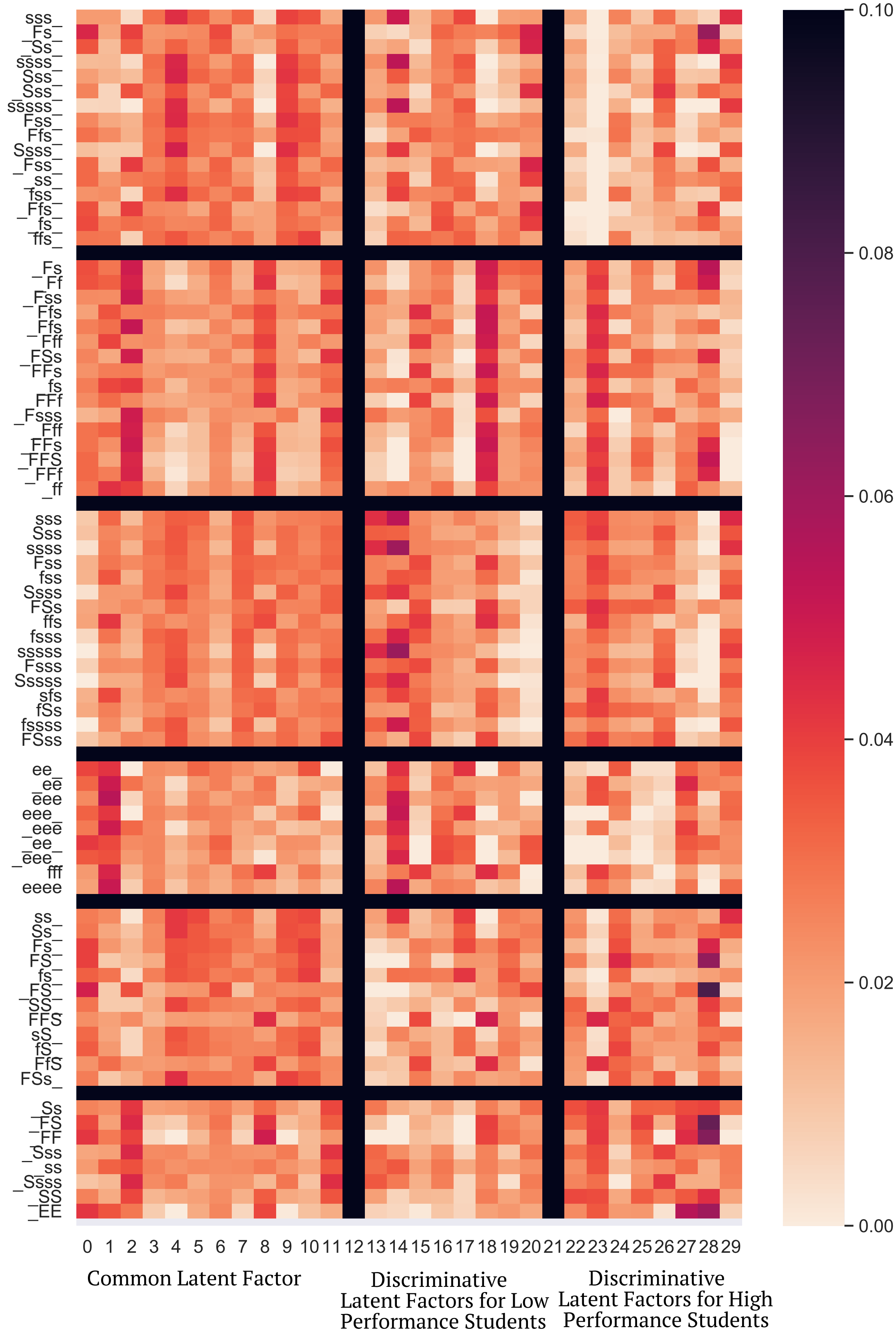


Figure 6.3: Latent factors 0 to 9 represent common patterns, 11 to 20 represent low performance students' patterns and 22 to 29 represent high performance students' patterns in Mastery Grids. Patterns are clustered using Spectral clustering algorithm in 6 groups. In each cluster, one or more latent factors have higher weights.

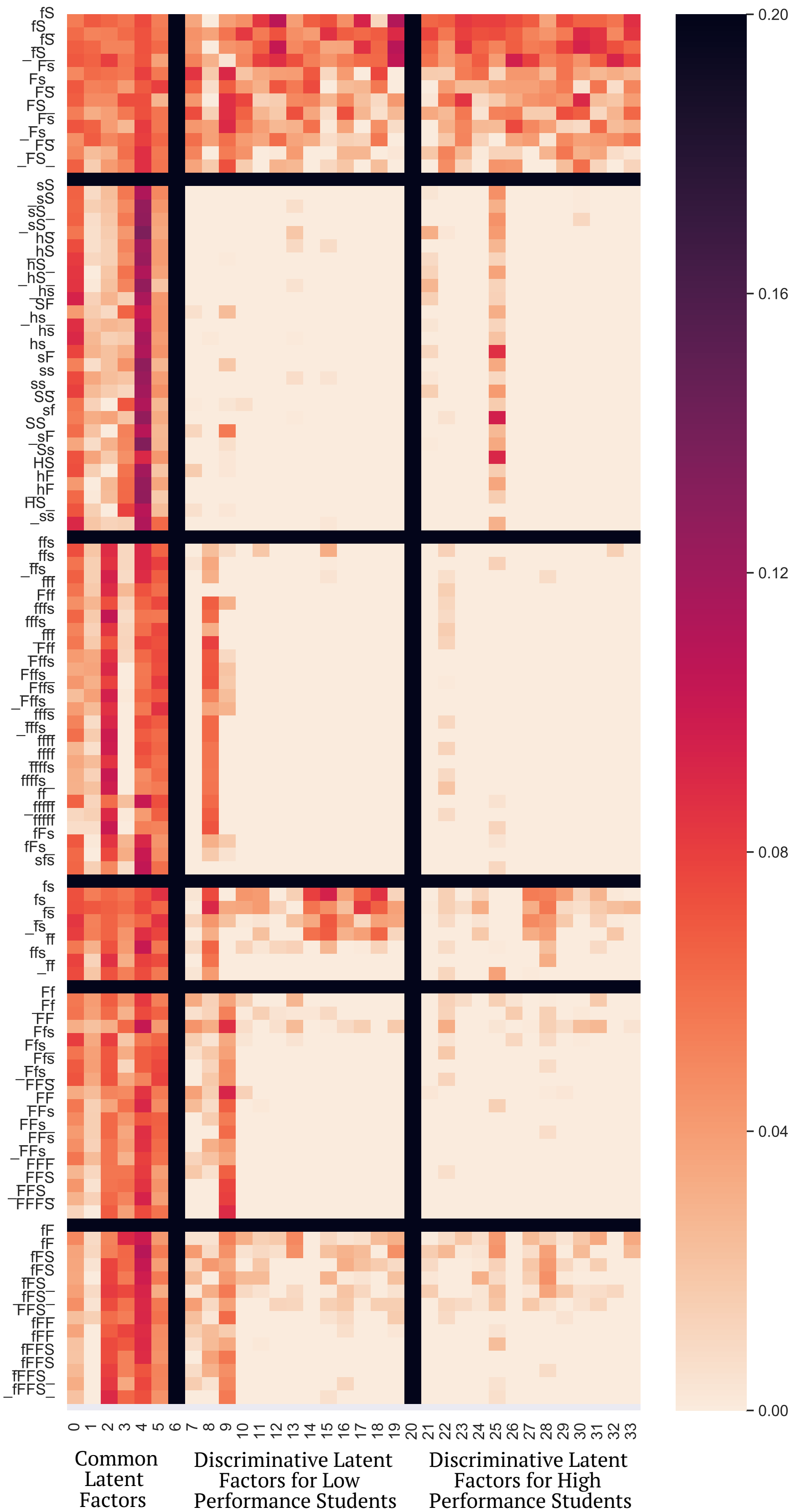


Figure 6.4: Latent factors 0 to 5 represent common patterns, 7 to 19 represent low performance students' patterns and 21 to 33 represent high performance students' patterns in OLI Psychology.

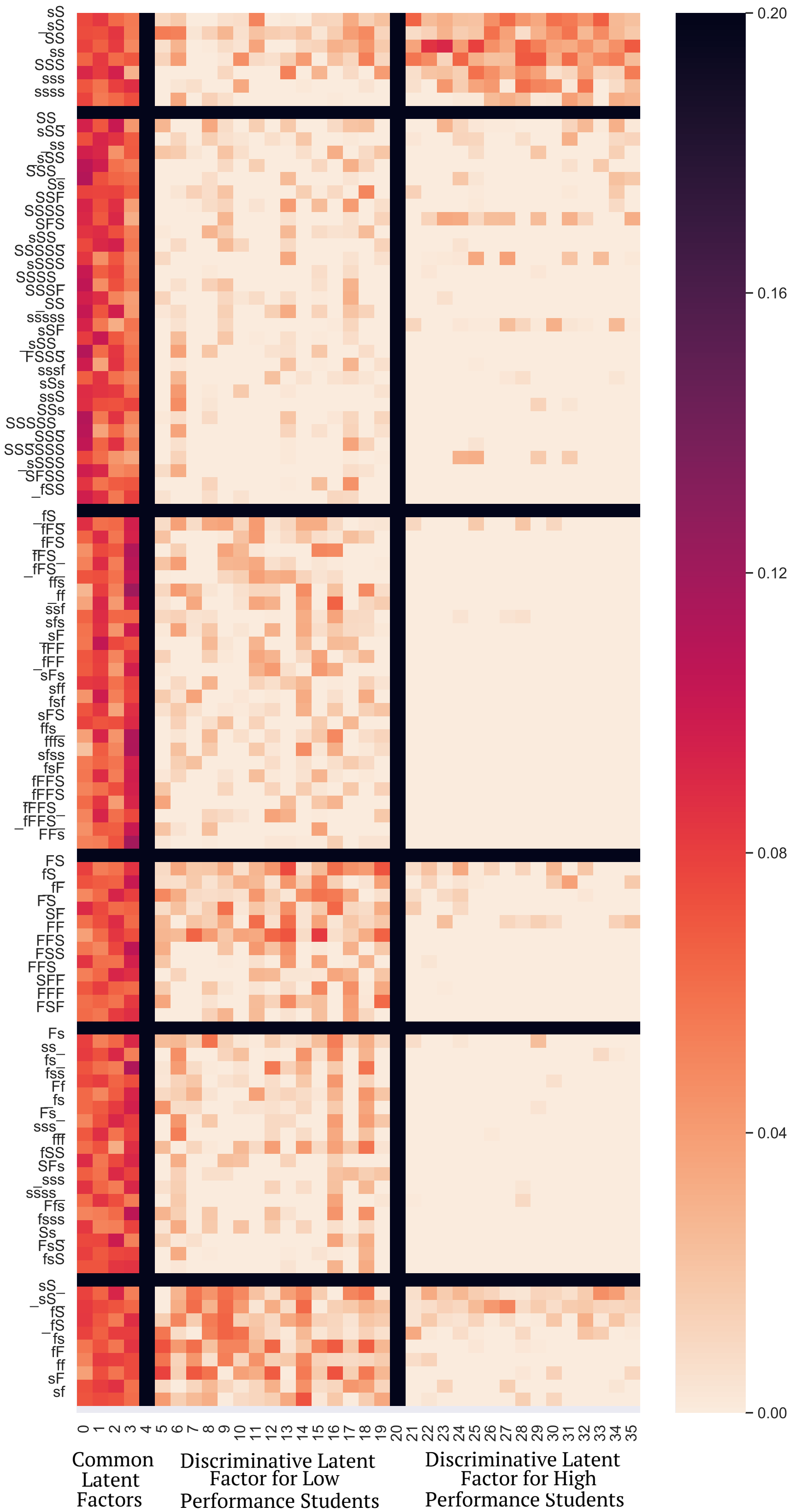


Figure 6.5: Latent factors 0 to 3 represent common patterns, 5 to 19 represent low performance students' patterns and 21 to 35 represent high performance students' patterns in OLI Statistics.

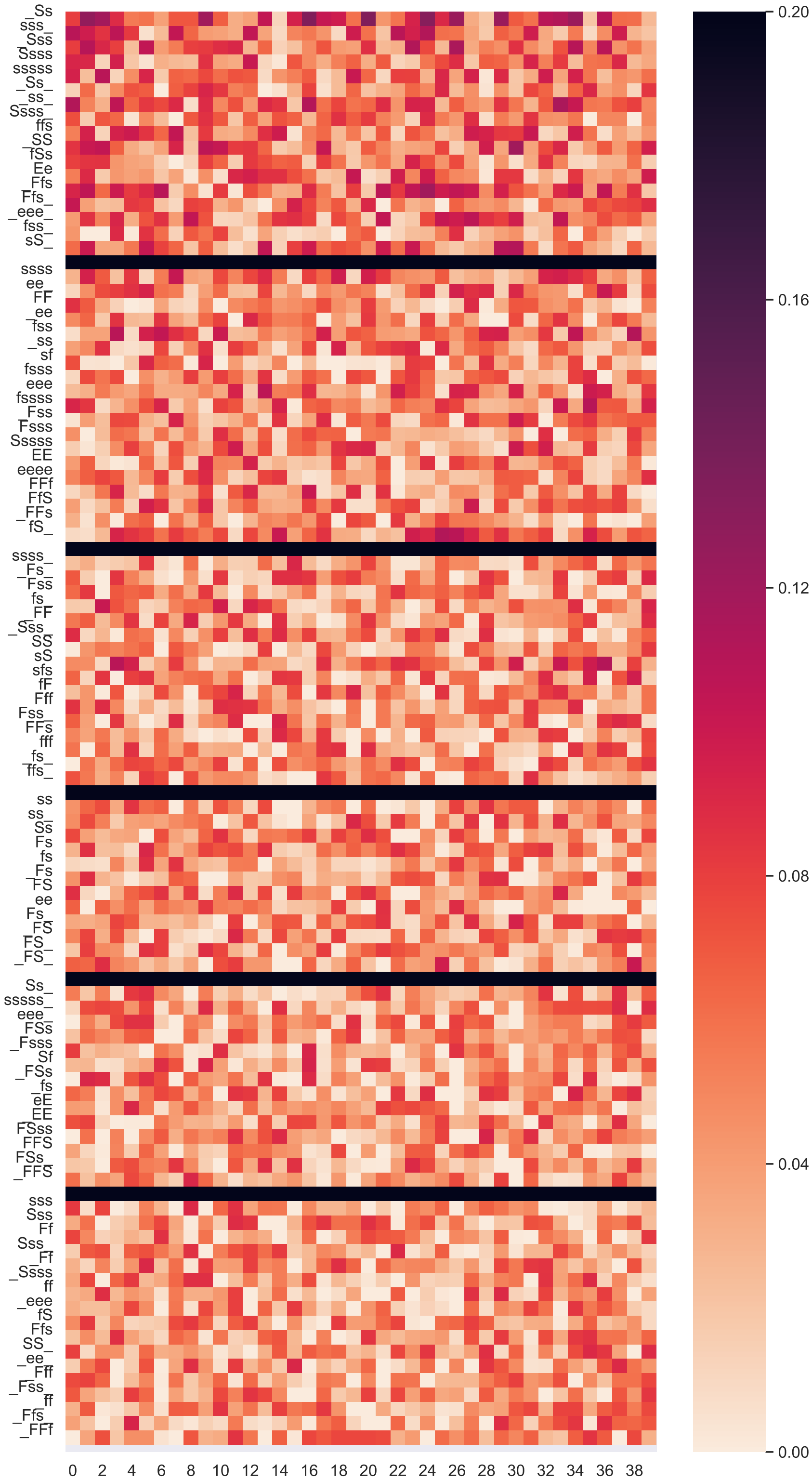


Figure 6.6: Pattern clusters with NMF on Mastery Grids

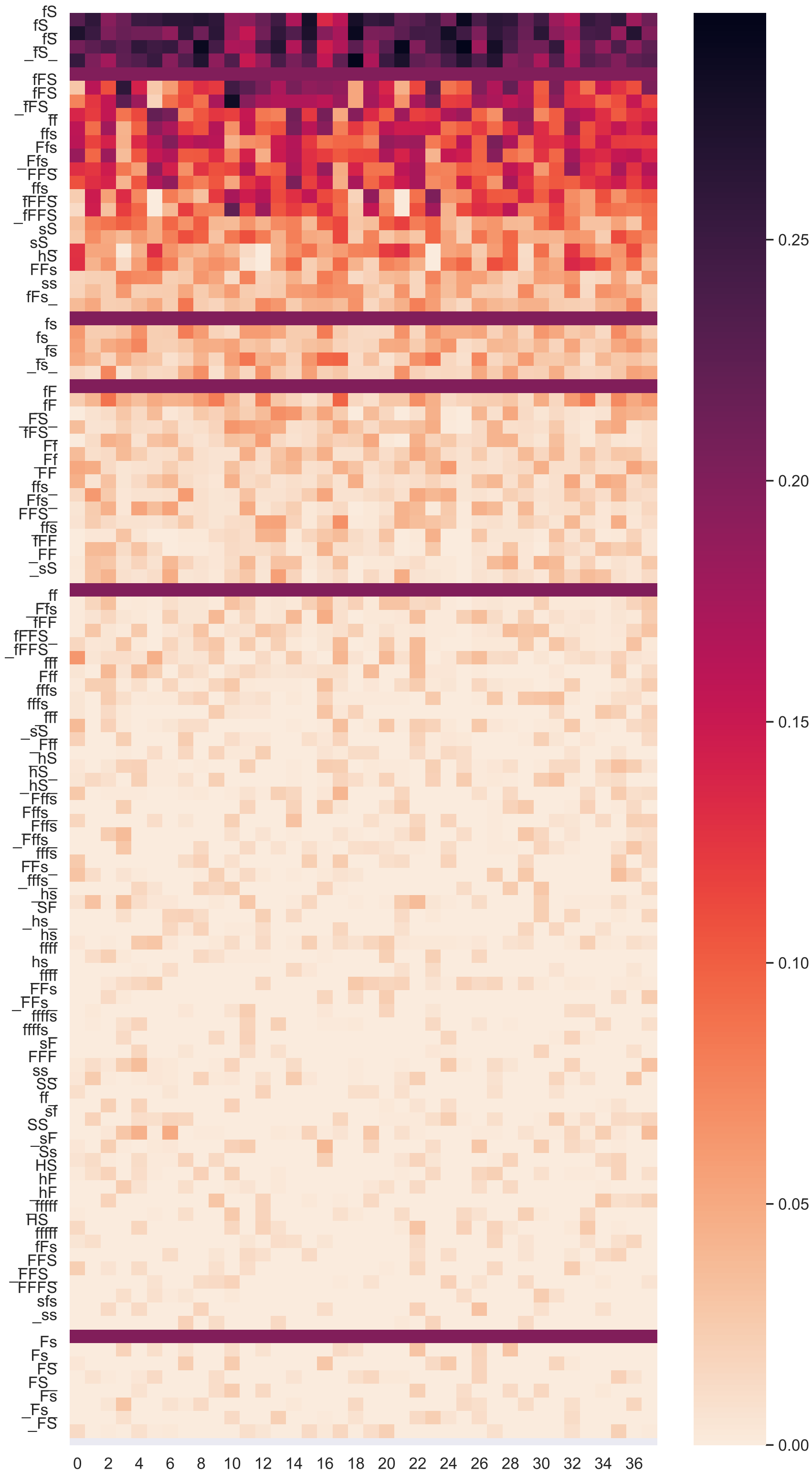


Figure 6.7: Pattern clusters with NMF on OLI Psychology

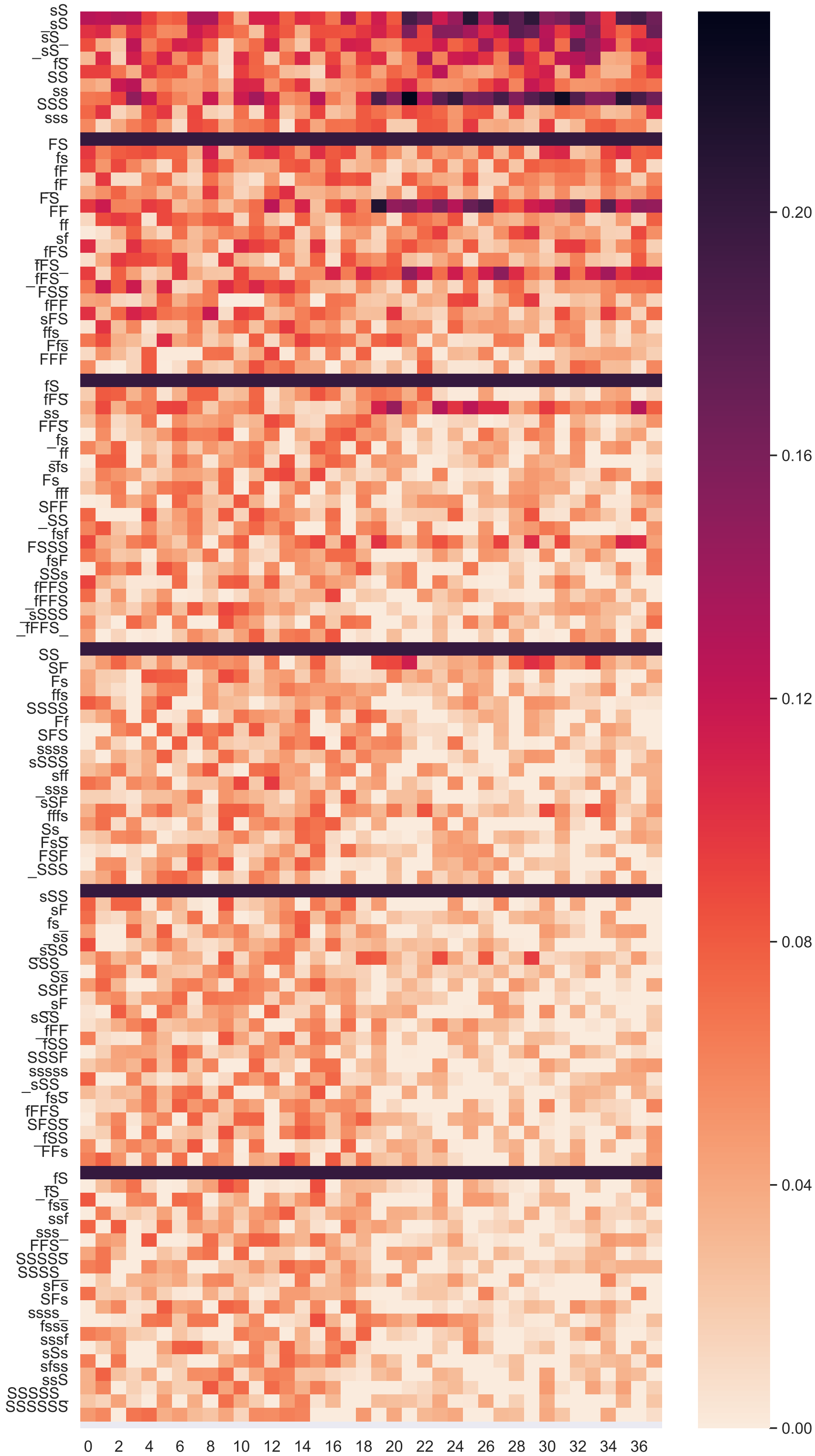


Figure 6.8: Pattern clusters with NMF on OLI Statistics

different student latent matrices do not provide additional information on this specific performance measure. So, we look at the different latent patterns in high and low learning gain students , to evaluate if there are any indicators of other performance measures in them. To this end, we calculate the Spearman correlation between the latent factors in these two matrices and the students' pre-test and post-test scores.

Mastery Grids; The results are shown in Tables 6.5 and 6.6 for the high-performing group and low-performing group respectively[1]. As we can see in Tables 6.5, latent factor $\#4$ in $H_{2,d}$ has a small, but significant positive correlation with students' pre-test scores. This means that high learning gain students, who have a high pre-test score show more of this latent behavior, compared to the ones that have a low pre-test score. This correlation can be observed in Figure 6.10. Looking at the previous section's results, we can see that this latent factor (corresponding to latent factor $\#26$ in $W_{c,d}$ in Figure 6.3), has the least association with the example-reading activity (cluster 4), while being more prevalent in the other pattern clusters. Interestingly, it shows that high learning gain students who already have a high prior knowledge when starting the course (indicated by a high pre-test) do not read the examples that are provided in the learning system. On the other hand, reading examples while having a low prior knowledge is associated with students with high learning gains.

In Table 6.6 for low learning-gain students, we can see that latent factor $\#3$ in $H_{1,d}$ ($\#14$ in Figure 6.3) has a slight but significant negative correlation with students' post-test score. This means that low learning gain students, who have a high post-test score show less of this latent behavior, compared to the ones that have a low post-test score. We can clearly see in Figure 6.3 that patterns with multiple short successes, such as ("Ssss", "fssss", "sssss", "Fsss"), belong to this latent factor. Also, the patterns that include many long attempts (in clusters 2, 5, and 6 of Figure 6.3) have the least association with this latent factor. This observation confirms the ineffectiveness of repetitive short success attempt: the students who do not gain much knowledge during the class and have a low performance at the end of it are the ones who aimlessly repeat the same topic well after mastering it and avoid moving on to other topics. Also, it shows that the students with low learning gain and high post-test scores (indicating the ones that have a high prior knowledge at the beginning of the class) tend to dwell more on each learning material (cluster 6), even if they get the problems wrong on their first attempts (cluster 2), and switch topics after succeeding in them

[1]The latent factors did not have any significant correlation with high learning gain students' post-test scores and low learning gain students' pre-tests.

(cluster 5).

OLI Psychology; The results are shown in Table 6.7 for the low-performing group and Table 6.8 and Table 6.9 for the high-performing group respectively[2]. As we can see in Table 6.7, latent factor $\#12$ in $H_{1,d}$ has a significant positive correlation with students' post-test scores. This means that low learning gain students, who have a high pre-test score, show more of this latent behavior compared to the ones that have a low pre-test score. This correlation can be observed in Figure 6.11. Looking at the previous section's results, we can see that this latent factor (corresponding to latent factor $\#19$ in $W_{c,d}$ in Figure 6.4), has the most association with the cluster of patterns with long success after short failures (such as "fS", "fS_" and "_fS"). It means that low-performance students who spend more time on a problem after a failure, have better post-test scores.

As we can see in Tables 6.8, latent factor $\#4$ in $H_{2,d}$ has a significant positive correlation with students' pre-test scores. This means that high learning gain students, who have a high pre-test score show more of this latent behavior, compared to the ones that have a low pre-test score. This correlation can be observed in Figure 6.12. Looking at the previous section's results, we can see that this latent factor (corresponding to latent factor $\#25$ in $W_{c,d}$ in Figure 6.4), has the most association with the cluster of patterns with hints. It means that high-performance students with prior knowledge tend to use hints more than low-performance students in this course. This is different in Mastery Grids. One reason could be the difference of hints and examples. Students could read examples without trying to solve a problem. But hints guide students to solve a problem correctly. So although students have higher knowledge, they want to make sure that they solve a problem successfully.

In Table 6.9 for high learning-gain students, we can see that latent factor $\#7$ in $H_{2,d}$ ($\#28$ in Figure 6.4) has a slight but significant negative correlation with students' post-test score. This means that high learning gain students, who have a high post-test score show less of this latent behavior, compared to the ones that have a low post-test score. This correlation is demonstrated in Figures 6.13. We can see in Figure 6.4 that patterns with short success after a short failure (cluster 4) belong to this latent factor. Our conclusion is that high-performance students that use short attempts have lower post-test scores. So such patterns are recognized as inefficient patterns for these students.

[2]The latent factors did not have any significant correlation with low learning gain students' pre-test scores.

Latent	s	t	correlation	p-value
0	0.179164	2.037871	0.365112	0.05147
1	0.190467	0.751712	0.143176	0.45872
2	0.190226	-0.79691	0.151592	0.43245
3	0.191545	0.505704	0.096865	0.61716
4	0.178176	2.1212	**0.377946****	0.04323
5	0.184166	-1.57594	0.290234	0.12668
6	0.189594	-0.90537	0.171652	0.37327
7	0.186515	1.321265	0.246435	0.19750

Table 6.5: The correlation between latent factors and test score of high performance students. Latent factor 4 has positive correlation with pre-test score in Mastery Grids.

Latent	s	t	correlation	p-value
0	0.203825	0.26538	0.054091	0.79294
1	0.203373	-0.42149	-0.085719	0.67713
2	0.200632	-0.91802	-0.184184	0.36777
3	0.190663	-1.8731	**-0.357131***	0.07328
4	0.202364	-0.64754	-0.131038	0.52349
5	0.203534	-0.37341	-0.076001	0.71212
6	0.204099	-0.07635	-0.015582	0.93974
7	0.199213	1.094501	0.218039	0.28451

Table 6.6: The correlation between latent factors and test score of low performance students. Latent factor 3 has negative correlation with post-test score in Mastery Grids.

OLI Statistics; The results are shown in Table 6.10, for the high-performing group[3]. As we can see in this table, latent factor $\#4$ in $H_{2,d}$ has a significant negative correlation with students' post-test scores. This means that high learning gain students, who have a high pre-test score, show less of this latent behavior compared to the ones that have a low pre-test score. This correlation can be observed in Figure 6.14. Looking at the previous section's results, we can see that this latent factor corresponds to latent factor $\#25$ in $W_{c,d}$ in Figure 6.5. So low-performance students have multiple successful attempts in one topic. This behavior of repeating the success is an efficient pattern and students should move to the next topic after mastering one.

6.5 Conclusion

In this chapter, we answered the third research question by proposing a method to discriminate between students' learning trait related versus performance related patterns in online learning

[3]The latent factors did not have any significant correlation with low learning gain students' post-test scores.

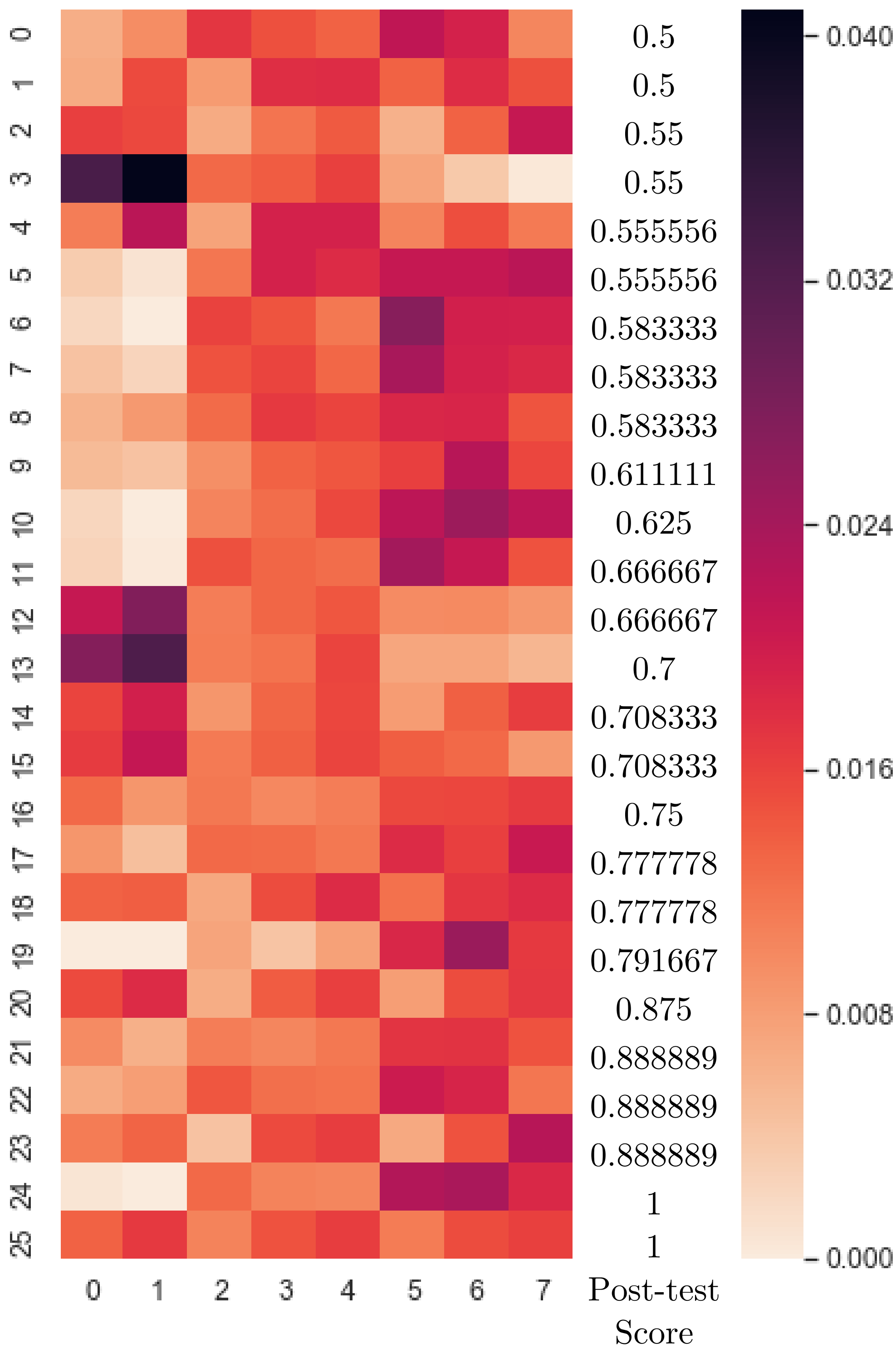


Figure 6.9: $H_{1,d}$ is the student-latent factor matrix and represents low performance students. The rows are ordered by students' post-test scores. The latent factor 3 has a correlation with post-test score and its values decrease with higher scores.

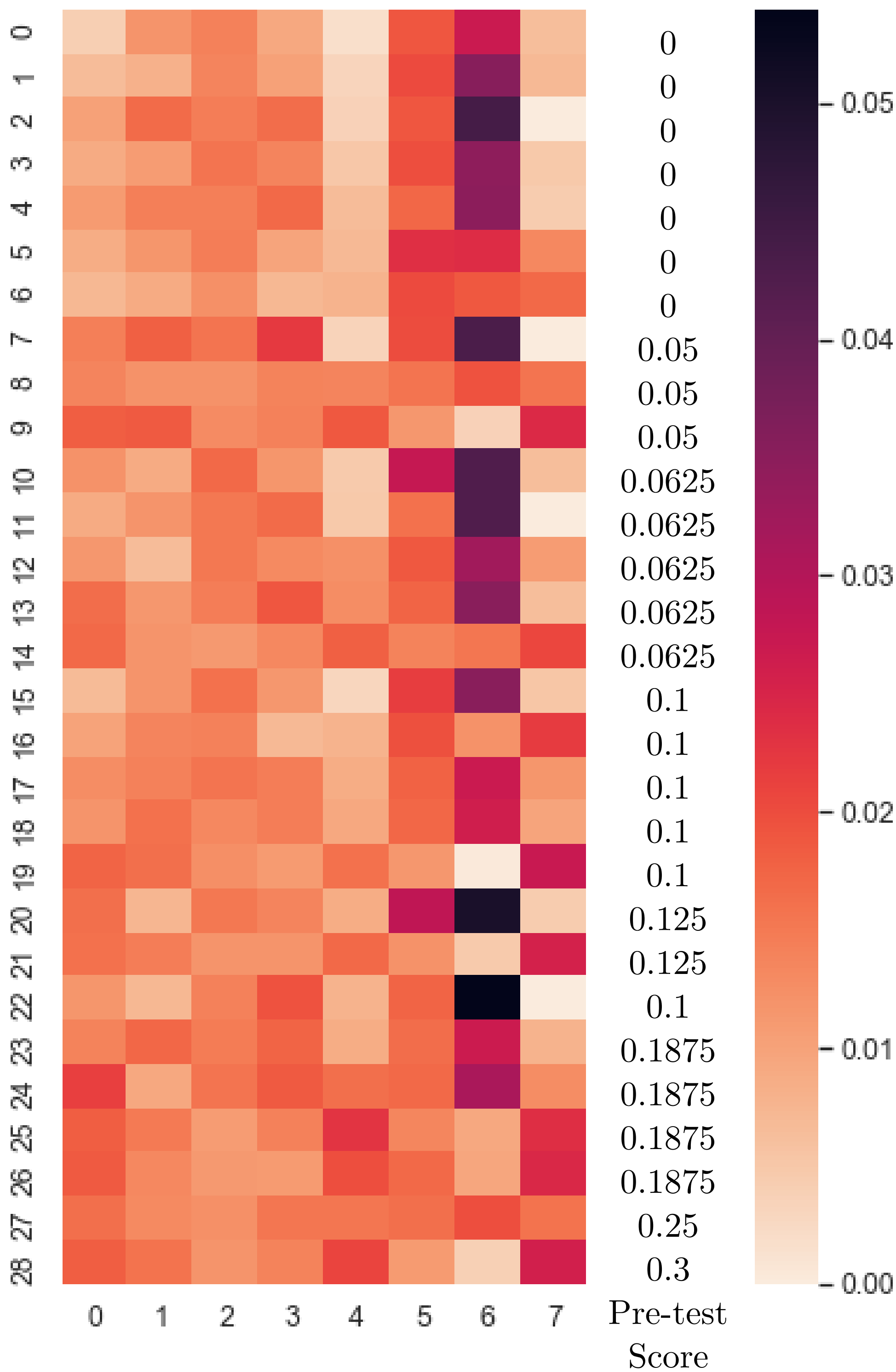


Figure 6.10: $H_{2,d}$ is a student-latent factor matrix and represents high performance students. The rows are ordered by students' pre-test scores. The latent factor 4 has a correlation with pre-test scores and its values increase with higher scores.

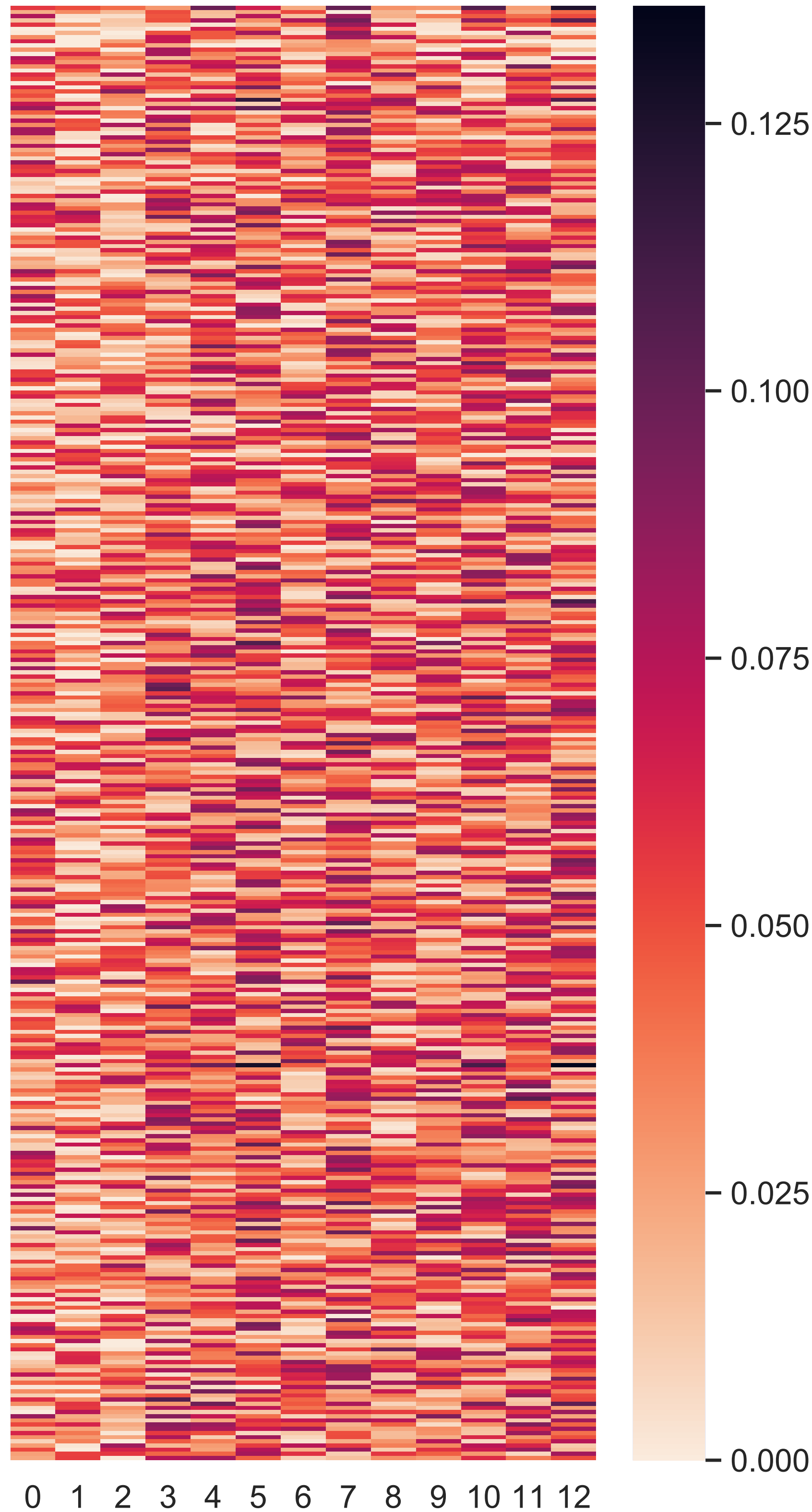


Figure 6.11: $H_{1,d}$ is the student-latent factor matrix and represents low performance students. The rows are ordered by students' post-test scores. The latent factor 12 has a correlation with post-test score and its values increase with higher scores.

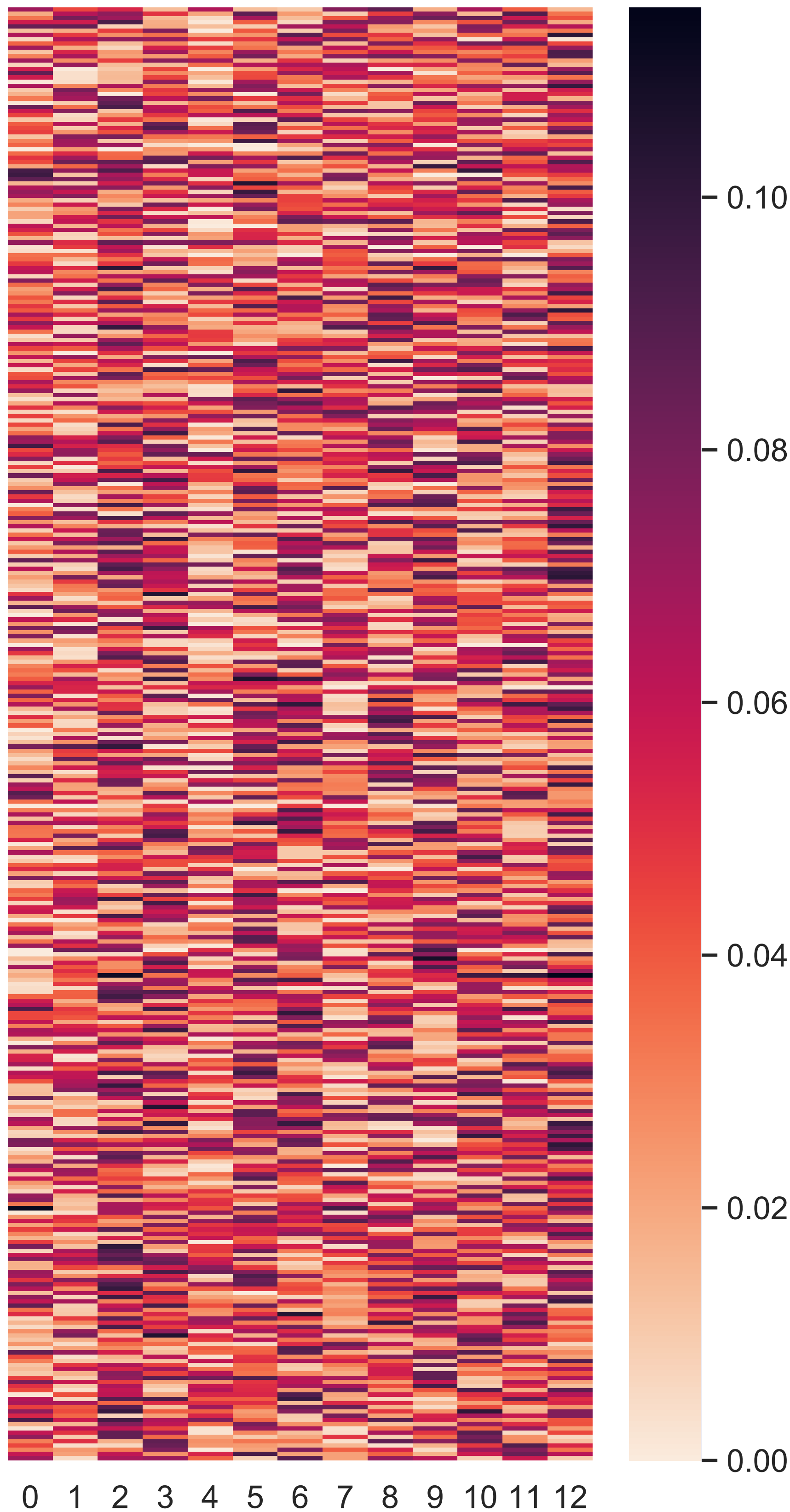


Figure 6.12: $H_{2,d}$ is the student-latent factor matrix and represents high performance students. The rows are ordered by students' pre-test scores. The latent factor 4 has a correlation with pre-test score and its values increase with higher scores.

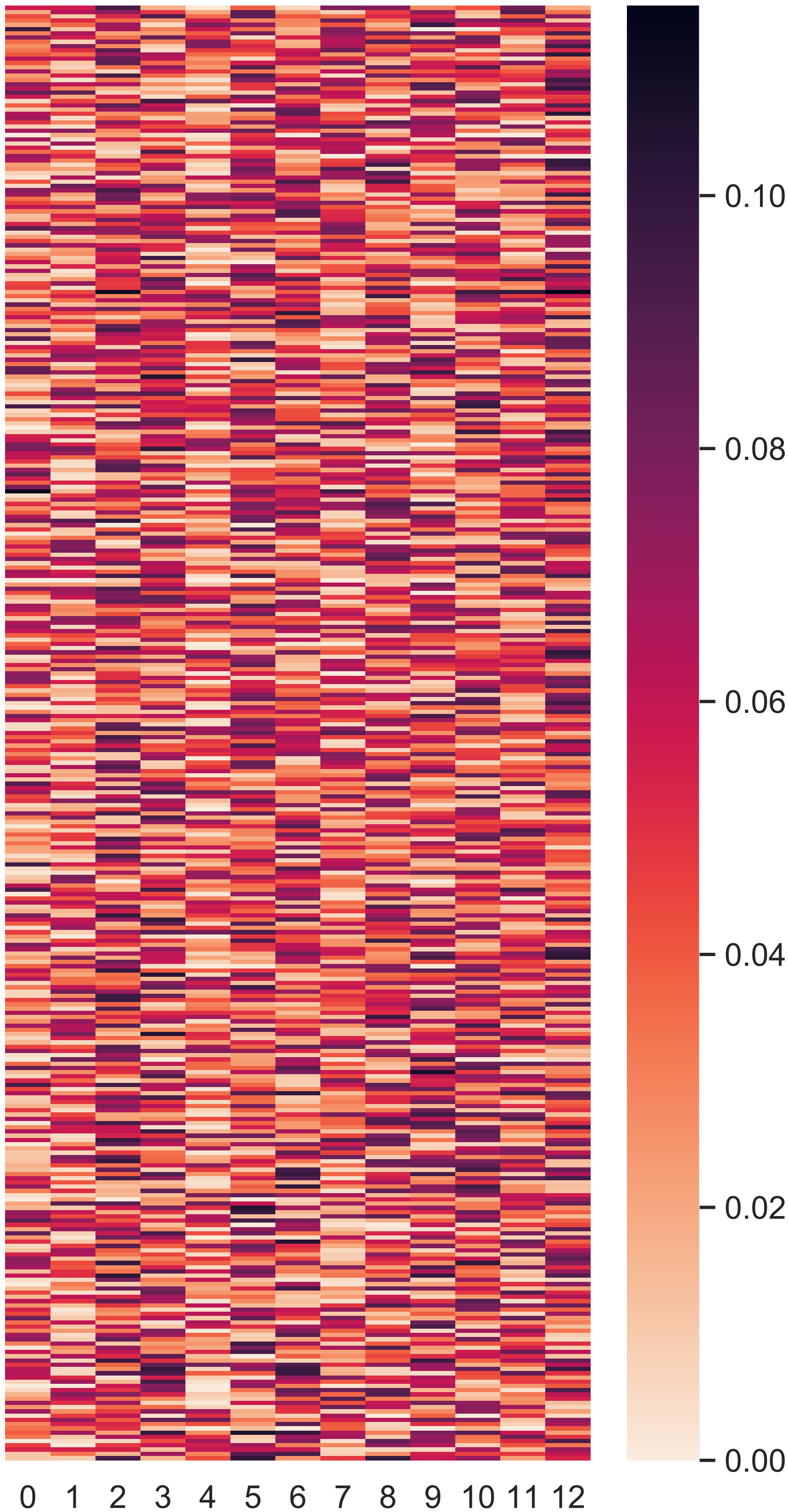


Figure 6.13: $H_{2,d}$ is the student-latent factor matrix and represents high performance students. The rows are ordered by students' post-test scores. The latent factor 7 has a correlation with post-test score and its values decrease with higher scores.

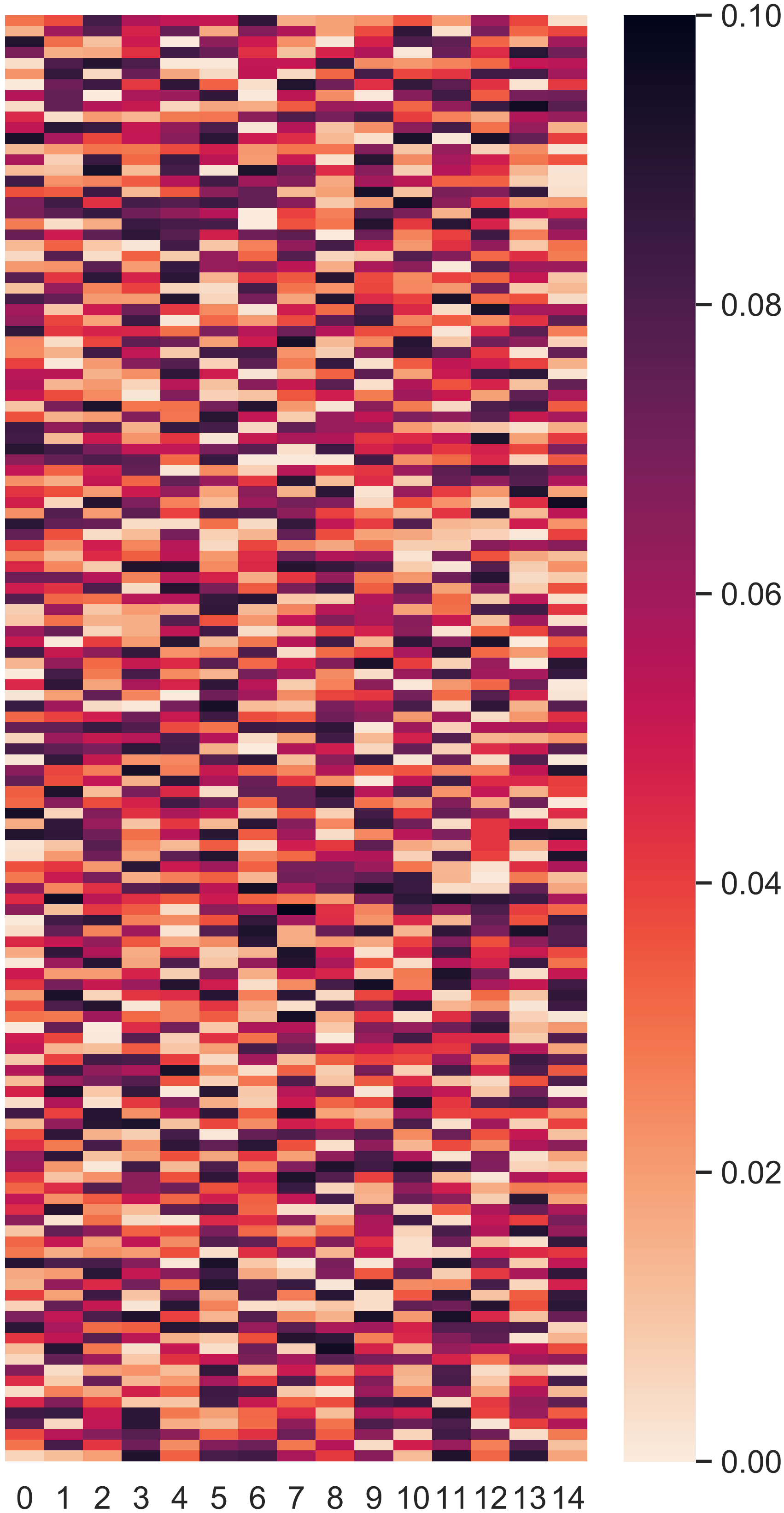


Figure 6.14: $H_{2,d}$ is the student-latent factor matrix and represents high performance students. The rows are ordered by students' post-test scores. The latent factor 4 has a correlation with post-test score and its values decrease with higher scores.

Latent	s	t	correlation	p-value
0	0.053709377	-0.810455248	-0.043529046	0.418235653
1	0.053735946	0.560468897	0.030117326	0.575522372
2	0.053634723	-1.273785882	-0.068319153	0.203594296
3	0.053740735	-0.502392203	-0.026998926	0.615711248
4	0.053718162	0.737201102	0.039601088	0.461499668
5	0.053716268	0.753591686	0.040480133	0.451606932
6	0.053737573	-0.54143385	-0.029095341	0.58855703
7	0.053756386	-0.225405611	-0.012116991	0.821796822
8	0.053759669	-0.092435857	-0.004969321	0.926405248
9	0.203373	1.062072142	0.057004507	0.288943669
10	0.200632	-0.195203341	-0.010493619	0.845348341
11	0.190663	0.66404707	0.035676665	0.50710244
12	0.053222909	2.65006192	**0.141044003****	0.008417217

Table 6.7: The correlation between latent factors and test score of low performance students. Latent factor 12 has positive correlation with post-test score in OLI Psychology.

Latent	s	t	correlation	p-value
0	0.054152952	-0.032549415	-0.001762647	0.974052953
1	0.054041637	-1.186292613	-0.064109195	0.236332637
2	0.053818764	2.061335891	**0.11093855***	0.040028665
3	0.054050141	1.139979299	0.061616042	0.255095002
4	0.053841834	1.988293158	**0.107053349***	0.047578402
5	0.054136426	0.457471335	0.024765863	0.647623629
6	0.054084385	0.930715819	0.050337193	0.352659009
7	0.053791211	-2.145433729	**-0.115405478****	0.032623945
8	0.054152544	-0.078761031	-0.00426511	0.937268898
9	0.054151553	-0.136686125	-0.007401766	0.891359559
10	0.05412715	0.571176519	0.030916157	0.568256233
11	0.053954007	1.587590781	0.085656884	0.113305673
12	0.053974245	-1.504290214	-0.081192928	0.133432268

Table 6.8: The correlation between latent factors and test score of high performance students. Latent factor 3 has negative correlation with pre-test score in OLI Psychology.

environments. Specifically, we built student micro-pattern vectors according to their interaction sequences with different learning material types, proposed SB-DNMF – a structure based discriminative matrix factorization model – to lean the common and distinct latent factors that explain the micro-patterns in high and low learning gain students, and introduced a new distance measure to calculate the structural similarities between micro-patterns. Our experiments showed that SB-DNMF provides a comparable fit vs. baselines to the data, finds meaningful trait and performance pattern clusters that represent efficient vs. inefficient patterns, and further detects meaningful asso-

Latent	s	t	correlation	p-value
0	0.053947267	-1.614401047	-0.087092524	0.107365362
1	0.054149623	-0.207343079	-0.011227549	0.835865698
2	0.054150251	0.187277022	0.010141098	0.851554817
3	0.054153017	0.015690847	0.000849707	0.987490206
4	0.054137286	-0.445471666	-0.024116627	0.656261604
5	0.053862891	-1.919283506	-0.103378158	0.055782779
6	0.054137516	0.44219659	0.023939425	0.658627274
7	0.053554827	-2.767761241	**-0.148226976***	0.00595244
8	0.054152446	-0.086184321	-0.004667092	0.931370476
9	0.054015572	1.31826632	0.07120691	0.188299299
10	0.054096717	0.842843066	0.045595043	0.399907002
11	0.05414896	0.226578029	0.012268965	0.820887638
12	0.053529454	-2.826855525	**-0.151320034****	0.004978133

Table 6.9: The correlation between latent factors and test score of high performance students. Latent factor 3 has negative correlation with post-test score in OLI Psychology.

ciations between the latent factors and students' post and pre-test scores. Our findings can be used to detect inefficient performance related behaviors in online learning systems and nudge students toward efficient patterns. Our experiments showed that efficient and inefficient patterns depend on the dataset. So one efficient pattern in one dataset could be inefficient in another one and vice versa.

Latent	s	t	correlation	p-value
0	0.085805185	-1.680143158	-0.144164994	0.095276828
1	0.086032161	1.451602909	0.124884535	0.148966818
2	0.086678448	0.316080459	0.027397363	0.752436937
3	0.086690816	-0.248860018	-0.021573878	0.803852948
4	0.085213738	-2.171370228	**-0.185030573****	0.031677471
5	0.08668395	0.28811168	0.024974659	0.773709747
6	0.086696924	0.20780098	0.018015706	0.835702226
7	0.085677949	1.796270627	0.153900783	0.074721391
8	0.086681422	-0.301284247	-0.026115747	0.763668281
9	0.086225821	1.225129576	0.105637803	0.222691661
10	0.086704492	0.1412664	0.012248431	0.887873238
11	0.085287216	2.116047987	**0.180471841***	0.036204292
12	0.086049774	-1.432427117	-0.12326003	0.154367783
13	0.086702861	0.15799573	0.013698682	0.874699794
14	0.086058643	1.422677838	0.122433725	0.157170804

Table 6.10: The correlation between latent factors and test score of high performance students. Latent factor 3 has negative correlation with post-test score in OLI Statistics.

CHAPTER 7

Matrix Factorization for Performance Prediction

7.1 Introduction

Student performance prediction is a crucial task in educational data mining. Cen et. al. demonstrated that an improved model which is able to predict students' performance, can save students' and educators' time and resources [14]. If students are aware of their performance in a course, they can utilize that and work on the other fields that they need to practice on. On the other hand, educators are interested in assessing the students to assist them in their learning. So predicting student performance is very beneficial.

To address the student performance prediction problem, we use the techniques used in recommendation systems. Matrix factorization (MF) is a collaborative filtering algorithm used in recommendation systems that decomposes a matrix into the product of two matrices with lower dimensions. In this chapter, first, we use k-nearest neighbors (KNN) algorithm on the original space to classify students into low-performance and high-performance groups. Then we propose 5 matrix factorization approaches to decompose pattern-vector matrix in order to predict students' performance. The goal of this chapter is to answer the fourth research question in 1.2:

Could detecting efficient and inefficient patterns help us to better predict students' performance?

7.2 Performance Prediction: Problem Formulation

We focus on predicting students' performance based upon their activity sequences. Thus, we attempt to determine whether students' pattern vectors including micro-patterns would be helpful for performance prediction or not. To do this, we propose two approaches that uses the original space and the latent factor space as features for classification. So the problem is defined as: Given students pattern vectors, our goal is to classify students as low-performers or high-performers. We use learning gain as an indicator of students' performance.

For prediction, first we split the data to train and test sets. We do this in 5 different settings.

The percentage of the students used as test set are 10%, 20%, 30%, 40% and 50% of the students. These splits show how many students are enough for training. In each of the settings, we use a percentage of the students' sequences for testing. These percentages are from 10% to 90%. It means that for example we use the first 10% of a student sequence to built a vector and use it for prediction. It shows how many attempts we need to accurately classify test students, given that $\%x$ of data is used for training. Since we normalize the pattern vectors, the effect of using shorter sequences is minimal. As we showed that the pattern vectors are stable for students, using for example 10% of the data for testing is reasonable since it shows the behavior of the student.

7.2.1 KNN on Original Space

In this secion, we apply KNN on students' original space. We use students' feature vectors (pattern-vector) as input to KNN and output is their performance. Formally, let X_i be a student pattern vector with p features $(x_{i1}, x_{i2}, ..., x_{ip})$. p is the total number of patterns. We use KNN to predict the class that student X_i belongs to. We perform 4-fold cross validation on the datasets and report the average of the results. The performance results of classification on Mastery Grids are shown in Table A.1 to Table A.5, on OLI Psychology are shown in Table A.6 to Table A.10 and on OLI Statistics are shown in Table A.11 to Table A.15. In this experiment on Mastery Grids, sequences less than 50% of the original sequences are removed because those were too short and did not have enough patterns. The highest accuracy for Mastery Grids occurs when the training set is 80% of the students and 70% of the sequences of test students are used for prediction. In this case, the accuracy of prediction is 66.11%. The highest accuracy for OLI Psychology occurs when the training set is 90% of the students and 70% of the sequences of test students are used for prediction. In this case, the accuracy of prediction is 61.68%. The highest accuracy for OLI Statistics occurs when the training set is 90% of the students and 90% of the sequences of test students are used for prediction. In this case, the accuracy of prediction is 90.56%. The results demonstrate that we are able to predict students performance using activity sequences.

7.2.2 Discriminative Latent Factors

In this section, we introduce a model to decompose pattern-student matrices and predict students' performance based upon discriminative latent factors. In order to use the latent factors in matrix H_i^T (student-latent factor) for prediction, we concentrate the data in this matrix. The proposed model, have constraints that enforces the average of matrix H_i^T to be similar to the average of the rows.

$$
\begin{aligned}
L = \; & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \lambda(\|H_1^T - \mu_{H_1}^T\|_F^2 + \|H_2^T - \mu_{H_1}^T\|_F^2) - \epsilon\|\frac{1}{n_1} H_{1,d}^T e - \frac{1}{n_2} H_{2,d}^T e\|_F^2 + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned} \tag{7.1}
$$

Here $\mu_{H_i}^T = \frac{\Sigma H_i^T}{n_{H_i}}$ is the average of H_i over students. Particularly, each row is an average vector that is repeated for students. The term $\|H_i^T - \mu_{H_i}^T\|_F^2$ imposes the constraint that compresses the matrix H_i so the information in each of this matrix is translated into a vector. The other terms in the cost function impose the similarity among common patterns ($\|W_{1,c} - W_{2,c}\|_F^2$) and dissimilarity among patterns ($\|W_{1,d}^T W_{2,d}\|_F^2$). Then we use the vectors representing H_1 and H_2 for prediction. Formally, we apply:

$$
\begin{aligned}
V_1 &= V_{test}.W_1 - \mu(H_{1,d}) \\
V_2 &= V_{test}.W_2 - \mu(H_{2,d})
\end{aligned} \tag{7.2}
$$

Here V_{test} is a student pattern vector from the test set. $\mu(H_{1,d})$ and $\mu(H_{2,d})$ are the vectors representing column-wise average of the $H_{1,d}$ and $H_{2,d}$. So these vectors indicate the average weight of discriminative latent factors over students. In fact V_1 and V_2 are the distance vectors from low-performance latent factors (H_1) and high-performance latent factors (H_2). If the distance is closer to the high-performers the student is classified as a high-performance student and if the distance is closer to the low-performance students, they are low-performance students. To do the classification, we follow the same setting in previous section for splitting the dataset to train set and test set. We apply Gradient Descent algorithm to minimize the cost function, use a validation set to find the optimized hyper-parameters and run 4-fold cross validation. The results are shown in Tables A.20 to A.30.

The best accuracy for Mastery Grids is 59.02%. It occurs when we use 80% of the students for training and 20% for testing and 70% of the test sequences are used. It shows that not necessarily more data is needed for prediction. This could happen because the behavior of the students might change at the end of the course. The other reason is that students study more before the exams and the effect of behavior will decrease at those times. For OLI Psychology, the best result is when the training set is 90% and test set is 10% of the students. Here more training data is needed, however, the higher accuracy in each run happens when the short sequences like 30% or 40% of them are used as test set. In OLI Statistics the highest accuracy occurs at 70% of students as training set and 30% as test set. In this dataset, prediction with longer sequences have higher accuracy.

7.2.3 Multi-View Model

Zhang et al. in [99] proposed a model for classification by using relationship among multiple views. In that work, they explored discriminative and non-discriminative information among different views via joint non-negative matrix factorization and used discriminative part for classification. We propose a model that uses the labels in a similar way to decompose the input pattern-student matrices into common and discriminative parts. The model imposes the supervised constraints to the input matrices. Specifically the objective function is:

$$\begin{aligned} L = \ & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\ & \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \\ & (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2) \end{aligned} \tag{7.3}$$

Here Y_i is the matrix having students' performance labels and matrix $B = [B_{c_i} B_{d_i}]$ is a projection matrix that maps the latent factors into label space. The second line in Equation (7.3) imposes this constraint. So the input matrices X_1 and X_2 are factorized into common and discriminative latent factors.

For Mastery Grids, the highest accuracy is $\%60.86$ with $\%90$ of students as training set and $\%10$ as test set and $\%90$ of the sequences length is used for testing. This is reasonable, since more data is used for training and the accuracy is higher in this case. For OLI Psychology, the accuracy is around $\%50$ and it seems that this model is not working well on this dataset. For OLI Statistics when the sequences are shorter, the accuracy is higher.

7.2.4 Multi-View and Discriminative Latent Factors

Another model that is used for classification implies the constraints from previous sections simultaneously. It means to have the constraint on discriminative parts and using the labels at the same time. So we formulate the object function as:

$$\begin{aligned}
L = \; & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \\
& \lambda(\|H_1^T - \frac{1}{n_1} H_1^T e e^T\|_F^2 + \|H_2^T - \frac{1}{n_2} H_2^T e e^T\|_F^2) - \epsilon\|\frac{1}{n_1} H_{1,d}^T e - \frac{1}{n_2} H_{2,d}^T e\|_F^2 + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned} \tag{7.4}$$

For Mastery Grids, the highest accuracy is 59.54% and with 70% and 80% of data as training set and longer sequences as test have better results. In OLI Psychology the accuracy is around 50% in most cases and the prediction is not working well. In OLI Statistics the accuracy increases with higher amount of data as training set. The overall results are not consistent and there is not a trend in the accuracy of the model based on the size of the train and test sets

7.2.5 Multi-View and Similarity

Another model is to imply constraints in multi-view model and use patterns' structures. So we combine the constraints from these two models to build a new model.

$$\begin{aligned}
L = \; & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \epsilon\|S - \varepsilon W W^T\|_F^2 + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned} \tag{7.5}$$

For Mastery Grids, as the training size increases, the sequences that are shorter have better accuracy. The best case happens when the training set is 80% and 60% of the sequences are used as test. In this model, the accuracy for OLI Psychology is around 50% and there is not any pattern in the results. In OLI Statistics, the best accuracy is with 50% training set and 90% of sequence length.

7.2.6 Multi-View, Similarity and Discriminative Latent Factors

The last model is presented by consolidating previous constraints in one model. It means to use the supervised constraints, the patterns' structures and the discriminative latent factors in one model.

$$\begin{aligned} L = \ & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\ & \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \epsilon\|S - \varepsilon W W^T\|_F^2 + \\ & \lambda(\|H_1^T - \frac{1}{n_1} H_1^T e e^T\|_F^2 + \|H_2^T - \frac{1}{n_2} H_2^T e e^T\|_F^2) - \epsilon\|\frac{1}{n_1} H_{1,d}^T e - \frac{1}{n_2} H_{2,d}^T e\|_F^2 + \\ & (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2) \end{aligned} \tag{7.6}$$

The highest accuracy in Mastery Grids is when 70% of the data is used for training and 90% of sequences' length are used as test. In this case the accuracy is 61.58%. In OLI Psychology, the highest accuracy is 53.47% which is slightly better 50%. In OLI Statistics, 39.66% is the highest accuracy. The model classifies most of the instances as positive and that is why the prediction is biased.

The accuracy of different methods are demonstrated in Figures 7.1 to 7.15. In most cases, the KNN has a better prediction accuracy. The results show that the amount of data used for training does not have an important role in prediction. The other conclusion is that there is not a relation with the length of the sequences and the prediction accuracy. The number of constraints that the models have is high and causes the distortion of the data. Each term applies a different constraint but increasing the constraints diminishes the impact of them.

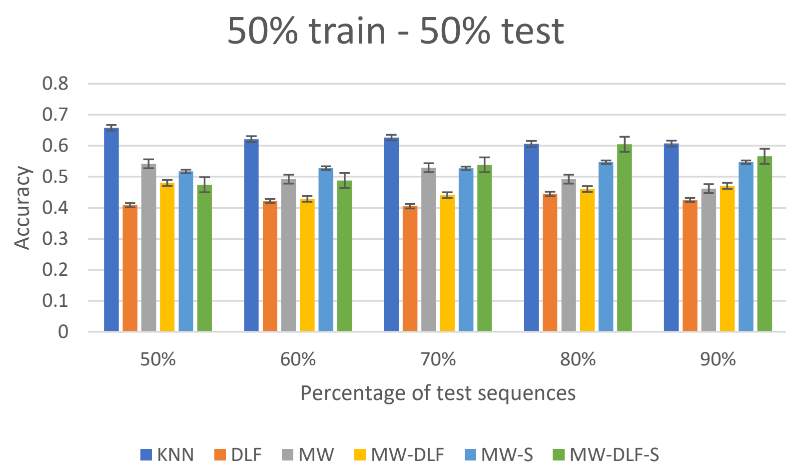


Figure 7.1: Classification Accuracy on Mastery grids for 50% train and 50% test

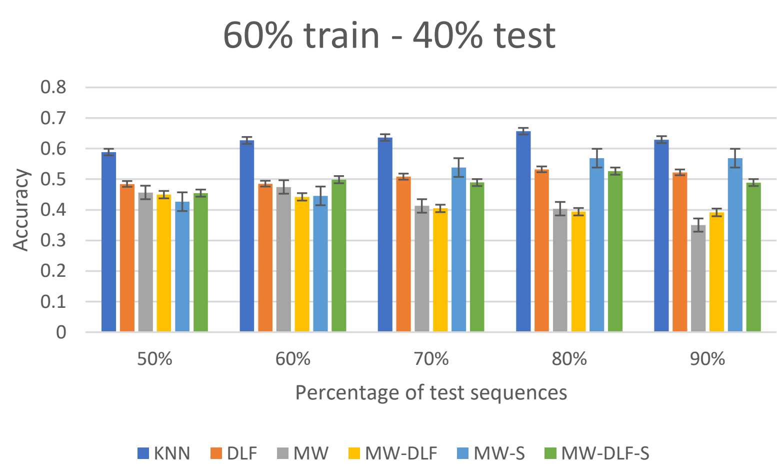


Figure 7.2: Classification Accuracy on Mastery grids for 60% train and 40% test

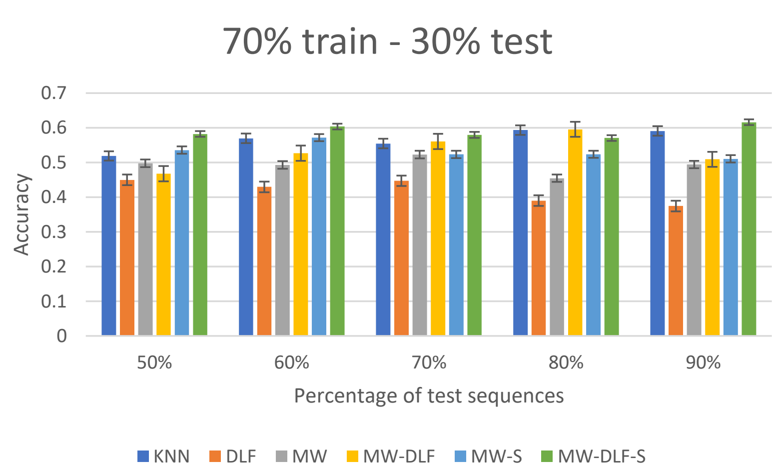


Figure 7.3: Classification Accuracy on Mastery grids for 70% train and 30% test

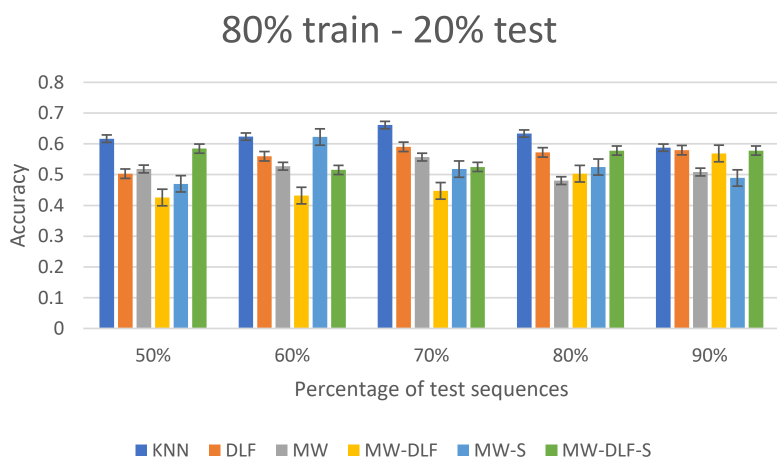


Figure 7.4: Classification Accuracy on Mastery grids for 80% train and 20% test

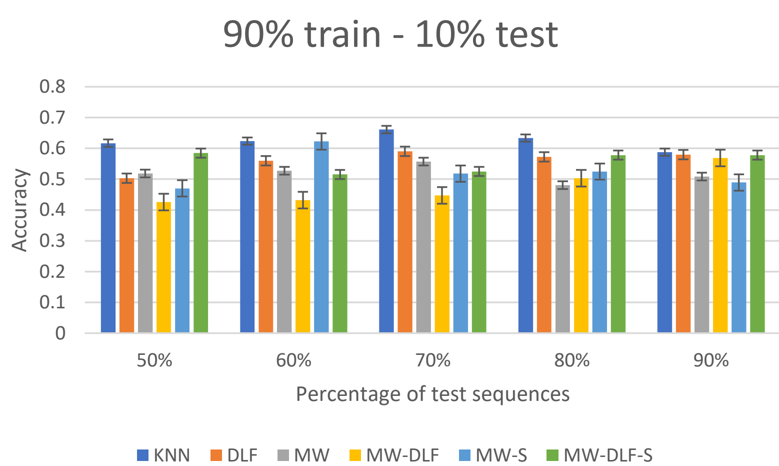


Figure 7.5: Classification Accuracy on Mastery grids for 90% train and 10% test

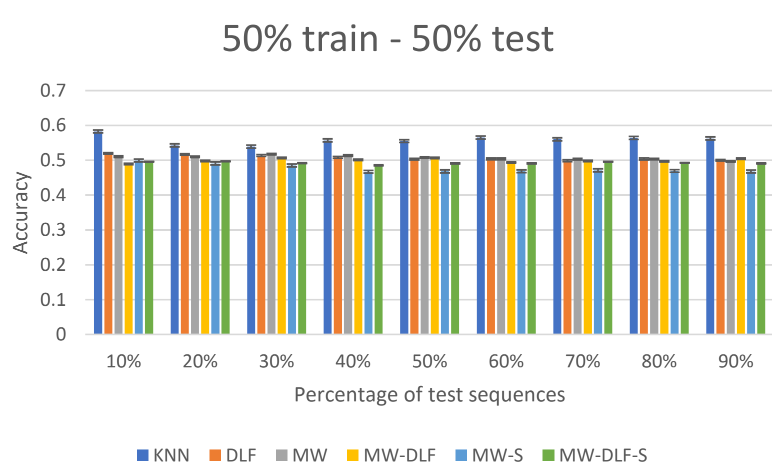


Figure 7.6: Classification Accuracy on OLI Psychology for 50% train and 50% test

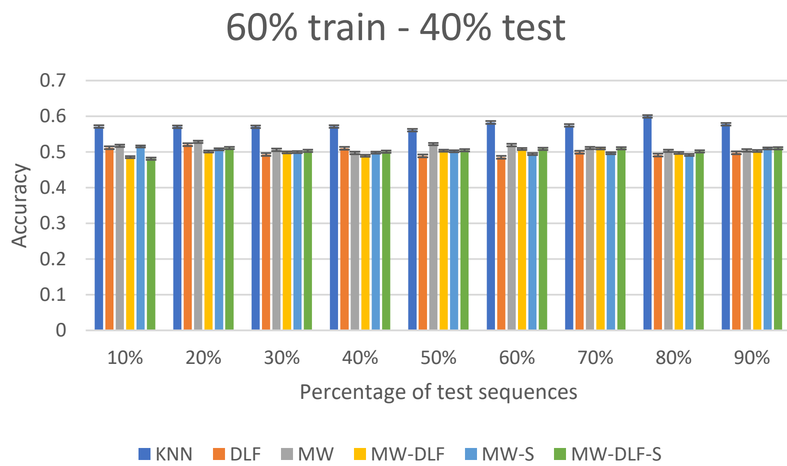


Figure 7.7: Classification Accuracy on OLI Psychology for 60% train and 40% test

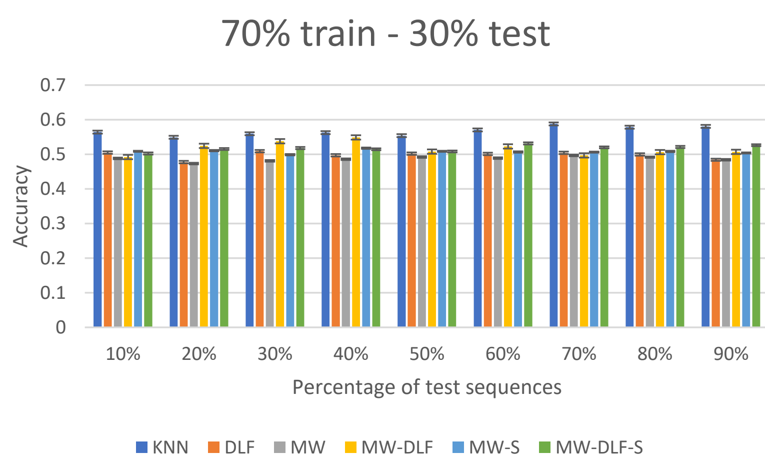


Figure 7.8: Classification Accuracy on OLI Psychology for 70% train and 30% test

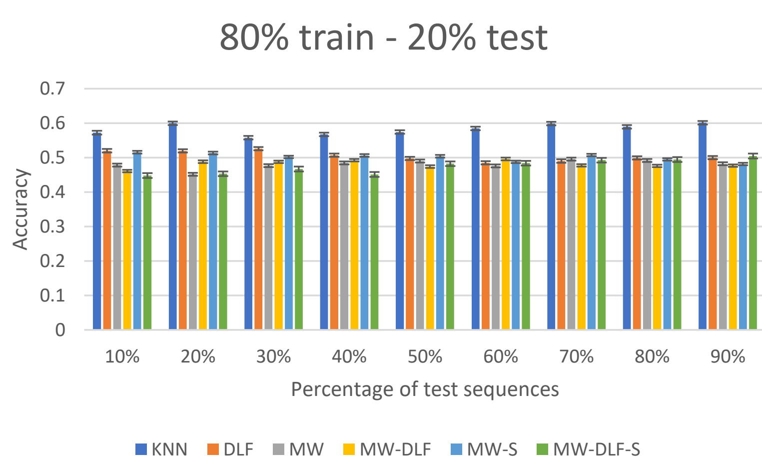


Figure 7.9: Classification Accuracy on OLI Psychology for 80% train and 20% test

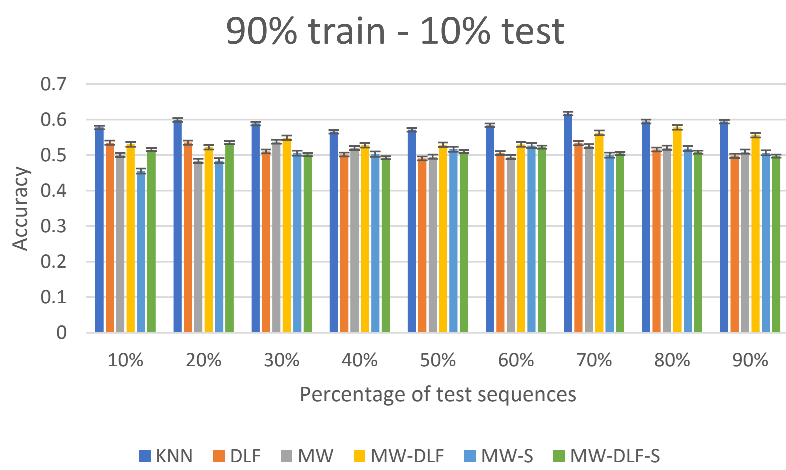


Figure 7.10: Classification Accuracy on OLI Psychology for 90% train and 10% test

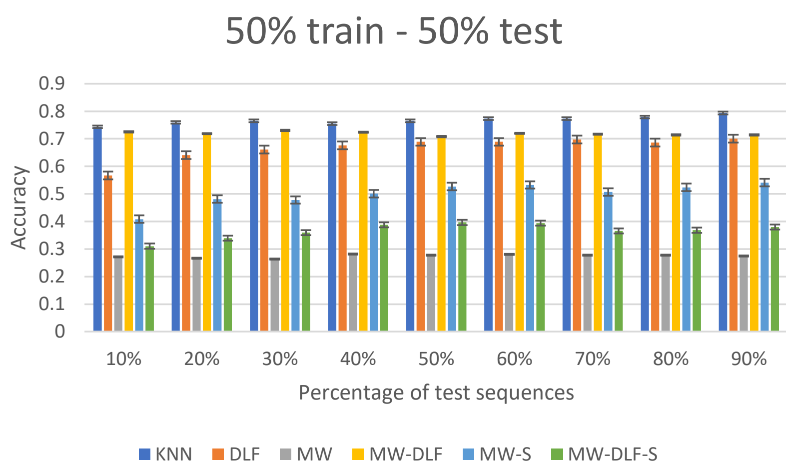


Figure 7.11: Classification Accuracy on OLI Statistics for 50% train and 50% test

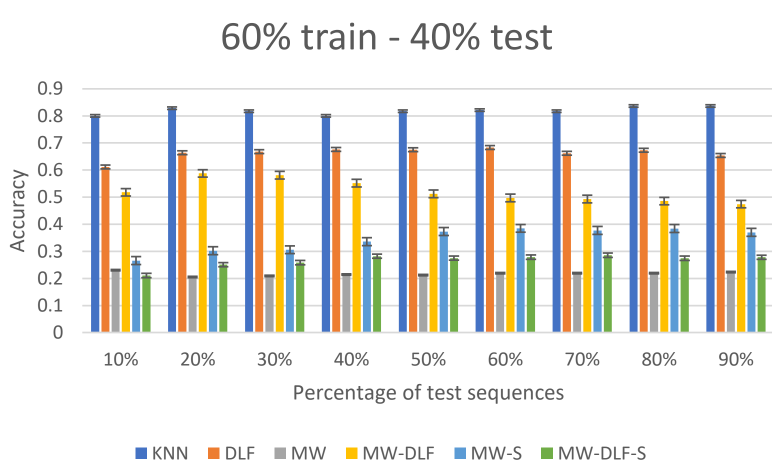


Figure 7.12: Classification Accuracy on OLI Statistics for 60% train and 40% test

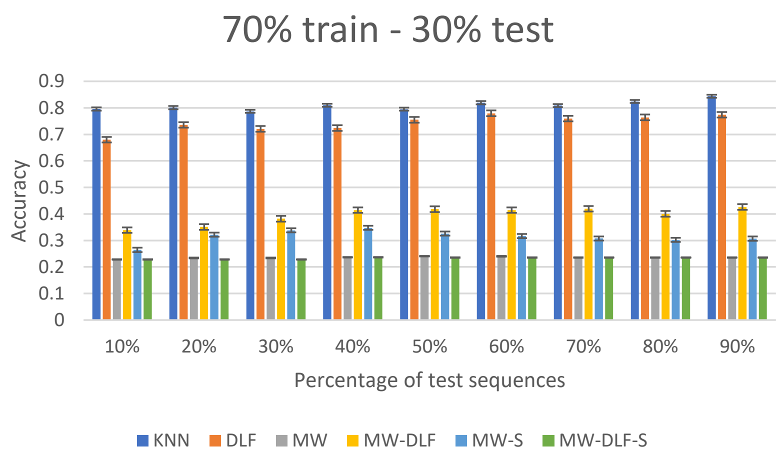


Figure 7.13: Classification Accuracy on OLI Statistics for 70% train and 30% test

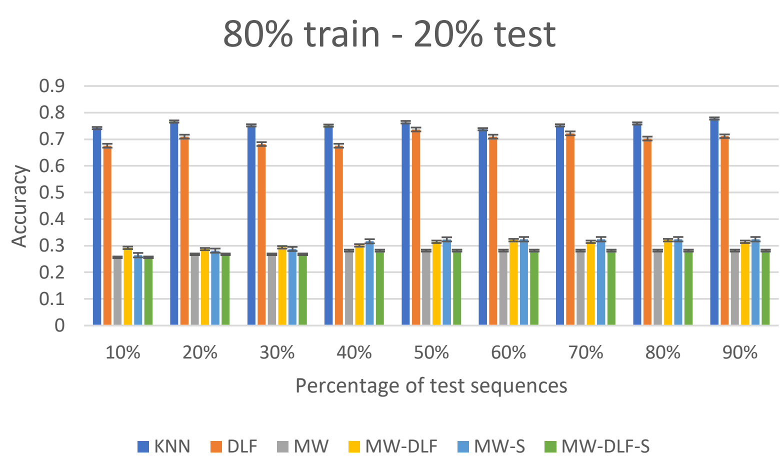


Figure 7.14: Classification Accuracy on OLI Statistics for 80% train and 20% test

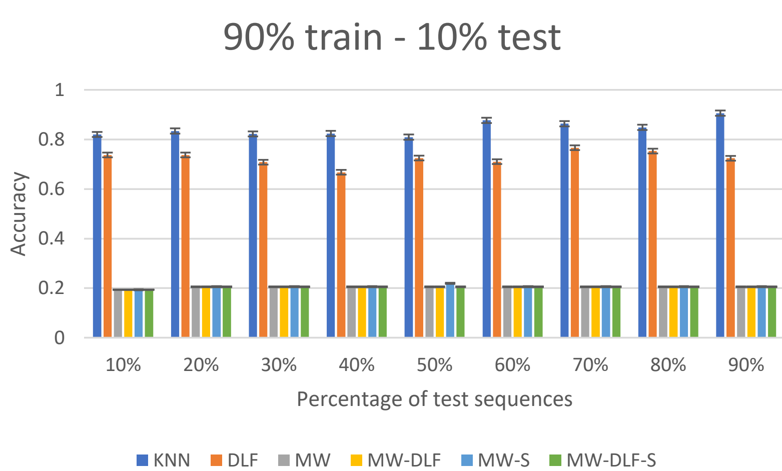


Figure 7.15: Classification Accuracy on OLI Statistics for 90% train and 10% test

7.3 Conclusion

Having students' behavioral patterns, we proposed classifiers to predict their performance and answer the fourth research question. Our experiments show that the pattern vectors that we derived from students' sequences are associated with students' performance. We performed the classification with different supervised and unsupervised methods on original and latent factors spaces of students' pattern vectors. The classification results from the original space outperformed the latent factors spaces. The models that use latent factors need to be investigated more. Another reason for low accuracy could be the over-processing of the data. The methods that we have used to represent students' sequences might have distorted the information that despite finding the relation between students' performance and the behavior, the performance prediction is not accurate.

Although the factors that impact students' grades are not limited to their behaviors, we demonstrated that to some extent the behaviors while having interaction with an online environment are associated with students' performance. Because these patterns take into account both types of interaction and the time spent on them. Generally, students' performance prediction is a very challenging task and we still need to incorporate more factors to improve the accuracy of the prediction.

CHAPTER 8

Conclusion

In this dissertation, we investigated students' behavior patterns while working with learning materials such as problems and examples in online learning platforms. We found frequent patterns of students' behavior and used them to construct behavior vectors that represent the duration and type of the students' activities. Our experiments showed that using pattern vectors, we can find the clusters of students with similar behaviors. In these groups, we distinguished students' efficient and inefficient patterns by comparing the frequency of the patterns used by low-performance and high-performance students. Then we proposed a framework to discriminate learning traits and performance traits from the pattern vectors. We demonstrated that using discriminative non-negative matrix factorization, we can detect pattern clusters that are associated with students' performance. In other words, we found the patterns that are used specifically by low-performance and high-performance students. The patterns that are used by low-performance students are inefficient and the patterns used by high-performance students are efficient. Since the patterns are micro-sequences of students' activities, we can use their structures to improve the model to discriminate among patterns more accurately. To accommodate the structure of the patterns, we proposed a new distance measure that considers the duration and the type of activity in their comparison. So we proposed SB-DNMF (Structure-based discriminative non-negative matrix factorization) that incorporates the similarity of the patterns to the framework. The experiments on 3 real-world datasets showed that our model distinguishes the learning traits from performance traits successfully. Our findings can be utilized to identify inefficient patterns and nudge students to avoid them and use efficient patterns. After finding the pattern, we used classification to predict students' performance from their patterns. Our experiments demonstrated that using sequential patterns, we can predict students' performance. In spite of that, the prediction accuracy is not remarkable since there are so many factors that affect students' performance and are not involved in the prediction model.

8.1 Limitations

The ideal dataset that we could use in our work, should be comprehensive and track learning materials that students use because students are free to study material out of the system and there are other factors that impact students' learning. The designed system should be able to provide enough materials for students to decrease the effect of other factors.

The datasets such as Mastery Grids that we used are designed for self-study and using it was not mandatory. This causes some students to play with the system and have abnormal behavior. That's why there are very long sequences that some students have. However, to have a robust performance analysis and prediction, the system should be mandatory and capable of identifying those unusual behaviors. One possible solution to remove gaming behavior is to define a threshold and remove the sequences longer than that. Another one is to remove the attempts with short spent time. Presumably, such activities are considered as gaming the system.

Another limitation of this work is in the section 6.2. The distance measure finds the distance between micro-patterns and takes into account the activity type and time spent on it. However, some of the micro-patterns contain "_" that splits the patterns based upon topics. This character is treated like as an attempt. But the distance measure should consider it differently to show the effect of changing a topic by giving it proper weight.

8.2 Discussion

Having a dataset dedicated to this work is essential. Not all educational datasets allow multiple attempts. So this work is not generalizable for any type of datasets. However, if we could design and develop an interface that has the properties needed and collect the data from students, we were able to perform behavior analysis across courses to verify if the students have similar behavior in different courses or not. But because of the limitations, this task is future work.

An improvement to this work could be giving weight to the activities based upon the difficulty. It means to have different weights for different problems and examples. It requires having various labels for activities. This way the amount of time and complexity of an attempt is taken into account so the behaviors are represented more accurately. However, the patterns will be more complex and it makes interpretation harder.

8.3 Future Work

In this dissertation, we applied the proposed model on students with low-performance and high-performance to compare the performance in extreme cases and removed the students with average performance. In the future, we can add other constraints to incorporate middle group students. The model should split the discovered latent factors to a set of common patterns and a set of discriminative patterns for each group. While performing lower-rank approximations on the input matrices, we can add constraints to enforce common patterns to be as similar as possible among all students and distinct patterns to be as different as each other.

Another future work is to find students' patterns across multiple courses. In current work, the patterns in one course are detected. Having a dataset that contains the sequences of activities in multiple courses, we can discover the students' behavior in different courses and compare them to see if the students use the same set of patterns in all courses or these are course-specific.

An application of this research is to early prediction of students' performance and carry out proper intervention to avoid students from failure in the course. The intervention seems necessary but it will change the students' behavior since they will be avoided using inefficient patterns. So the training set should be adopted with the new data. One possible solution is to discard students' activity sequences before the intervention.

Another interesting addition to this work is applying deep learning to improve classification accuracy. A possible way is to combine the datasets to have more data for training. Neural networks and deep learning are prevalently used with sequential data. These methods deal with sequences of arbitrary length. However, one downside of using more data in this work is the lack of homogeneity. It means that to have more data, we need to join students from different classes with various conditions such as instructors and the environment. This condition should be taken into account in case of using deep learning methods with more data.

The experiments showed that the proposed methods are generalizable if the dataset contains the requirement: Allowing students to have multiple attempts. Applying proposed methods on the different datasets demonstrated that in each dataset we can find the behaviors that are specific for a dataset. Some of the behaviors were similar among datasets and some were different. But in each dataset meaningful behaviors were detected. Moreover, the hyperparameter tuning showed that they are depending on the dataset, and based on the distribution of the data, the parameters differ.

APPENDIX A

Appendix

$$
\begin{aligned}
L = \ & \gamma(\|X_1 - W_1H_1^T\|_F^2 + \|X_2 - W_2H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned} \tag{A.1}
$$

$$
\begin{aligned}
L = \ & \gamma(\|X_1 - W_1H_1^T\|_F^2 + \|X_2 - W_2H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \\
& \lambda(\|H_1^T - \frac{1}{n_1} H_1^T ee^T\|_F^2 + \|H_2^T - \frac{1}{n_2} H_2^T ee^T\|_F^2) - \epsilon\|\frac{1}{n_1} H_{1,d}^T e - \frac{1}{n_2} H_{2,d}^T e\|_F^2 + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned} \tag{A.2}
$$

	50%	60%	70%	80%	90%
TP	1.75	1.25	2	1.5	1.25
TN	1.75	2.25	2	2.25	2.25
FP	0.25	0	0.25	0	0
FN	2.25	2.75	2	2.5	2.75
Precision	0.875	1	0.950	1	0.750
Recall	0.512	0.337	0.500	0.400	0.275
Accuracy	64.2%	57.7%	**64.8%**	61.3%	52.9%

Table A.1: Classification Results with KNN on Original Space - Mastery Grids - 90% train 10% test

	50%	60%	70%	80%	90%
TP	2	2	2	2	1.5
TN	4.5	4.75	5.25	5	5
FP	0.5	0.5	0	0.25	0.25
FN	3.25	3.25	3.5	3.5	4
Precision	0.900	0.900	1	0.937	0.916
Recall	0.398	0.398	0.371	0.385	0.285
Accuracy	61.6%	62.3%	**66.1%**	63.3%	58.7%

Table A.2: Classification Results with KNN on Original Space - Mastery Grids - 20% train 80% test

	50%	60%	70%	80%	90%
TP	2	2	1.75	2	2.75
TN	6.25	7.25	7.5	8.25	7.5
FP	2.5	1.75	1.5	1.25	2
FN	5.25	5.25	6	5.75	5
Precision	0.516	0.625	0.625	0.562	0.572
Recall	0.236	0.236	0.224	0.220	0.343
Accuracy	51.8%	56.9%	55.4%	**59.3%**	59.0%

Table A.3: Classification Results with KNN on Original Space - Mastery Grids - 70% train 30% test

	50%	60%	70%	80%	90%
TP	1.25	2.25	2.5	3	2.5
TN	10.25	10.5	10.75	11	11
FP	0.5	1	0.75	1	1
FN	7.75	6.75	7	6.5	7
Precision	0.833	0.750	0.762	0.750	0.750
Recall	0.147	0.260	0.263	0.316	0.260
Accuracy	58.8%	62.6%	63.6%	**65.6%**	62.9%

Table A.4: Classification Results with KNN on Original Space - Mastery Grids - 60% train 40% test

	50%	60%	70%	80%	90%
TP	4.75	4.5	5.5	4.25	4
TN	11.25	11	10.75	11.5	11.75
FP	1.5	2.25	2.5	1.75	1.5
FN	7	7.25	7.25	8.5	8.75
Precision	0.796	0.682	0.746	0.761	0.844
Recall	0.431	0.386	0.456	0.341	0.316
Accuracy	**65.7%**	62.0%	62.6%	60.6%	60.7%

Table A.5: Classification Results with KNN on Original Space - Mastery Grids - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	14.5	15.75	18.75	17.25	16	20.25	21	17.5	19.5
TN	22.25	23.75	20.5	20.75	22.5	19.5	21	23	21
FP	10.75	9.5	13	13	11.25	14.25	12.75	10.75	12.75
FN	16.25	16.75	14.25	16	17.5	14	13.25	16.75	14.75
Precision	0.573	0.630	0.588	0.5693	0.581	0.586	0.625	0.619	0.607
Recall	0.474	0.485	0.568	0.519	0.478	0.591	0.613	0.511	0.569
Accuracy	57.7%	59.9%	58.8%	56.5%	57.1%	58.3%	**61.6%**	59.4%	59.4%

Table A.6: Classification Results with KNN on Original Space - OLI Psychology - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	31.5	39.75	38	37	37.75	40.5	40.5	39	41.5
TN	41.75	39.5	37	40.25	40.75	40	42	42.25	41.5
FP	18.5	22	26.25	23.75	23.25	24.25	22.25	22	22.75
FN	36.25	31	33.25	35.25	35	33	33	34.5	32.25
Precision	0.630	0.649	0.592	0.611	0.623	0.625	0.645	0.641	0.651
Recall	0.466	0.564	0.534	0.511	0.520	0.551	0.551	0.532	0.564
Accuracy	57.2%	59.9%	55.7%	56.7%	57.4%	58.4%	59.8%	58.9%	**60.1%**

Table A.7: Classification Results with KNN on Original Space - OLI Psychology - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	42.25	44.25	54.5	53.25	50.5	58.5	56	54	55.75
TN	66.75	65	59	62.75	63.75	60.25	66.75	66.75	65.5
FP	29.25	32.25	40	37	36	39.75	33.5	33.5	34.75
FN	54.75	57.5	49.25	53	56	49.5	52.5	54.5	53
Precision	0.597	0.574	0.578	0.592	0.584	0.595	0.627	0.617	0.616
Recall	0.439	0.432	0.528	0.504	0.475	0.543	0.519	0.499	0.516
Accuracy	56.4%	54.8%	55.9%	56.2%	55.3%	57.0%	**58.7%**	57.8%	58.0%

Table A.8: Classification Results with KNN on Original Space - OLI Psychology - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	57.25	66.25	64.75	66	65.25	70.5	66.5	77.25	68.25
TN	89.5	85.5	89.75	90.5	88.5	89.75	92	88.75	91.75
FP	39.75	47	44.75	45.25	47.25	46.5	44.75	48	45
FN	70.5	67.25	71.5	72.25	73	68.25	72.75	62.75	72
Precision	0.591	0.586	0.598	0.594	0.584	0.602	0.595	0.616	0.603
Recall	0.448	0.496	0.475	0.477	0.471	0.507	0.475	0.551	0.486
Accuracy	57.0%	57.0%	57.0%	57.1%	56.1%	58.2%	57.4%	**59.9%**	57.7%

Table A.9: Classification Results with KNN on Original Space - OLI Psychology - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	65.25	71	82.25	80.5	88.25	89	68	73.5	94.25
TN	121.7	109.2	100.2	111	103.2	106.7	127	123.2	102
FP	43.5	59.5	71.25	62.5	70.25	67	47.5	51.25	72.5
FN	90.5	92.25	84.75	90	83.25	83.75	105.5	100.7	80.25
Precision	0.603	0.538	0.537	0.574	0.558	0.569	0.590	0.587	0.567
Recall	0.418	0.434	0.491	0.475	0.514	0.514	0.391	0.420	0.539
Accuracy	**58.2%**	54.2%	53.9%	55.6%	55.5%	56.5%	56.0%	56.4%	56.2%

Table A.10: Classification Results with KNN on Original Space - OLI Psychology - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	0.5	1	1	0.75	0.25	1.5	1.5	1.25	2
TN	14.25	14.25	14	14.25	14.5	14.5	14.25	14.25	14.5
FP	0.25	0.25	0.5	0.25	0	0	0.25	0.25	0
FN	3	2.75	2.75	3	3.5	2.25	2.25	2.5	1.75
Precision	0.375	0.625	0.541	0.375	0.250	1	0.666	0.687	0.750
Recall	0.133	0.291	0.300	0.250	0.083	0.445	0.458	0.395	0.533
Accuracy	81.9%	83.3%	82.2%	82.3%	80.9%	87.7%	86.3%	84.8%	**90.5%**

Table A.11: Classification Results with KNN on Original Space - OLI Statistics - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	1	3	2	2.25	3.25	3.5	2	1.75	2.25
TN	24.75	24.75	25.25	25.5	25	23.75	25.75	26.25	26.5
FP	1	1.75	1.25	1	1.5	2.75	0.75	0.25	0
FN	8	6.75	7.75	8.25	7.25	7	8.5	8.75	8.25
Precision	0.583	0.635	0.566	0.729	0.641	0.552	0.750	0.875	1
Recall	0.115	0.301	0.205	0.219	0.311	0.321	0.201	0.182	0.212
Accuracy	74.2%	76.6%	75.2%	75.1%	76.4%	73.8%	75.2%	75.9%	**77.8%**

Table A.12: Classification Results with KNN on Original Space - OLI Statistics - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3	2.25	1.75	4	3.5	5	4.5	3.75	4.5
TN	36.5	37.5	37.25	36.5	36.5	36.25	36.25	37.75	38
FP	1.75	0.75	1	1.75	2	2.25	2.25	0.75	0.5
FN	8.5	9.25	9.75	8	8.5	7	7.5	8.25	7.5
Precision	0.666	0.762	0.322	0.692	0.803	0.752	0.746	0.887	0.937
Recall	0.271	0.189	0.156	0.353	0.298	0.404	0.355	0.316	0.397
Accuracy	79.5%	80.0%	78.6%	80.9%	79.5%	81.9%	80.8%	82.3%	**84.3%**

Table A.13: Classification Results with KNN on Original Space - OLI Statistics - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	2.75	4.75	3.5	2	2.75	2	2.5	3.75	3
TN	48.5	48.75	49.25	50.25	51.25	52.25	51.5	51.5	52.25
FP	2.75	3	2.5	1.5	1.25	0.25	1	1	0.25
FN	10	8	9.25	11.5	10.75	11.5	11	9.75	10.5
Precision	0.437	0.622	0.457	0.506	0.717	0.916	0.750	0.824	0.95
Recall	0.190	0.371	0.260	0.147	0.197	0.170	0.188	0.308	0.244
Accuracy	80.0%	82.8%	81.7%	80.0%	81.7%	82.1%	81.7%	83.6%	**83.6%**

Table A.14: Classification Results with KNN on Original Space - OLI Statistics - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	1.25	3.75	3.5	5	5	6.5	5.75	6	7.75
TN	61	61.75	62.5	61.25	62.5	61.75	62.5	62.75	62.25
FP	0.75	1.75	1	2.25	1.5	2.25	1.5	1.25	1.75
FN	20.75	19	19.25	19.25	19.25	17.75	18.5	18.25	16.5
Precision	0.375	0.792	0.909	0.739	0.816	0.841	0.795	0.858	0.842
Recall	0.063	0.175	0.169	0.196	0.205	0.264	0.238	0.246	0.332
Accuracy	74.3%	75.9%	76.5%	75.5%	76.4%	77.3%	77.3%	77.9%	**79.3%**

Table A.15: Classification Results with KNN on Original Space - OLI Statistics - 50% train 50% test

	50%	60%	70%	80%	90%
TP	1.25	1.5	1.5	1.75	2
TN	1.25	1.5	1.25	1.25	1.25
FP	0.75	0.75	1	1	1
FN	2.75	2.5	2.5	2.25	2
Precision	0.316	0.333	0.309	0.559	0.559
Recall	0.437	0.500	0.500	0.550	0.600
Accuracy	45.2%	51.0%	47.4%	50.5%	**53.7%**

Table A.16: Classification Results on Discriminative Latent factors - Mastery Grids - 90% train 10% test

	50%	60%	70%	80%	90%
TP	3	3.25	3.5	3.5	3.75
TN	2.25	2.75	3	2.75	2.5
FP	2.75	2.5	2.25	2.5	2.75
FN	2.25	2	2	2	1.75
Precision	0.516	0.552	0.635	0.669	0.632
Recall	0.569	0.619	0.657	0.657	0.692
Accuracy	50.2%	55.9%	**59.0%**	57.2%	57.9%

Table A.17: Classification Results on Discriminative Latent factors - Mastery Grids - 80% train 20% test

	50%	60%	70%	80%	90%
TP	4	4	4.25	4	4
TN	3.25	3	3.25	2.75	2.5
FP	5.5	6	5.75	6.75	7
FN	3.25	3.25	3.5	3.75	3.75
Precision	0.425	0.404	0.432	0.378	0.366
Recall	0.628	0.625	0.576	0.551	0.557
Accuracy	**44.9%**	42.9%	44.7%	39.0%	37.4%

Table A.18: Classification Results on Discriminative Latent factors - Mastery Grids - 70% train 30% test

	50%	60%	70%	80%	90%
TP	4.5	5	5.75	6	6
TN	5	5	5	5.5	5.25
FP	5.75	6.5	6.5	6.5	6.75
FN	4.5	4	3.75	3.5	3.5
Precision	0.460	0.451	0.487	0.5021	0.494
Recall	0.540	0.587	0.635	0.668	0.676
Accuracy	48.4%	48.5%	50.8%	**53.2%**	52.2%

Table A.19: Classification Results on Discriminative Latent factors - Mastery Grids - 60% train 40% test

	50%	60%	70%	80%	90%
TP	5.5	6	5.75	6.25	6.25
TN	4.5	4.5	4.75	5.25	4.75
FP	8.25	8.75	8.5	8	8.5
FN	6.25	5.75	7	6.5	6.5
Precision	0.457	0.467	0.481	0.517	0.506
Recall	0.516	0.567	0.487	0.526	0.532
Accuracy	40.8%	42.1%	40.5%	**44.4%**	42.5%

Table A.20: Classification Results on Discriminative Latent factors - Mastery Grids - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	12.25	14	13.25	12.5	11.5	13	13	13	13
TN	23.25	23	22	22.25	22.5	22.5	24.5	23.25	22
FP	12	13	14	13.75	13.5	13.5	11.5	12.75	14
FN	18.75	19	19.75	20.75	21.75	21.25	21.25	21.25	21.25
Precision	0.507	0.511	0.508	0.486	0.464	0.533	0.562	0.535	0.505
Recall	0.395	0.424	0.401	0.377	0.347	0.380	0.381	0.381	0.381
Accuracy	53.4%	**53.5%**	51.0%	50.1%	49.0%	50.5%	53.3%	51.5%	49.8%

Table A.21: Classification Results on Discriminative Latent factors - OLI Psychology - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	35.25	34.5	36	34.25	34	34.5	34.75	35	35
TN	31.25	34.25	34.75	34.75	34	32.25	32.75	33.75	34
FP	29	27.25	28.5	29.25	30	32	31.5	30.5	30.25
FN	32.5	36.25	35.25	38	38.75	39	38.75	38.5	38.75
Precision	0.583	0.591	0.593	0.577	0.580	0.563	0.563	0.581	0.571
Recall	0.527	0.495	0.513	0.482	0.476	0.478	0.482	0.485	0.485
Accuracy	52.0%	51.9%	**52.6%**	50.6%	49.7%	48.4%	49.0%	49.9%	50.0%

Table A.22: Classification Results on Discriminative Latent factors - OLI Psychology - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	34.75	30.5	32.25	30.75	29.5	27.75	27.25	26.5	24.75
TN	62.75	64.5	71	71.75	74	76.5	78	77.75	76.5
FP	33.25	32.75	28	28	25.75	23.5	22.25	22.5	23.75
FN	62.25	71.25	71.5	75.5	77	80.25	81.25	82	84
Precision	0.513	0.488	0.546	0.552	0.564	0.584	0.598	0.605	0.588
Recall	0.362	0.305	0.319	0.297	0.286	0.265	0.260	0.253	0.236
Accuracy	50.4%	47.7%	**50.9%**	49.7%	50.1%	50.1%	50.4%	49.9%	48.4%

Table A.23: Classification Results on Discriminative Latent factors - OLI Psychology - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	59.5	60.5	54	60	59	58	61.25	60.5	58
TN	72	78	79.5	79.75	75	75.25	76.5	75.5	79.75
FP	57.25	54.5	55	56	60.75	61	60.25	61.25	57
FN	68.25	73	82.25	78.25	79.25	80.75	78	79.5	82.25
Precision	0.512	0.529	0.481	0.519	0.478	0.460	0.486	0.479	0.484
Recall	0.466	0.451	0.394	0.431	0.423	0.414	0.434	0.428	0.409
Accuracy	51.1%	**52.0%**	49.2%	51.0%	48.9%	48.4%	49.9%	49.1%	49.7%

Table A.24: Classification Results on Discriminative Latent factors - OLI Psychology - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	60.25	62	66	66.25	68.25	69	69.5	71	70.25
TN	106.5	109.5	107.7	108.5	105.2	105.5	104	104.5	104.2
FP	58.75	59.25	63.75	65	68.25	68.25	70.5	70	70.25
FN	95.5	101.2	101	104.2	103.2	103.7	104	103.25	104.2
Precision	0.505	0.506	0.475	0.495	0.487	0.519	0.516	0.521	0.542
Recall	0.390	0.386	0.404	0.394	0.404	0.404	0.405	0.413	0.407
Accuracy	**51.9%**	51.6%	51.3%	50.8%	50.3%	50.3%	49.8%	50.3%	50.0%

Table A.25: Classification Results on Discriminative Latent factors - OLI Psychology - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	1	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.5
TN	12.25	12.75	12.25	11.5	12.5	12.25	13.25	13	12.75
FP	2.25	1.75	2.25	3	2	2.25	1.25	1.5	1.75
FN	2.5	3	3	3	3	3	3	3	3.25
Precision	0.169	0.083	0.187	0.118	0.208	0.196	0.225	0.183	0.175
Recall	0.312	0.250	0.250	0.250	0.250	0.250	0.250	0.250	0.166
Accuracy	73.7%	73.7%	70.8%	66.7%	72.4%	71.0%	**76.6%**	75.3%	72.3%

Table A.26: Classification Results on Discriminative Latent factors - OLI Statistics - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3.75	3.5	2.25	2.5	3.5	3	2.75	2.5	2
TN	19.75	22.25	22.5	22.5	23.75	23.25	24	23.5	24.25
FP	6	4.25	4	4	2.75	3.25	2.5	3	2.25
FN	5.25	6.25	7.5	8	7	7.5	7.75	8	8.5
Precision	0.393	0.464	0.363	0.416	0.648	0.416	0.398	0.336	0.347
Recall	0.430	0.363	0.228	0.229	0.307	0.262	0.229	0.210	0.178
Accuracy	67.5%	71.0%	68.2%	67.5%	**73.6%**	71.0%	72.2%	70.2%	71.1%

Table A.27: Classification Results on Discriminative Latent factors - OLI Statistics - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	2.75	2	1	2	2	3	2.75	3	3
TN	31	34.5	34.75	34.25	36	36.25	35.5	35.5	36
FP	7.25	3.75	3.5	4	2.5	2.25	3	3	2.5
FN	8.75	9.5	10.5	10	10	9	9.25	9	9
Precision	0.291	0.287	0.196	0.322	0.457	0.668	0.587	0.555	0.585
Recall	0.251	0.212	0.103	0.196	0.184	0.282	0.269	0.290	0.282
Accuracy	68.0%	73.5%	72.0%	72.4%	75.5%	**77.9%**	75.9%	76.4%	77.4%

Table A.28: Classification Results on Discriminative Latent factors - OLI Statistics - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3.75	4	3.75	5.25	5	5.5	5.5	6	5.75
TN	35.5	38.75	39.25	38.75	39.5	39.5	38	38.25	37.25
FP	15.75	13	12.5	13	13	13	14.5	14.25	15.25
FN	9	8.75	9	8.25	8.5	8	8	7.5	7.75
Precision	0.173	0.236	0.213	0.297	0.337	0.335	0.330	0.239	0.224
Recall	0.298	0.307	0.284	0.414	0.367	0.411	0.403	0.472	0.456
Accuracy	61.1%	66.4%	66.7%	67.5%	67.5%	**68.3%**	66.2%	67.2%	65.4%

Table A.29: Classification Results on Discriminative Latent factors - OLI Statistics - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	9.75	10.25	9.5	9.25	10	10.5	12.25	12.25	12.25
TN	37.75	45	47.5	50	50.75	50.25	49.25	48.25	49.5
FP	24	18.5	16	13.5	13.25	13.75	14.75	15.75	14.5
FN	12.25	12.5	13.25	15	14.25	13.75	12	12	12
Precision	0.276	0.345	0.363	0.407	0.455	0.455	0.477	0.470	0.504
Recall	0.436	0.440	0.410	0.378	0.408	0.423	0.497	0.505	0.497
Accuracy	56.6%	64.0%	66.0%	67.5%	68.8%	68.8%	69.7%	68.6%	**70.0%**

Table A.30: Classification Results on Discriminative Latent factors - OLI Statistics - 50% train 50% test

	50%	60%	70%	80%	90%
TP	2.75	2	2	2.25	2.75
TN	1	0.75	1	1	1
FP	1	1.5	1.25	1.25	1.25
FN	1.25	2	2	1.75	1.25
Precision	0.720	0.428	0.470	0.511	0.657
Recall	0.650	0.475	0.475	0.600	0.712
Accuracy	58.3%	42.7%	45.8%	54.1%	**60.8%**

Table A.31: Classification Results on Latent factors using multi-view - Mastery Grids - 90% train 10% test

	50%	60%	70%	80%	90%
TP	2.75	2.75	3	2.5	2.5
TN	2.5	2.75	3	2.75	3
FP	2.5	2.5	2.25	2.5	2.25
FN	2.5	2.5	2.5	3	3
Precision	0.527	0.519	0.586	0.500	0.541
Recall	0.505	0.505	0.557	0.471	0.471
Accuracy	51.8%	52.7%	**55.6%**	48.0%	50.8%

Table A.32: Classification Results on Latent factors using multi-view - Mastery Grids - 80% train 20% test

	50%	60%	70%	80%	90%
TP	4.5	5	5.5	4.75	4.75
TN	3.5	3	3.25	3	3.75
FP	5.25	6	5.75	6.5	5.75
FN	2.75	2.25	2.25	3	3
Precision	0.437	0.468	0.513	0.461	0.482
Recall	0.524	0.631	0.686	0.608	0.608
Accuracy	49.7%	49.2%	**52.2%**	45.4%	49.4%

Table A.33: Classification Results on Latent factors using multi-view - Mastery Grids - 70% train 30% test

	50%	60%	70%	80%	90%
TP	4.75	4.75	4.5	4.25	3.5
TN	4	4.75	4	4.25	4
FP	6.75	6.75	7.5	7.75	8
FN	4.25	4.25	5	5.25	6
Precision	0.414	0.433	0.357	0.337	0.286
Recall	0.571	0.581	0.530	0.513	0.419
Accuracy	45.6%	**47.4%**	41.2%	40.3%	35.0%

Table A.34: Classification Results on Latent factors using multi-view - Mastery Grids - 60% train 40% test

	50%	60%	70%	80%	90%
TP	8.75	8.75	9.25	8.5	8.25
TN	4.5	3.5	4.5	4.25	3.75
FP	8.25	9.75	8.75	9	9.5
FN	3	3	3.5	4.25	4.5
Precision	0.513	0.484	0.514	0.489	0.466
Recall	0.746	0.761	0.729	0.681	0.658
Accuracy	**54.1%**	49.1%	52.9%	49.2%	46.1%

Table A.35: Classification Results on Latent factors using multi-view - Mastery Grids - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	17	16	19	19.25	18.5	18.75	19.5	18.75	17.75
TN	14.75	15.75	16.75	15.5	14.75	14.75	16	16.5	16.75
FP	18.25	17.5	16.75	18.25	19	19	17.75	17.25	17
FN	13.75	16.5	14	14	15	15.5	14.75	15.5	16.5
Precision	0.475	0.478	0.531	0.519	0.496	0.495	0.527	0.518	0.508
Recall	0.561	0.494	0.575	0.579	0.553	0.547	0.569	0.547	0.518
Accuracy	50.0%	48.3%	**53.7%**	52.0%	49.5%	49.4%	52.5%	52.0%	50.9%

Table A.36: Classification Results on Latent factors using multi-view - OLI Psychology - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	41.25	41	43.25	45.25	45	44.75	46.25	47	46.5
TN	20	18.75	21	21	22.25	21	22.25	21	20.25
FP	40.25	42.75	42.25	43	41.75	43.25	42	43.25	44
FN	26.5	29.75	28	27	27.75	28.75	27.25	26.5	27.25
Precision	0.506	0.491	0.506	0.511	0.519	0.510	0.523	0.519	0.513
Recall	0.611	0.582	0.610	0.631	0.621	0.612	0.633	0.643	0.634
Accuracy	47.8%	45.1%	47.7%	48.4%	49.0%	47.6%	**49.5%**	49.1%	48.2%

Table A.37: Classification Results on Latent factors using multi-view - OLI Psychology - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	59.25	53.5	56	59.25	58.5	60.5	58.5	58.25	56.75
TN	35	40.75	41.5	40.75	43	41.25	45.25	44.5	44.5
FP	61	56.5	57.5	59	56.75	58.75	55	55.75	55.75
FN	37.75	48.25	47.75	47	48	47.5	50	50.25	52
Precision	0.492	0.478	0.487	0.494	0.504	0.502	0.509	0.505	0.501
Recall	0.607	0.521	0.535	0.556	0.547	0.557	0.534	0.533	0.520
Accuracy	48.8%	47.3%	48.1%	48.5%	49.2%	48.9%	**49.6%**	49.1%	48.4%

Table A.38: Classification Results on Latent factors using multi-view - OLI Psychology - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	80.5	83.75	81.5	83.75	86	87.5	88.75	88.75	87.75
TN	52.5	56.75	55.5	52.5	57	55.25	52.25	50.5	52
FP	76.75	75.75	79	83.25	78.75	81	84.5	86.25	84.75
FN	47.25	49.75	54.75	54.5	52.25	51.25	50.5	51.25	52.5
Precision	0.512	0.522	0.503	0.499	0.521	0.518	0.511	0.506	0.507
Recall	0.630	0.626	0.594	0.603	0.620	0.628	0.636	0.633	0.624
Accuracy	51.7%	**52.8%**	50.6%	49.7%	52.2%	51.9%	51.1%	50.3%	50.4%

Table A.39: Classification Results on Latent factors using multi-view - OLI Psychology - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	80.75	78.25	81.5	81.25	80.5	78.25	76.25	74.5	69.25
TN	83	91	93.75	95.25	94.5	96.25	98.75	101	104
FP	82.25	77.75	77.75	78.25	79	77.5	75.75	73.5	70.5
FN	75	85	85.5	89.25	91	94.5	97.25	99.75	105.2
Precision	0.496	0.501	0.510	0.506	0.505	0.497	0.493	0.505	0.497
Recall	0.513	0.478	0.488	0.476	0.469	0.452	0.439	0.428	0.396
Accuracy	51.0%	50.9%	**51.7%**	51.3%	50.7%	50.3%	50.3%	50.3%	49.6%

Table A.40: Classification Results on Latent factors using multi-view - OLI Psychology - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3.5	3.75	3.75	3.75	3.75	3.75	3.75	3.75	3.75
TN	0	0	0	0	0	0	0	0	0
FP	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.194	0.205	0.205	0.205	0.205	0.205	0.205	0.205	0.205
Recall	1	1	1	1	1	1	1	1	1
Accuracy	19.4%	**20.5%**	**20.5%**	20.5%	20.5%	20.5%	**20.5%**	20.5%	20.5%

Table A.41: Classification Results on Latent factors using multi-view - OLI Statistics - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	9	9.75	9.75	10.5	10.5	10.5	10.5	10.5	10.5
TN	0	0	0	0	0	0	0	0	0
FP	25.75	26.5	26.5	26.5	26.5	26.5	26.5	26.5	26.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.256	0.267	0.267	0.281	0.281	0.281	0.281	0.281	0.281
Recall	1	1	1	1	1	1	1	1	1
Accuracy	25.6%	26.7%	26.7%	28.1%	28.1%	28.1%	**28.1%**	28.1%	28.1%

Table A.42: Classification Results on Latent factors using multi-view - OLI Statistics - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	11.5	11.5	11.5	12	12	12	12	12	12
TN	0	0.25	0.25	0	0.25	0.25	0	0	0
FP	38.25	38	38	38.25	38.25	38.25	38.5	38.5	38.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.228	0.229	0.229	0.236	0.236	0.236	0.235	0.235	0.235
Recall	1	1	1	1	1	1	1	1	1
Accuracy	22.8%	23.4%	23.4%	23.6%	24.0%	24.0%	**23.5%**	23.5%	23.5%

Table A.43: Classification Results on Latent factors using multi-view - OLI Statistics - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	12.75	12.75	12.75	13.25	13.5	13.5	13.5	13.5	13.5
TN	2	0.5	0.75	0.75	0.5	1	1	1	1.25
FP	49.25	51.25	51	51	52	51.5	51.5	51.5	51.25
FN	0	0	0	0.25	0	0	0	0	0
Precision	0.203	0.198	0.199	0.205	0.206	0.206	0.207	0.206	0.207
Recall	1	1	1	0.984	1	1	1	1	1
Accuracy	23.0%	20.5%	20.9%	21.4%	21.2%	21.9%	**21.9%**	21.9%	22.3%

Table A.44: Classification Results on Latent factors using multi-view - OLI Statistics - 60% train 40% test

$$
\begin{aligned}
L = \ & \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \epsilon\|S - \varepsilon W W^T\|_F^2 + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned}
\tag{A.3}
$$

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	22	22.5	22.25	24.25	24.25	24.25	24.25	24.25	24
TN	0.75	0.5	0.5	0.5	0.25	0.5	0.25	0.25	0.25
FP	61	63	63	63	63.75	63.5	63.75	63.75	63.75
FN	0	0.25	0.5	0	0	0	0	0	0.25
Precision	0.264	0.263	0.260	0.277	0.275	0.275	0.275	0.275	0.273
Recall	1	0.986	0.972	1	1	1	1	1	0.986
Accuracy	27.1%	26.6%	26.3%	28.1%	27.7%	28.0%	**27.7%**	27.7%	27.4%

Table A.45: Classification Results on Latent factors using multi-view - OLI Statistics - 50% train 50% test

	50%	60%	70%	80%	90%
TP	2.75	2.5	2.25	2	2.25
TN	3.75	3.75	3.75	3.75	3.5
FP	3.25	4	4	4.5	4.75
FN	3.75	4	4.5	4.75	4.5
Precision	0.500	0.411	0.369	0.295	0.330
Recall	0.418	0.368	0.304	0.273	0.323
Accuracy	**48.0%**	43.7%	39.9%	35.7%	37.5%

Table A.46: Classification Results on Discriminative Latent factors using multi-view - Mastery Grids - 90% train 10% test

	50%	60%	70%	80%	90%
TP	3	3	2.75	3.5	3.75
TN	2.5	2.75	3.25	3	3
FP	2.75	2.75	2.25	2.5	2.5
FN	4.75	4.75	5	4.25	4
Precision	0.533	0.533	0.597	0.607	0.666
Recall	0.445	0.445	0.409	0.500	0.608
Accuracy	42.5%	43.1%	44.7%	50.2%	**56.8%**

Table A.47: Classification Results on Discriminative Latent factors using multi-view - Mastery Grids - 80% train 20% test

	50%	60%	70%	80%	90%
TP	2	2.75	3.25	3.5	2.75
TN	3.75	3.5	3.75	4.25	4
FP	3.25	3.75	3.5	3.25	3.5
FN	3	2.25	2.5	2.25	3
Precision	0.237	0.476	0.560	0.571	0.518
Recall	0.312	0.465	0.487	0.512	0.375
Accuracy	46.7%	52.6%	56.0%	**59.5%**	50.9%

Table A.48: Classification Results on Discriminative Latent factors using multi-view - Mastery Grids - 70% train 30% test

	50%	60%	70%	80%	90%
TP	3.25	3.5	3.5	4	3.75
TN	5.5	5.5	5	4.5	4.75
FP	5.25	6	6.5	7.5	7.25
FN	5.75	5.5	6	5.5	5.75
Precision	0.395	0.377	0.355	0.363	0.361
Recall	0.399	0.441	0.432	0.477	0.413
Accuracy	**44.9%**	44.2%	40.4%	39.4%	39.1%

Table A.49: Classification Results on Discriminative Latent factors using multi-view - Mastery Grids - 60% train 40% test

	50%	60%	70%	80%	90%
TP	5.75	5.25	5.75	6	6
TN	6	5.5	5.75	6	6.25
FP	6.75	7.75	7.5	7.25	7
FN	6	6.5	7	6.75	6.75
Precision	0.471	0.399	0.427	0.450	0.459
Recall	0.485	0.438	0.449	0.456	0.460
Accuracy	**48.0%**	42.8%	44.0%	46.0%	47.0%

Table A.50: Classification Results on Discriminative Latent factors using multi-view - Mastery Grids - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	19.5	20.25	19.75	19.25	20.75	20.25	21.75	20.75	19.75
TN	14.25	14	16.75	16	14.75	15.75	16.5	18.5	18
FP	18.75	19.25	16.75	17.75	19	18	17.25	15.25	15.75
FN	11.25	12.25	13.25	14	12.75	14	12.5	13.5	14.5
Precision	0.505	0.507	0.544	0.518	0.526	0.523	0.577	0.591	0.562
Recall	0.635	0.625	0.598	0.579	0.619	0.591	0.635	0.606	0.576
Accuracy	53.0%	52.1%	54.8%	52.7%	52.9%	53.0%	56.2%	**57.7%**	55.5%

Table A.51: Classification Results on Discriminative Latent factors using multi-view - OLI Psychology - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	38.25	38.75	38.25	39.25	36.25	38	35.25	36.25	37.5
TN	21	26	27.5	28	28.75	30.5	30.75	29.5	28.5
FP	39.25	35.5	35.75	36	35.25	33.75	33.5	34.75	35.75
FN	29.5	32	33	33	36.5	35.5	38.25	37.25	36.25
Precision	0.488	0.519	0.517	0.523	0.508	0.530	0.510	0.510	0.511
Recall	0.566	0.547	0.537	0.543	0.499	0.518	0.481	0.494	0.509
Accuracy	46.1%	48.8%	48.8%	49.2%	47.4%	**49.6%**	47.7%	47.5%	47.6%

Table A.52: Classification Results on Discriminative Latent factors using multi-view - OLI Psychology - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	55.25	58	56.5	60.5	55.25	56.5	53.75	53.75	52.5
TN	39.75	46.25	52.5	52.5	49.5	52.25	50	52	53.5
FP	56.25	51	46.5	47.25	50.25	47.75	50.25	48.25	46.75
FN	41.75	43.75	47.25	45.75	51.25	51.5	54.75	54.75	56.25
Precision	0.490	0.527	0.546	0.559	0.520	0.539	0.509	0.520	0.525
Recall	0.567	0.568	0.543	0.568	0.515	0.520	0.490	0.490	0.479
Accuracy	49.2%	52.4%	53.7%	**54.8%**	50.7%	52.2%	49.6%	50.6%	50.6%

Table A.53: Classification Results on Discriminative Latent factors using multi-view - OLI Psychology - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	60.75	60.75	60.25	61	61.25	61.5	62	59	60.5
TN	64	72.5	74.75	73	76.75	78.25	78.75	78.5	78.75
FP	65.25	60	59.75	62.75	59	58	58	58.25	58
FN	67	72.75	76	77.25	77	77.25	77.25	81	79.75
Precision	0.487	0.505	0.502	0.492	0.508	0.513	0.515	0.503	0.511
Recall	0.477	0.455	0.442	0.440	0.441	0.441	0.443	0.420	0.430
Accuracy	48.5%	50.1%	49.8%	48.9%	50.3%	**50.8%**	51.0%	49.6%	50.2%

Table A.54: Classification Results on Discriminative Latent factors using multi-view - OLI Psychology - 60% train 60% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	77.75	81.25	83.25	81.75	83.25	80.5	84.5	83.25	83.75
TN	79.5	84	88.25	90.5	91.5	90.5	88.75	90	92.25
FP	85.75	84.75	83.25	83	82	83.25	85.75	84.5	82.25
FN	78	82	83.75	88.75	88.25	92.25	89	91	90.75
Precision	0.469	0.486	0.495	0.491	0.501	0.486	0.489	0.489	0.495
Recall	0.495	0.495	0.495	0.477	0.483	0.464	0.484	0.475	0.477
Accuracy	48.9%	49.7%	**50.6%**	50.0%	50.6%	49.3%	49.7%	49.6%	50.4%

Table A.55: Classification Results on Discriminative Latent factors using multi-view - OLI Psychology - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3.5	3.75	3.75	3.75	3.75	3.75	3.75	3.75	3.75
TN	0	0	0	0	0	0	0	0	0
FP	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.194	0.205	0.205	0.205	0.205	0.205	0.205	0.205	0.205
Recall	1	1	1	1	1	1	1	1	1
Accuracy	19.4%	20.5%	20.5%	20.5%	20.5%	20.5%	20.5%	20.5%	**20.5%**

Table A.56: Classification Results on Discriminative Latent factors using multi-view - OLI Statistics - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	8.75	9.5	9.5	10.25	10.25	10.25	10.25	10.25	10.25
TN	1.5	1	1.25	1	1.5	1.75	1.5	1.75	1.5
FP	24.25	25.5	25.25	25.5	25	24.75	25	24.75	25
FN	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Precision	0.263	0.270	0.273	0.286	0.291	0.294	0.291	0.294	0.291
Recall	0.972	0.977	0.977	0.979	0.979	0.979	0.979	0.979	0.979
Accuracy	29.1%	28.7%	29.4%	30.1%	31.4%	**32.1%**	31.4%	**32.1%**	31.4%

Table A.57: Classification Results on Discriminative Latent factors using multi-view - OLI Statistics - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	10	9.25	9.25	9.75	10.25	10	9.75	10	10
TN	7	8.25	9.75	11	10.75	10.75	11.25	10	11.25
FP	31.25	30	28.5	27.25	27.75	27.75	27.25	28.5	27.25
FN	1.5	2.25	2.25	2.25	1.75	2	2.25	2	2
Precision	0.242	0.227	0.238	0.263	0.271	0.272	0.261	0.268	0.275
Recall	0.878	0.803	0.796	0.808	0.855	0.831	0.801	0.823	0.823
Accuracy	33.9%	35.0%	38.2%	41.4%	41.8%	41.4%	41.9%	40.0%	**42.6**%

Table A.58: Classification Results on Discriminative Latent factors using multi-view - OLI Statistics - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	5.5	7	6.5	6.5	6.75	6.5	6.25	6.75	6.75
TN	28	31.25	31.25	29.75	27.25	26.5	26.5	25.5	24.75
FP	23.25	20.5	20.5	22	25.25	26	26	27	27.75
FN	7.25	5.75	6.25	7	6.75	7	7.25	6.75	6.75
Precision	0.289	0.387	0.296	0.303	0.285	0.261	0.232	0.239	0.233
Recall	0.461	0.597	0.565	0.531	0.545	0.531	0.529	0.560	0.559
Accuracy	51.8%	**58.7%**	58.1%	55.1%	51.2%	49.7%	49.3%	48.5%	47.4%

Table A.59: Classification Results on Discriminative Latent factors using multi-view - OLI Statistics - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	0.5	0	0.5	0.5	0.25	0.5	0.25	0.25	0.25
TN	60.25	62	62.5	63	62.25	63	63	62.75	62.75
FP	1.5	1.5	1	0.5	1.75	1	1	1.25	1.25
FN	21.5	22.75	22.25	23.75	24	23.75	24	24	24
Precision	0.062	0	0.083	0.125	0.031	0.083	0.050	0.041	0.041
Recall	0.027	0	0.027	0.026	0.013	0.026	0.013	0.013	0.013
Accuracy	72.5%	71.8%	**73.0%**	72.3%	70.8%	71.9%	71.6%	71.3%	71.3%

Table A.60: Classification Results on Discriminative Latent factors using multi-view - OLI Statistics - 50% train 50% test

	50%	60%	70%	80%	90%
TP	2.75	2.75	2.5	2.25	1.75
TN	0.5	0.5	0.5	0.5	0.75
FP	1.5	1.75	1.75	1.75	1.5
FN	1.25	1.25	1.5	1.75	2.25
Precision	0.661	0.578	0.583	0.541	0.525
Recall	0.712	0.712	0.675	0.612	0.437
Accuracy	**60.7%**	54.1%	51.0%	47.4%	39.1%

Table A.61: Classification Results using multi-view and patterns' structure - Mastery Grids - 90% train 10% test

	50%	60%	70%	80%	90%
TP	3.25	3.75	3.75	4	4
TN	1	2.25	1.5	1.25	1
FP	3.25	2.25	3	3.25	3.5
FN	2	1.5	1.75	1.5	1.5
Precision	0.507	0.628	0.570	0.569	0.552
Recall	0.644	0.730	0.707	0.742	0.742
Accuracy	47.0%	**62.2%**	51.7%	52.4%	48.9%

Table A.62: Classification Results using multi-view and patterns' structure - Mastery Grids - 80% train 20% test

	50%	60%	70%	80%	90%
TP	3	3.25	2.75	2.25	2
TN	5.5	6	6	6.75	6.75
FP	3.25	3	3	2.75	2.75
FN	4.25	4	5	5.5	5.75
Precision	0.347	0.372	0.347	0.350	0.332
Recall	0.444	0.472	0.361	0.308	0.283
Accuracy	53.5%	**57.1%**	52.3%	52.3%	51.0%

Table A.63: Classification Results using multi-view and patterns' structure - Mastery Grids - 70% train 30% test

	50%	60%	70%	80%	90%
TP	2.75	2.5	3.5	4	3.75
TN	5.75	6.75	7.75	8.25	8.5
FP	5	4.75	3.75	3.75	3.5
FN	6.25	6.5	6	5.5	5.75
Precision	0.370	0.399	0.513	0.547	0.543
Recall	0.354	0.323	0.416	0.454	0.413
Accuracy	42.6%	44.5%	53.7%	**56.8%**	**56.8%**

Table A.64: Classification Results using multi-view and patterns' structure - Mastery Grids - 60% train 40% test

	50%	60%	70%	80%	90%
TP	5.5	5.5	5.75	6.25	6.5
TN	7.25	7.75	8	8	7.75
FP	5.5	5.5	5.25	5.25	5.5
FN	6.25	6.25	7	6.5	6.25
Precision	0.448	0.468	0.488	0.508	0.548
Recall	0.449	0.460	0.433	0.483	0.507
Accuracy	51.7%	52.8%	52.7%	**54.7%**	**54.7%**

Table A.65: Classification Results using multi-view and patterns' structure - Mastery Grids - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	11.75	14.25	14	14.5	14.25	15.25	14.75	15.75	14.75
TN	17.25	17.5	19.5	19.25	20.5	20.5	19.25	19.5	19.75
FP	15.75	15.75	14	14.5	13.25	13.25	14.5	14.25	14
FN	19	18.25	19	18.75	19.25	19	19.5	18.5	19.5
Precision	0.432	0.499	0.542	0.474	0.493	0.556	0.526	0.540	0.508
Recall	0.388	0.440	0.424	0.434	0.426	0.443	0.429	0.458	0.429
Accuracy	45.4%	48.4%	50.5%	50.2%	51.6%	**52.6%**	49.9%	51.7%	50.6%

Table A.66: Classification Results using multi-view and patterns' structure - OLI Psychology - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	38.75	42.25	44	44.75	48	46.75	49	47.75	45.75
TN	27.25	25.5	23.5	24	20.75	20.25	20.75	20.25	20.5
FP	33	36	39.75	40	43.25	44	43.5	44	43.75
FN	29	28.5	27.25	27.5	24.75	26.75	24.5	25.75	28
Precision	0.545	0.539	0.529	0.529	0.527	0.512	0.531	0.520	0.506
Recall	0.574	0.604	0.623	0.626	0.665	0.641	0.672	0.655	0.624
Accuracy	**51.5%**	51.3%	50.2%	50.6%	50.3%	48.7%	50.7%	49.4%	48.1%

Table A.67: Classification Results using multi-view and patterns' structure - OLI Psychology - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	67.25	68	72.25	72.5	73	73.75	75.25	75.5	73.25
TN	30.75	33.5	28.75	34	31.75	31.5	30.25	30.5	32
FP	65.25	63.75	70.25	65.75	68	68.5	70	69.75	68.25
FN	29.75	33.75	31.5	33.75	33.5	34.25	33.25	33	35.5
Precision	0.507	0.514	0.505	0.529	0.520	0.520	0.521	0.523	0.521
Recall	0.696	0.670	0.700	0.686	0.692	0.688	0.699	0.703	0.679
Accuracy	50.8%	51.0%	49.8%	**51.7%**	50.8%	50.6%	50.6%	50.8%	50.4%

Table A.68: Classification Results using multi-view and patterns' structure - OLI Psychology - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	56.75	53.75	56.25	58.25	58.25	58.25	58.25	57.5	58.5
TN	75.75	81.25	79	78.25	79.25	77.5	78.75	78.5	82.75
FP	53.5	51.25	55.5	57.5	56.5	58.75	58	58.25	54
FN	71	79.75	80	80	80	80.5	81	82.5	81.75
Precision	0.517	0.504	0.496	0.499	0.525	0.516	0.519	0.504	0.522
Recall	0.443	0.401	0.411	0.419	0.419	0.418	0.416	0.409	0.416
Accuracy	**51.5%**	50.7%	49.9%	49.8%	50.2%	49.3%	49.6%	49.1%	51.0%

Table A.69: Classification Results using multi-view and patterns' structure - OLI Psychology - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	90.25	89	91.25	90.25	87.5	85.5	85.25	84.75	85.25
TN	70	74	73	70.5	74	76.75	78.75	79	78
FP	95.25	94.75	98.5	103	99.5	97	95.75	95.5	96.5
FN	65.5	74.25	75.75	80.25	84	87.25	88.25	89.5	89.25
Precision	0.480	0.486	0.480	0.462	0.465	0.473	0.459	0.463	0.462
Recall	0.574	0.542	0.542	0.526	0.506	0.491	0.487	0.483	0.485
Accuracy	**49.9%**	49.0%	48.4%	46.7%	46.8%	46.8%	47.1%	46.9%	46.7%

Table A.70: Classification Results using multi-view and patterns' structure - OLI Psychology - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3.5	3.75	3.75	3.75	3.75	3.75	3.75	3.75	3.75
TN	0	0	0	0	0.25	0	0	0	0
FP	14.5	14.5	14.5	14.5	14.25	14.5	14.5	14.5	14.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.194	0.205	0.205	0.205	0.209	0.205	0.205	0.205	0.205
Recall	1	1	1	1	1	1	1	1	1
Accuracy	19.4%	20.5%	20.5%	20.5%	**21.9%**	20.5%	20.5%	20.5%	20.5%

Table A.71: Classification Results using multi-view and patterns' structure - OLI Statistics - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	9	9.75	9.75	10.5	10.5	10.5	10.5	10.5	10.5
TN	0.25	0.5	0.75	1.25	1.5	1.5	1.5	1.5	1.5
FP	25.5	26	25.75	25.25	25	25	25	25	25
FN	0	0	0	0	0	0	0	0	0
Precision	0.258	0.271	0.272	0.291	0.293	0.293	0.293	0.293	0.294
Recall	1	1	1	1	1	1	1	1	1
Accuracy	26.4%	28.1%	28.8%	31.6%	32.3%	**32.4%**	**32.4%**	**32.4%**	**32.4%**

Table A.72: Classification Results using multi-view and patterns' structure - OLI Statistics - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	11	11.25	10.75	11	11.25	11.5	11.5	11.5	11.25
TN	2.25	5	6.25	6.75	5.5	4.75	4.25	4	4.5
FP	36	33.25	32	31.5	33	33.75	34.25	34.5	34
FN	0.5	0.25	0.75	1	0.75	0.5	0.5	0.5	0.75
Precision	0.233	0.254	0.253	0.261	0.255	0.255	0.251	0.251	0.249
Recall	0.968	0.984	0.953	0.926	0.946	0.968	0.968	0.968	0.946
Accuracy	26.4%	32.1%	33.8%	**34.7%**	32.5%	31.6%	30.7%	30.2%	30.6%

Table A.73: Classification Results using multi-view and patterns' structure - OLI Statistics - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	11.5	11.75	10.5	11.5	11	11	11.25	12	11.5
TN	5.5	7.75	9.25	10.5	13.75	14.5	13.75	13.5	13
FP	45.75	44	42.5	41.25	38.75	38	38.75	39	39.5
FN	1.25	1	2.25	2	2.5	2.5	2.25	1.5	2
Precision	0.198	0.208	0.193	0.218	0.222	0.223	0.224	0.236	0.225
Recall	0.890	0.921	0.791	0.855	0.812	0.787	0.815	0.872	0.843
Accuracy	26.5%	30.2%	30.5%	33.6%	37.3%	**38.4%**	37.6%	38.4%	36.9%

Table A.74: Classification Results using multi-view and patterns' structure - OLI Statistics - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	18.5	17.5	16.5	18	19.5	19.5	18.75	18.5	18
TN	15.75	24	24.75	26	27	27.5	26	27.75	29.75
FP	46	39.5	38.75	37.5	37	36.5	38	36.25	34.25
FN	3.5	5.25	6.25	6.25	4.75	4.75	5.5	5.75	6.25
Precision	0.285	0.307	0.299	0.326	0.347	0.345	0.325	0.326	0.336
Recall	0.834	0.768	0.712	0.735	0.803	0.792	0.756	0.739	0.720
Accuracy	40.8%	48.1%	47.7%	50.0%	52.6%	53.2%	50.6%	52.3%	**54.0%**

Table A.75: Classification Results using multi-view and patterns' structure - OLI Statistics - 50% train 50% test

$$\begin{aligned}
L = \;& \gamma(\|X_1 - W_1 H_1^T\|_F^2 + \|X_2 - W_2 H_2^T\|_F^2) + \alpha\|W_{1,c} - W_{2,c}\|_F^2 + \beta\|W_{1,d}^T W_{2,d}\|_F^2 + \\
& \theta(\|Y_1 - [B_{c_1} B_{d_1}] \begin{bmatrix} H_{1,c}^T \\ H_{1,d}^T \end{bmatrix} \|^2 + \|Y_2 - [B_{c_2} B_{d_2}] \begin{bmatrix} H_{2,c}^T \\ H_{2,d}^T \end{bmatrix} \|^2) + \epsilon\|S - \varepsilon W W^T\|_F^2 + \\
& \lambda(\|H_1^T - \frac{1}{n_1} H_1^T e e^T\|_F^2 + \|H_2^T - \frac{1}{n_2} H_2^T e e^T\|_F^2) - \epsilon\|\frac{1}{n_1} H_{1,d}^T e - \frac{1}{n_2} H_{2,d}^T e\|_F^2 + \\
& (\|W_1\|^2 + \|W_2\|^2 + \|H_1\|^2 + \|H_2\|^2)
\end{aligned} \tag{A.4}$$

	50%	60%	70%	80%	90%
TP	2.75	2.75	2.5	2.75	2.75
TN	0.25	0.5	0.75	0.75	0.5
FP	1.75	1.75	1.5	1.5	1.75
FN	1.25	1.25	1.5	1.25	1.25
Precision	0.601	0.684	0.666	0.708	0.684
Recall	0.675	0.675	0.625	0.675	0.675
Accuracy	47.6%	54.6%	55.0%	**58.1%**	54.6%

Table A.76: Classification Results On discriminative latent factors using multi-view and patterns' structure - Mastery Grids - 90% train 10% test

	50%	60%	70%	80%	90%
TP	3.75	3.5	3.75	3.5	3.5
TN	2.25	2	2	2.75	2.75
FP	2.75	3.25	3.25	2.5	2.5
FN	1.5	1.75	1.75	2	2
Precision	0.588	0.525	0.538	0.593	0.707
Recall	0.742	0.707	0.707	0.657	0.657
Accuracy	**58.4%**	51.5%	52.5%	57.7%	57.7%

Table A.77: Classification Results On discriminative latent factors using multi-view and patterns' structure - Mastery Grids - 80% train 20% test

	50%	60%	70%	80%	90%
TP	5	5.5	5.5	5.5	5.5
TN	4.25	4.25	4.25	4.25	5
FP	4.5	4.75	4.75	5.25	4.5
FN	2.25	1.75	2.25	2.25	2.25
Precision	0.497	0.507	0.500	0.469	0.534
Recall	0.628	0.684	0.634	0.634	0.631
Accuracy	58.2%	60.3%	57.9%	57.0%	**61.5%**

Table A.78: Classification Results On discriminative latent factors using multi-view and patterns' structure - Mastery Grids - 70% train 30% test

	50%	60%	70%	80%	90%
TP	4	4.5	4.5	4.5	4.25
TN	5	5.75	5.75	6.75	6.25
FP	5.75	5.75	5.75	5.25	5.75
FN	5	4.5	5	5	5.25
Precision	0.410	0.435	0.421	0.452	0.399
Recall	0.467	0.504	0.454	0.454	0.412
Accuracy	45.4%	49.8%	48.9%	**52.6%**	48.9%

Table A.79: Classification Results On discriminative latent factors using multi-view and patterns' structure - Mastery Grids - 60% train 40% test

	50%	60%	70%	80%	90%
TP	7	6.75	8.25	9.5	9
TN	4.75	5.5	5.75	6.25	5.75
FP	8	7.75	7.5	7	7.5
FN	4.75	5	4.5	3.25	3.75
Precision	0.467	0.458	0.540	0.584	0.549
Recall	0.577	0.550	0.641	0.733	0.694
Accuracy	47.4%	48.8%	53.8%	**60.4%**	56.6%

Table A.80: Classification Results On discriminative latent factors using multi-view and patterns' structure - Mastery Grids - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	10.75	13.5	13.25	11.25	12.75	14.25	14.5	14.25	13.75
TN	22	21.5	20	21.75	21.5	21.25	19.75	20.25	20
FP	11	11.75	13.5	12	12.25	12.5	14	13.5	13.75
FN	20	19	19.75	22	20.75	20	19.75	20	20.5
Precision	0.502	0.540	0.503	0.471	0.523	0.550	0.539	0.544	0.510
Recall	0.349	0.415	0.401	0.339	0.379	0.417	0.424	0.417	0.402
Accuracy	51.5%	**53.4%**	50.1%	49.2%	50.9%	52.2%	50.4%	50.8%	49.7%

Table A.81: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Psychology - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	25.75	26	26.5	28.5	33	32.25	32	32	32.75
TN	31.5	33.75	36.25	33	33	34.5	36	36.25	37
FP	28.75	27.75	27	31	31	29.75	28.25	28	27.25
FN	42	44.75	44.75	43.75	39.75	41.25	41.5	41.5	41
Precision	0.476	0.473	0.512	0.466	0.507	0.509	0.521	0.523	0.524
Recall	0.375	0.360	0.367	0.388	0.447	0.433	0.430	0.429	0.437
Accuracy	44.8%	45.3%	46.7%	45.1%	48.2%	48.3%	49.2%	49.4%	**50.4%**

Table A.82: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Psychology - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	28.75	30	30.75	32.25	30.5	34.5	32.5	33	35.75
TN	68.25	72.5	74.25	73.75	74.25	76	76	75.75	74.25
FP	27.75	24.75	24.75	26	25.5	24	24.25	24.5	26
FN	68.25	71.75	73	74	76	73.5	76	75.5	73
Precision	0.490	0.527	0.549	0.558	0.519	0.584	0.560	0.590	0.612
Recall	0.289	0.284	0.285	0.294	0.272	0.307	0.284	0.290	0.314
Accuracy	50.2%	51.5%	51.8%	51.4%	50.8%	**53.1%**	52.0%	52.1%	52.6%

Table A.83: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Psychology - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	82.75	92	94.25	97.25	100.7	104	104.2	102.2	104
TN	40.75	44	42	40	37.5	35.75	36.5	36.5	37.25
FP	88.5	88.5	92.5	95.75	98.25	100.5	100.2	100.2	99.5
FN	45	41.5	42	41	37.5	34.75	35	37.75	36.25
Precision	0.482	0.509	0.504	0.503	0.505	0.507	0.508	0.503	0.509
Recall	0.649	0.688	0.690	0.702	0.727	0.748	0.746	0.728	0.739
Accuracy	48.1%	**51.1%**	50.3%	50.1%	50.4%	50.8%	50.9%	50.1%	51.0%

Table A.84: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Psychology - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	67	70	70	71.25	74.25	75	75.75	75.75	73.25
TN	92	95	96.5	95.75	95.25	95.25	96.75	96.25	98.25
FP	73.25	73.75	75	77.75	78.25	78.5	77.75	78.25	76.25
FN	88.75	93.25	97	99.25	97.25	97.75	97.75	98.5	101.2
Precision	0.473	0.480	0.469	0.469	0.482	0.496	0.480	0.476	0.482
Recall	0.424	0.422	0.412	0.412	0.428	0.429	0.431	0.430	0.415
Accuracy	49.5%	**49.6%**	49.1%	48.5%	49.1%	49.1%	49.5%	49.2%	49.1%

Table A.85: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Psychology - 50% train 50% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	3.5	3.75	3.75	3.75	3.75	3.75	3.75	3.75	3.75
TN	0	0	0	0	0	0	0	0	0
FP	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5	14.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.194	0.205	0.205	0.205	0.205	0.205	0.205	0.205	0.205
Recall	1	1	1	1	1	1	1	1	1
Accuracy	19.4%	**20.5%**	**20.5%**	**20.5%**	**20.5%**	**20.5%**	**20.5%**	**20.5%**	**20.5%**

Table A.86: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Statistics - 90% train 10% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	9	9.75	9.75	10.5	10.5	10.5	10.5	10.5	10.5
TN	0	0	0	0	0	0	0	0	0
FP	25.75	26.5	26.5	26.5	26.5	26.5	26.5	26.5	26.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.256	0.267	0.267	0.281	0.281	0.281	0.281	0.281	0.281
Recall	1	1	1	1	1	1	1	1	1
Accuracy	25.6%	26.7%	26.7%	**28.1%**	**28.1%**	**28.1%**	**28.1%**	**28.1%**	**28.1%**

Table A.87: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Statistics - 80% train 20% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	11.5	11.5	11.5	12	12	12	12	12	12
TN	0	0	0	0	0	0	0	0	0
FP	38.25	38.25	38.25	38.25	38.5	38.5	38.5	38.5	38.5
FN	0	0	0	0	0	0	0	0	0
Precision	0.228	0.228	0.228	0.236	0.235	0.235	0.235	0.235	0.235
Recall	1	1	1	1	1	1	1	1	1
Accuracy	22.8%	22.8%	22.8%	23.6%	**23.5%**	**23.5%**	**23.5%**	**23.5%**	**23.5%**

Table A.88: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Statistics - 70% train 30% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	12.5	12.5	12	13	13.25	13.25	13.25	13.5	13.25
TN	1	3.75	4.75	5.5	5	5.25	5.75	4.75	5.25
FP	50.25	48	47	46.25	47.5	47.25	46.75	47.75	47.25
FN	0.25	0.25	0.75	0.5	0.25	0.25	0.25	0	0.25
Precision	0.197	0.202	0.197	0.214	0.214	0.215	0.216	0.217	0.215
Recall	0.968	0.968	0.916	0.944	0.972	0.972	0.972	1.000	0.972
Accuracy	21.1%	25.1%	25.8%	28.2%	27.4%	27.8%	**28.5%**	27.4%	27.8%

Table A.89: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Statistics - 60% train 40% test

	10%	20%	30%	40%	50%	60%	70%	80%	90%
TP	21.5	22	21.75	23	22.75	23.5	23.5	23.5	23.75
TN	4.5	7.25	9.25	11	12.25	11.25	8.75	9	9.75
FP	57.25	56.25	54.25	52.5	51.75	52.75	55.25	55	54.25
FN	0.5	0.75	1	1.25	1.5	0.75	0.75	0.75	0.5
Precision	0.271	0.280	0.288	0.311	0.309	0.312	0.298	0.299	0.307
Recall	0.972	0.958	0.944	0.938	0.924	0.960	0.960	0.960	0.973
Accuracy	31.0%	33.9%	35.9%	38.8%	**39.6%**	39.3%	36.5%	36.8%	37.9%

Table A.90: Classification Results On discriminative latent factors using multi-view and patterns' structure - OLI Statistics - 50% train 50% test

BIBLIOGRAPHY

[1] Rakesh Agrawal and Ramakrishnan Srikant, *Mining sequential patterns*, Proceedings of the eleventh international conference on data engineering, IEEE, 1995, pp. 3–14.

[2] Nil-Jana Akpinar, Aaditya Ramdas, and Umut Acar, *Analyzing student strategies in blended courses using clickstream data*, arXiv preprint arXiv:2006.00421 (2020).

[3] Sattar Ameri, Mahtab J Fard, Ratna B Chinnam, and Chandan K Reddy, *Survival analysis based framework for early prediction of student dropouts*, the 25th ACM International on Conference on Information and Knowledge Management, ACM, 2016, pp. 903–912.

[4] Yuichiro Anzai and Herbert A Simon, *The theory of learning by doing.*, Psychological review **86** (1979), no. 2, 124.

[5] David Azcona, I-Han Hsiao, and Alan F Smeaton, *Detecting students-at-risk in computer programming classes with learning analytics from students' digital footprints*, User Modeling and User-Adapted Interaction **29** (2019), no. 4, 759–788.

[6] Ryan S Baker, *Challenges for the future of educational data mining: The baker learning analytics prizes*, Journal of Educational Data Mining **11** (2019), no. 1, 1–17.

[7] Mina Shirvani Boroujeni and Pierre Dillenbourg, *Discovery and temporal analysis of mooc study patterns*, Journal of Learning Analytics, LAK, 2019, pp. 16–33.

[8] Ahcène Boubekki, Shailee Jain, and Ulf Brefeld, *Mining user trajectories in electronic text books*, the 11th International Conference on Educational Data Mining (EDM 2018), 2018.

[9] Shahab Boumi and Adan Vela, *Application of hidden markov models to quantify the impact of enrollment patterns on student performance.*, International Educational Data Mining Society (2019).

[10] Shahab Boumi, Adan Vela, and Jacquelyn Chini, *Quantifying the relationship between student enrollment patterns and student performance*, arXiv preprint arXiv:2003.10874 (2020).

[11] Sebastien Boyer and Kalyan Veeramachaneni, *Transfer learning for predictive models in massive open online courses*, Lecture Notes in Computer Science, pp. 54–63, Madrid, Spain, 2015.

[12] Luiz Antonio Buschetto Macarini, Cristian Cechinel, Matheus Francisco Batista Machado, Vinicius Faria Culmant Ramos, and Roberto Munoz, *Predicting students success in blended learning—evaluating different interactions inside learning management systems*, Applied Sciences **9** (2019), no. 24, 5523.

[13] Mehmet Celepkolu and Kristy Elizabeth Boyer, *Predicting student performance based on eye gaze during collaborative problem solving*, Proceedings of the Group Interaction Frontiers in Technology, 2018, pp. 1–8.

[14] Hao Cen, Kenneth Koedinger, and Brian Junker, *Learning factors analysis–a general method for cognitive model evaluation and improvement*, International Conference on Intelligent Tutoring Systems, Springer, 2006, pp. 164–175.

[15] Yuejie Chi, Yue M Lu, and Yuxin Chen, *Nonconvex optimization meets low-rank matrix factorization: An overview*, IEEE Transactions on Signal Processing **67** (2019), no. 20, 5239–5269.

[16] Irene-Angelica Chounta and Paulo F Carvalho, *Square it up! how to model step duration when predicting student performance*, Proceedings of the 9th International Conference on Learning Analytics & Knowledge, 2019, pp. 330–334.

[17] Moody Chu, Fasma Diele, Robert Plemmons, and Stefania Ragni, *Optimality, computation, and interpretation of nonnegative matrix factorizations*, SIAM Journal on Matrix Analysis, Citeseer, 2004.

[18] Mi Chunqiao, Peng Xiaoning, and Deng Qingyou, *An artificial neural network approach to student study failure risk early warning prediction based on tensorflow*, International Conference on Advanced Hybrid Information Processing, Springer, 2017, pp. 326–333.

[19] Fisnik Dalipi, Ali Shariq Imran, and Zenun Kastrati, *Mooc dropout prediction using machine learning techniques: Review and research challenges*, 2018 IEEE Global Engineering Education Conference (EDUCON), IEEE, 2018, pp. 1007–1014.

[20] John Dewey, *Democracy and education (1916)*, Jo Ann Boydston (ed.). The Middle Works of John Dewey **9** (1966), 1899–1924.

[21] Shayan Doroudi, Kenneth Holstein, Vincent Aleven, and Emma Brunskill, *Sequence matters, but how exactly? a method for evaluating activity sequences from data.*, International Educational Data Mining Society (2016).

[22] P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, *Fast vertical mining of sequential patterns using co-occurrence information*, Proc. 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2014), Part 1, Springer, LNAI, 8443 (2014), 40–52.

[23] Geoffrey Garrido, Matthew W Guthrie, and Zhongzhou Chen, *How are students' online learning behavior related to their course outcomes in an introductory physics course?*, Physics Education Research Conference 2019, 2020.

[24] Chase Geigle and ChengXiang Zhai, *Modeling student behavior with two-layer hidden markov models*, Journal of Educational Data Mining **9** (2017), no. 1.

[25] Ben Gelman, Matt Revelle, Carlotta Domeniconi, Kalyan Veeramachaneni, and Aditya Johri, *Acting the same differently: A cross-course comparison of user behavior in moocs*, the 9th International Conference on Educational Data Mining (EDM 2016) (Tiffany Barnes, Min Chi, and Mingyu Feng, eds.), 2016, pp. 376–381.

[26] Niki Gitinabard, Sarah Heckman, Tiffany Barnes, and Collin F. Lynch, *What will you do next? a sequence analysis on the student transitions between online platforms in blended courses*, the 12th International Conference on Educational Data Mining, 2019, pp. 59–68.

[27] Niki Gitinabard, Farzaneh Khoshnevisan, Collin F Lynch, and Elle Yuan Wang, *Your actions or your associates? predicting certification and dropout in moocs with behavioral and social features.*, International Conference on Educational Data Mining, ERIC, 2018.

[28] Julio Guerra, Shaghayegh Sahebi, Yu-Ru Lin, and Peter Brusilovsky, *The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises*, 2014, pp. 153–160.

[29] Christian Hansen, Casper Hansen, Niklas Hjuler, Stephen Alstrup, and Christina Lioma, *Sequence modelling for analysing student interaction with educational systems*, the 10th International Conference on Educational Data Mining, 2017, pp. 232–237.

[30] Jiazhen He, James Bailey, Benjamin IP Rubinstein, and Rui Zhang, *Identifying at-risk students in massive open online courses*, Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015.

[31] Khe Foon Hew, Xiang Hu, Chen Qiao, and Ying Tang, *What predicts student satisfaction with moocs: A gradient boosting trees supervised machine learning and sentiment analysis approach*, Computers & Education **145** (2020), 103724.

[32] Roya Hosseini, Peter Brusilovsky, Michael Yudelson, and Arto Hellas, *Stereotype modeling for problem-solving performance predictions in moocs and traditional courses*, Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization, 2017, pp. 76–84.

[33] I-Han Hsiao and Peter Brusilovsky, *Motivational social visualizations for personalized e-learning*, European Conference on Technology Enhanced Learning, Springer, 2012, pp. 153–165.

[34] I-Han Hsiao, Sergey Sosnovsky, and Peter Brusilovsky, *Adaptive navigation support for parameterized questions in object-oriented programming*, European Conference on Technology Enhanced Learning, Springer, 2009, pp. 88–98.

[35] Hui-Chun Hung, I-Fan Liu, Che-Tien Liang, and Yu-Sheng Su, *Applying educational data mining to explore students' learning patterns in the flipped learning approach for coding education*, Symmetry **12** (2020), no. 2, 213.

[36] Suhang Jiang, Adrienne Williams, Katerina Schenke, Mark Warschauer, and Diane O'dowd, *Predicting mooc performance with week 1 behavior*, Educational data mining 2014, 2014.

[37] Jelena Jovanović, Dragan Gašević, Shane Dawson, Abelardo Pardo, and Negin Mirriahi, *Learning analytics to unveil learning strategies in a flipped classroom*, The Internet and Higher Education **33** (2017), no. 4, 74–85.

[38] Hassan Khosravi, Ali Bozorgkhan, and Oliver Schulte, *Transaction-based link strength prediction in a social network*, 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, 2013, pp. 191–198.

[39] Hannah Kim, Jaegul Choo, Jingu Kim, Chandan K Reddy, and Haesun Park, *Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization*, the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 567–576.

[40] Jinho Kim, Jung Yeon Park, and Wim Van Den Noortgate, *On the use of bayesian probabilistic matrix factorization for predicting student performance in online learning environments*, (2020).

[41] John S Kinnebrew, James R Segedy, and Gautam Biswas, *Analyzing the temporal evolution of students' behaviors in open-ended learning environments*, Metacognition and learning **9** (2014), no. 2, 187–215.

[42] Ren F Kizilcec, Chris Piech, and Emily Schneider, *Deconstructing disengagement: analyzing learner subpopulations in massive open online courses*, Proceedings of the third international conference on learning analytics and knowledge, 2013, pp. 170–179.

[43] René F Kizilcec and Sherif Halawa, *Attrition and achievement gaps in online learning*, Proceedings of the Second (2015) ACM Conference on Learning@ Scale, 2015, pp. 57–66.

[44] Marius Kloft, Felix Stiehler, Zhilin Zheng, and Niels Pinkwart, *Predicting mooc dropout over weeks using machine learning methods*, Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs, 2014, pp. 60–65.

[45] Kenneth R Koedinger, Ryan SJd Baker, Kyle Cunningham, Alida Skogsholm, Brett Leber, and John Stamper, *A data repository for the edm community: The pslc datashop*, Handbook of educational data mining **43** (2010), 43–56.

[46] Kenneth R Koedinger, Jihee Kim, Julianna Zhuxin Jia, Elizabeth A McLaughlin, and Norman L Bier, *Learning is not a spectator sport: Doing is better than watching for learning from a mooc*, Proceedings of the second (2015) ACM conference on learning@ scale, 2015, pp. 111–120.

[47] Yehuda Koren, Robert Bell, and Chris Volinsky, *Matrix factorization techniques for recommender systems*, Computer (2009), no. 8, 30–37.

[48] Mukesh Kumar, AJ Singh, and Disha Handa, *Literature survey on student's performance prediction in education using data mining techniques*, International Journal of Education and Management Engineering **7** (2017), no. 6, 40–49.

[49] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan, *Disclda: Discriminative learning for dimensionality reduction and classification*, Advances in neural information processing systems, 2009, pp. 897–904.

[50] Andrew S Lan, Christopher G Brinton, Tsung-Yen Yang, and Mung Chiang, *Behavior-based latent variable model for learner engagement.*, International Educational Data Mining Society (2017).

[51] V Levenshtein, *Leveinshtein distance*, 1965.

[52] Chih-Jen Lin, *Projected gradient methods for nonnegative matrix factorization*, Neural computation **19** (2007), no. 10, 2756–2779.

[53] Jianhua Lin, *Divergence measures based on the shannon entropy*, IEEE Transactions on Information theory **37** (1991), no. 1, 145–151.

[54] Stephan Lorenzen, Ninh Pham, and Stephen Alstrup, *On predicting student performance using low-rank matrix factorization techniques*, European Conference on e-Learning, Academic Conferences International Limited, 2017, pp. 326–334.

[55] Owen HT Lu, Anna YQ Huang, Jeff CH Huang, Albert JQ Lin, Hiroaki Ogata, and Stephen JH Yang, *Applying learning analytics for the early prediction of students' academic performance in blended learning*, Journal of Educational Technology & Society **21** (2018), no. 2, 220–232.

[56] R Martinez, Kalina Yacef, and Judy Kay, *Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop*, Educational Data Mining 2011 (2011), no. June 2010, 111–120.

[57] Mehrdad Mirzaei, Shaghayegh Sahebi, and Peter Brusilovsky, *Annotated examples and parameterized exercises: Analyzing students' behavior patterns*, International Conference on Artificial Intelligence in Education, Springer, 2019, pp. 308–319.

[58] Carl H Mooney and John F Roddick, *Sequential pattern mining–approaches and algorithms*, ACM Computing Surveys (CSUR) **45** (2013), no. 2, 1–39.

[59] Kousuke Mouri, Atsushi Shimada, Chengjiu Yin, and Keiichi Kaneko, *Discovering hidden browsing patterns using non-negative matrix factorization*, the 11th International Conference on Educational Data Mining, (EDM 2018), 2018, pp. 568–571.

[60] Kousuke Mouri, Fumiya Suzuki, Atsushi Shimada, Noriko Uosaki, Chengjiu Yin, Keiichi Kaneko, and Hiroaki Ogata, *Educational data mining for discovering hidden browsing patterns using non-negative matrix factorization*, Interactive Learning Environments (2019).

[61] Ding Mucong, Yanbang Wang, Erik Hemberg, and Una-May O'Reilly, *Transfer learning using representation learning in massive open online courses*, In Proceedings of the 9th International Conference on Learning Analytics & Knowledge, LAK, 2019, pp. 145–154.

[62] Prema Nedungadi and TK Smruthy, *Personalized multi-relational matrix factorization model for predicting student performance*, Intelligent Systems Technologies and Applications, Springer, 2016, pp. 163–172.

[63] Andrew Y Ng, Michael I Jordan, and Yair Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in neural information processing systems, 2002, pp. 849–856.

[64] Jennifer K Olsen, Vincent Aleven, and Nikol Rummel, *Predicting student performance in a collaborative learning environment.*, International Educational Data Mining Society (2015).

[65] V Paul Pauca, Farial Shahnaz, Michael W Berry, and Robert J Plemmons, *Text mining using non-negative matrix factorizations*, SIAM International Conference on Data Mining, 2004, pp. 452–456.

[66] Agoritsa Polyzou and George Karypis, *Feature extraction for next-term prediction of poor student performance*, IEEE Transactions on Learning Technologies **12** (2019), no. 2, 237–248.

[67] Keshav Singh Rawat and I.V. Malhan, *A hybrid classification method based on machine learning classifiers to predict performance in educational data mining*, In Proceedings of 2nd International Conference on Communication, Computing and Networking, ICNCC, 2019, pp. 677–684.

[68] Zhiyun Ren, Xia Ning, and Huzefa Rangwala, *Ale: Additive latent effect models for grade prediction*, Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM, 2018, pp. 477–485.

[69] Joseph Rollinson and Emma Brunskill, *From predictive models to instructional policies.*, International Educational Data Mining Society (2015).

[70] JPJ Rousseeuw, *A graphical aid to the interpretation and validation of cluster analysis*, Journal of Computational Application Math (1989).

[71] Shaghayegh Sahebi, Yu-Ru Lin, and Peter Brusilovsky, *Tensor factorization for student modeling and performance prediction in unstructured domain*, the 9th International Conference on Educational Data Mining, 2016, pp. 502–505.

[72] Robert Sawyer, Jonathan Rowe, Roger Azevedo, and James Lester, *Filtered time series analyses of student problem-solving behaviors in game-based learning*, the 11th International Conference on Educational Data Mining (EDM 2018), 2018.

[73] Roger C Schank, Tamara R Berman, and Kimberli A Macpherson, *Learning by doing*, Instructional-design theories and models: A new paradigm of instructional theory **2** (1999), no. 2, 161–181.

[74] Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons, *Document clustering using nonnegative matrix factorization*, Information Processing & Management **42** (2006), no. 2, 373–386.

[75] Junming Shao, Zhongjing Yu, Peiyan Li, Wei Han, Christian Sorg, and Qinli Yang, *Exploring common and distinct structural connectivity patterns between schizophrenia and major depression via cluster-driven nonnegative matrix factorization*, 2017 IEEE International Conference on Data Mining (ICDM), IEEE, 2017, pp. 1081–1086.

[76] Kshitij Sharma, Patrick Jermann, and Pierre Dillenbourg, *Identifying styles and paths toward success in moocs*, the 8th International Conference on Educational Data Mining (EDM 2015) (Olga Santos, Jesús G. Boticario, Cristobal Romero, Mykola Pechenizkiy, Agathe Merceron, Piotr Mitros, José María Luna, Cristian Mihaescu, Pablo Moreno, Arnon Hershkovitz, Sebastian Ventura, and Michel Desmarais, eds.), 2015.

[77] Wen-Chung Shih, *Mining sequential patterns to explore users' learning behavior in a visual programming app*, 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 2, IEEE, 2018, pp. 126–129.

[78] Mina Shirvani Boroujeni and Pierre Dillenbourg, *Discovery and temporal analysis of latent study patterns in mooc interaction sequences*, In Proceedings of the 8th International Conference on Learning Analytics and Knowledge, LAK, 2018, pp. 206–215.

[79] Sergey Sosnovsky, Peter Brusilovsky, Danielle H Lee, Vladimir Zadorozhny, and Xin Zhou, *Re-assessing the value of adaptive navigation support in e-learning context*, International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer, 2008, pp. 193–203.

[80] Martin Stapel, Zhilin Zheng, and Niels Pinkwart, *An ensemble method to predict student performance in an online math learning environment.*, International Educational Data Mining Society (2016).

[81] Yuta Taniguchi, Daiki Suehiro, Atsushi Shimada, and Hiroaki Ogata, *Revealing hidden impression topics in students' journals based on nonnegative matrix factorization*, 2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT), IEEE, 2017, pp. 298–300.

[82] Colin Taylor, Kalyan Veeramachaneni, and Una-May O'Reilly, *Likely to stop? predicting stopout in massive open online courses*, arXiv preprint arXiv:1408.3382 (2014).

[83] Nguyen Thai-Nghe, Lucas Drumond, Tomáš Horváth, Lars Schmidt-Thieme, et al., *Multi-relational factorization models for predicting student performance*, KDD Workshop on Knowledge Discovery in Educational Data (KDDinED), 2011.

[84] Nikola Tomasevic, Nikola Gvozdenovic, and Sanja Vranes, *An overview and comparison of supervised data mining techniques for student exam performance prediction*, Computers & Education **143** (2020), 103676.

[85] Maria Tsiakmaki, Georgios Kostopoulos, Giannis Koutsonikos, Christos Pierrakeas, Sotiris Kotsiantis, and Omiros Ragos, *Predicting university students' grades based on previous academic achievements*, 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA), IEEE, 2018, pp. 1–6.

[86] Anouschka van Leeuwen, Nynke Bos, Heleen van Ravenswaaij, and Jurgen van Oostenrijk, *The role of temporal patterns in students' behavior for predicting course performance: A comparison of two blended learning courses*, British Journal of Educational Technology **50** (2019), no. 2, 921–933.

[87] Liujing Wang, Naiyang Guan, Dianxi Shi, Zunlin Fan, and Longfei Su, *Robust semi-supervised non-negative matrix factorization with structured normalization*, IEEE Access **7** (2019), 133996–134013.

[88] X Wen, K Pelechrinis, Y Lin, X Liu, Y Ahn, and N Cao, *Facit: Factorizing tensors into interpretable and scrutinizable patterns*, 2019 IEEE Visualization Conference (VIS), 2019, pp. 1–5.

[89] Xidao Wen, Yu-Ru Lin, Xi Liu, Peter Brusilovsky, and Jordan Barría Pineda, *Iterative discriminant tensor factorization for behavior comparison in massive open online courses*, The World Wide Web Conference, 2019, pp. 2068–2079.

[90] Jacob Whitehill, Joseph Williams, Glenn Lopez, Cody Coleman, and Justin Reich, *Beyond prediction: First steps toward automatic intervention in mooc student stopout*, the 8th International Conference on Educational Data Mining (EDM 2015), 2015.

[91] Wanli Xing, Rui Guo, Eva Petakovic, and Sean Goggins, *Participation-based student final performance prediction model through interpretable genetic programming: Integrating learning analytics, educational data mining and theory*, Computers in Human Behavior **47** (2015), 168–181.

[92] Jie Xu, Kyeong Ho Moon, and Mihaela Van Der Schaar, *A machine learning approach for tracking and predicting student performance in degree programs*, IEEE Journal of Selected Topics in Signal Processing **11** (2017), no. 5, 742–753.

[93] Wei Xu, Xin Liu, and Yihong Gong, *Document clustering based on non-negative matrix factorization*, the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, 2003, pp. 267–273.

[94] Chengjiu Yin, Fumiya Okubo, Atsushi Shimada, Misato Oi, Sachio Hirokawa, and Hiroaki Ogata, *Identifying and analyzing the learning behaviors of students using e-books*, 23rd International Conference on Computers in Education, ICCE 2015, Asia-Pacific Society for Computers in Education, 2015.

[95] Leila Zahedi and Stephanie J Lunn, *Leveraging machine learning techniques to analyze computing persistence in undergraduate programs*, 2020 ASEE Virtual Annual Conference Content Access (2020).

[96] Ms Leila Zahedi, *Implications of gamification in learning environments on computer science students: A comprehensive study*, 126th Annual Conference and Exposition of American Society for Engineering Education, 2019.

[97] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang, *Collective classification via discriminative matrix factorization on sparsely labeled networks*, Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 1563–1572.

[98] Yupei Zhang, Yue Yun, Huan Dai, Jiaqi Cui, and Xuequn Shang, *Graphs regularized robust matrix factorization and its application on student grade prediction*, Applied Sciences **10** (2020), no. 5, 1755.

[99] Zhong Zhang, Zhili Qin, Peiyan Li, Qinli Yang, and Junming Shao, *Multi-view discriminative learning via joint non-negative matrix factorization*, International Conference on Database Systems for Advanced Applications, Springer, 2018, pp. 542–557.

[100] Jun Zhu, Amr Ahmed, and Eric P Xing, *Medlda: maximum margin supervised topic models*, the Journal of machine Learning research **13** (2012), no. 1, 2237–2278.