The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Detecting Trait versus Performance Student Behavioral Patterns Using Discriminative Non-Negative Matrix Factorization

Mehrdad Mirzaei,[1] **Shaghayegh Sahebi,**[1] **Peter Brusilovsky**[2]
[1] Department of Computer Science, University at Albany - SUNY, Albany, New York 12203
[2] School of Computing and Information, University of Pittsburgh, Pittsburgh, Pennsylvania 15260
{mmirzaei, ssahebi}@albany.edu, peterb@pitt.edu

Abstract

Recent studies have shown that students follow stable behavioral patterns while learning in online educational systems. These 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. In this work, 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 represent each student's behavior as a vector of high-support sequential micro-patterns. Then, 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.

Introduction

In many online learning environments, students have the freedom to access learning materials, repeatedly, in any order, and at their own pace. With fewer restrictions, a variety of interaction sequences emerge as learners work with such systems. For example, in an interaction session, a student may start by studying some reading material for a while, then move on to solving relevant problems, and eventually, take a quiz before leaving the system. Recent studies on extracting behavioral patterns from these sequences have shown that students follow stable behavioral patterns while working with these systems (Guerra et al. 2014; Mirzaei, Sahebi, and Brusilovsky 2019; Gitinabard et al. 2019; Wen et al. 2019). For example, some students tend to study the reading materials, while others are more interested in learning by solving problems (Mirzaei, Sahebi, and Brusilovsky 2019). In addition to learning patterns, some studies have discovered inefficient learning behaviors in student sequences. For example, Guerra et al. 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 (Guerra et al. 2014). 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 work, we mine the trait versus performance behavioral patterns in students by summarizing student sequences as frequent micro-pattern vectors, 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 a real-word dataset of sequences from students interacting with an online programming tutoring system, with two different learning material types: problems, and worked-examples. 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.

Related Work

With the amount of information from students' interaction log in online educational systems increasingly growing, it is compelling for researchers utilizing this data to study and improve the learning process. Such data can be utilized to model students' behavior while interacting with online courses. Students' behavior from log data are used to predict students' performance (Xing et al. 2015) to either intervene the student and avoid failure or encourage them to pursue productive behaviors (Chunqiao, Xiaoning, and Qingyou 2017). Another usage is to predict dropout in online open-access courses (Boyer and Veeramachaneni 2015;

Whitehill et al. 2015; Ameri et al. 2016).

Sequence mining has been widely used in educational researches to study students' activities in online systems. Exploratory sequence analysis of students' actions could unveil learning strategies in flipped classes (Jovanović et al. 2017). This method helps instructors to design courses and scaffolds. Students can also take advantage of the approach to improve their learning behaviors. Analyzing the sequence of transitions between online platforms in (Gitinabard et al. 2019) has shown meaningful patterns that are helpful for both instructors and students. Mining students' sequential patterns of actions is used in (Maldonado et al. 2010) to extract students' behavioral patterns while interacting around a tabletop. They used the patterns to distinguish between high achievement and low achievement groups. Previous researches have shown that students' behaviors can impact their performance since the behavior could be productive or non-productive. In (Guerra et al. 2014) the patterns are extracted using sequential pattern mining methods from interaction with exercises and in (Mirzaei, Sahebi, and Brusilovsky 2019) the patterns are extracted from interaction with multiple learning materials. In those researches, distinctive patterns are recognized for each group, however, there are some patterns that are common among all students that should be taken into account.

Matrix factorization methods have been introduced in recommendation systems (Koren, Bell, and Volinsky 2009) and widely used in other areas such as document clustering (Kim et al. 2015; Xu, Liu, and Gong 2003; Shahnaz et al. 2006; Pauca et al. 2004). In (Mouri et al. 2019) non-negative Matrix Factorization (NMF) is used to detect high-performance learners' browsing patterns from the collected log data to increase students' thinking skills. Algorithm DICS in (Zhang et al. 2018) exploits the relationships in different views to build a classifier. This approach uses joint NMF to explore discriminative and non-discriminative information existing in common and specific sections among multiple views. Another way of representing students' behaviors are by using tensors. Tensor-based methods are used to model students' behavior and predict their performance (Sahebi, Lin, and Brusilovsky 2016). In (Wen et al. 2019) multi-way interactions are considered as behavior and common and discriminative patterns are discovered with a framework of iterative discriminant factorization.

Joint discriminative non-negative matrix factorization has been used previously in (Kim et al. 2015) to discover common and distinctive topics in documents. Their topic modeling method simultaneously finds common and distinct topics from multiple datasets. We apply this approach to detect common and distinct extracted patterns from students' sequential behaviors with different performances.

Dataset

Our dataset is collected from an online tutoring system that includes programming problems and worked examples. Students are free to choose the problems they would like to work on, and the examples they would like to study in any order. Each programming problem is a multiple-choice or a short-answer question, presenting a code snippet to students and asking for the results of executing that code. The students can repeat answering to the same problem multiple times. However, every time simple code parameters, such as variable values, change and as a result, the correct answer to that problem changes. The annotated examples are code snippets that include natural language explanations for different lines of code. Our collected data includes every student's sequence of activities, in the form of problem or example identifiers, if the student's answer to the problem is correct (success) or incorrect (failure), and the time the student spends on each activity. Each problem or worked-example in the dataset is assigned to a specific course topic. Additionally, students' prior knowledge in the material (as pre-test scores) and knowledge at the end of the course (as post-test scores) are available in the dataset. The dataset includes 83 student activity sequences on 103 problems and 42 examples. 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%.

Discriminative Learning of Student Behavior

In this section, we describe the process of extracting patterns from student learning behaviors. An illustration of our framework is presented in Figure 1.

In summary, our framework follows the following steps:

1. coding student activity and constructing student sequences;
2. building student pattern matrices; and
3. finding discriminative vs. common patterns between high- and low-performing students.

In the following sections, we describe each of these steps.

Constructing Student Activity Sequences

In this part, we follow the work of Mirzaei et al. to code student activity sessions based on activity attempts' type, outcome, and duration (Mirzaei, Sahebi, and Brusilovsky 2019). Table 1 shows a short description of all attempt labels.

Attempt type. Since students can work with various types of learning material (in our case, problems and worked-examples), we code activities based on the learning material type. Specifically, for worked examples we use the letters "e" or "E", and for problems, we use the letters "s", "S", "f", or "F" according to outcome and duration.

Attempt Outcome. Attempting to solve problems can have different outcomes. In our case, students can have a correct (success) or incorrect (failure) answer. We code each kind of feedback with different letters: a student's successful outcome is presented with "s" or "S", and the unsuccessful one is presented with "f" or "F".

Attempt Duration. We code the time spent on each attempt for each learning material as a short (represented by lowercase letters, like "s") or long (represented by capital letters, like "S") attempt. To determine if an activity should be categorized as short vs. long, we compare the time taken on the

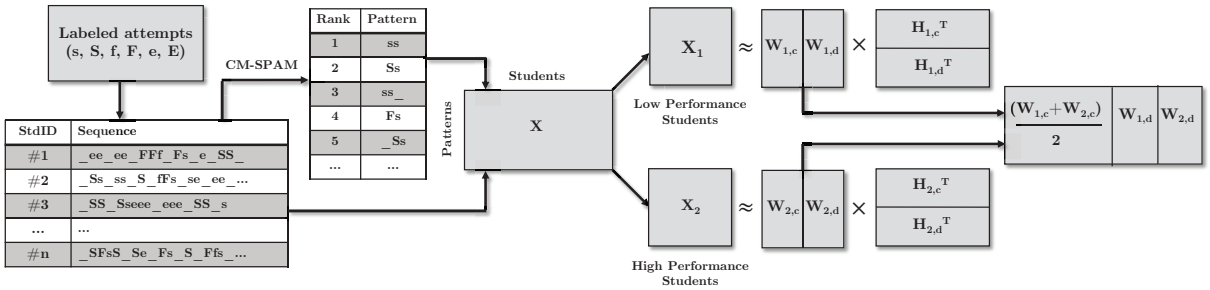

Figure 1: Most frequent patterns are extracted from sequences by CM-SPAM. These patterns are rows of matrix X and students are columns. We split matrix X based on the performance of the students to X_1 and X_2. Then with discriminative non-negative matrix factorization, common and distinct patterns are extracted.

Label	Attempt	Label	Attempt
S	Long Success	s	Short Success
F	Long Failure	f	Short Failure
E	Long Example	e	Short Example

Table 1: Attempt coding labels

activity in this attempt with the median time-taken on this activity across all attempts of all students. If this attempt takes longer than the median time, the attempt is coded as a long attempt; Otherwise, it is coded as a short one.

Each student can attempt learning materials from various topics in any order. Using the assigned learning material topic, we separate student activity sequences into multiple *topic sessions*. A new topic session starts when the student moves to a different topic, meaning that all student activities within a session focus on the same topic. To indicate the start and the end of each session, we use a special symbol "_". For instance, "_Fse_" is a student session that starts with working on a problem for a long time and failing at it, then working on the problem again for a short time and succeeding in it, and finally moving on to studying an example for a short time.

Building the Pattern Matrix

Following the work of Guerra et al., in this part we use the coded student sequences to build students' micro-pattern vectors (Guerra et al. 2014). More specifically, we extract high-frequent micro-patterns from the coded sequences, and then build student pattern vectors based on those frequent micro-patterns.

For the first step, we use CM-SPAM (Fournier-Viger et al. 2014), a sequential pattern mining algorithm, to find the frequent micro-patterns with minimum support of 5.4%. We choose this minimum support to keep the most important patterns, while maintaining an adequate statistical power for the experiments. Then, we discard the short patterns, or the ones with length less than two, as they do not convey a sequential notion. This leads to 77 different frequent micro-patterns. For the second step, we use these 77 most frequent micro-patterns as features to build student pattern vectors. For each student, we calculate the normalized frequency of each micro-pattern in their complete coded sequence. The normalization is done such that the sum of values for micro-patterns for each student equals to one. This normalization compensates for students having various sequences lengths and allows the student vectors to be on the same scale. We can then build a pattern matrix that represents all student behaviors by concatenating their normalized micro-pattern vectors.

Discriminative Non-negative Factorization of Patterns

Our main goal in this work 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 as:

$$\text{normalized-learning-gain}_s = \frac{\text{post-test}_s - \text{pre-test}_s}{\text{max-post-test} - \text{min-pre-test}}$$

in which max-post-test and min-pre-test are the maximum and minimum possible scores in post-test and pre-test, respectively. We group the top 40% ($n = 29$) of students with the highest normalized learning gain as high-performing students, and the bottom 40% ($n = 26$) as low-performing students. We leave out the students in the middle (20%) to achieve better discrimination between student performances in the two high and low groups.

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

To verify this distinction between different sets of patterns, we apply discriminative non-negative matrix factor-

ization (Kim et al. 2015) that was proposed for discriminatory topic modeling in documents. To do this, we split the pattern-student matrix X, built in previous section based on the students' performance to achieve matrix X_1 for low-performing students, and X_2 for low-performing ones. 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:

$$L = \left\| X_1 - W_1 H_1^T \right\|_F^2 + \left\| X_2 - W_2 H_2^T \right\|_F^2 \qquad (1)$$

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:

$$W_1 = [W_{1,c} \quad W_{1,d}], \quad W_2 = [W_{2,c} \quad W_{2,d}]$$
$$H_1 = \begin{bmatrix} H_{1,c} \\ H_{1,d} \end{bmatrix} \quad H_2 = \begin{bmatrix} H_{2,c} \\ H_{2,d} \end{bmatrix} \qquad (2)$$

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 (3).

$$f_c(W_{1,c}, W_{2,c}) = \left\| W_{1,c} - W_{2,c} \right\|_F^2$$
$$f_d(W_{1,d}, W_{2,d}) = \left\| W_{1,d}^T W_{2,d} \right\|_F^2 \qquad (3)$$

Eventually, considering regularization on W and H for generalizability purposes, we will minimize the objective function in Equation (4), with respect to W and H, and constraining them to be non-negative, using Gradient Descent (GD) algorithm.

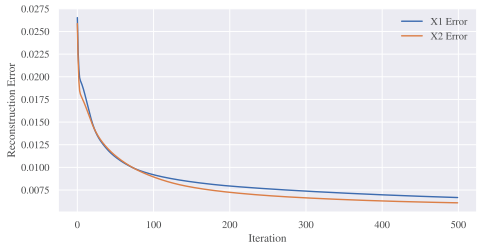

Figure 2: Reconstruction error (RMSE) of $\left\| X_1 - W_1 H_1^T \right\|$ and $\left\| X_2 - W_2 H_2^T \right\|$ with 500 iterations

$$L = \left\| X_1 - W_1 H_1^T \right\|_F^2 + \left\| X_2 - W_2 H_2^T \right\|_F^2 + \alpha \left\| W_{1,c} - W_{2,c} \right\|_F^2 + \beta \left\| W_{1,d}^T W_{2,d} \right\|_F^2 + \gamma(\|W\|^2 + \|H\|^2) \qquad (4)$$

Experiments

Finding Pattern Latent Vectors

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 $K = 15$, $K_c = 10$, and $K_d = 5$. In Figure 2, we show the convergence of the GD algorithm in reconstructing X_1 and X_2 in the first 500 iterations.

The discovered latent factors for each pattern are shown in Figure 3. 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. 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 cluster in previous literature (Mirzaei, Sahebi, and Brusilovsky 2019).

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.

Clustering Patterns

To further understand the students' learning behavior trait and performance patterns, we cluster these patterns, according to the discovered latent factors, into different groups using the spectral clustering algorithm, to find 6 different clusters. The horizontal bars in Figure 3 are dividing the patterns into the discovered clusters. 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 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. (Guerra et al. 2014). We see similar trait-based clusters within low-performance patterns: the second and fourth sets of patterns in Figure 3.

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 Figure 4. 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. 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. Using these observations, we can use the same latent factors to predict students' performance in our future work.

Conclusions

In this paper, we proposed 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-

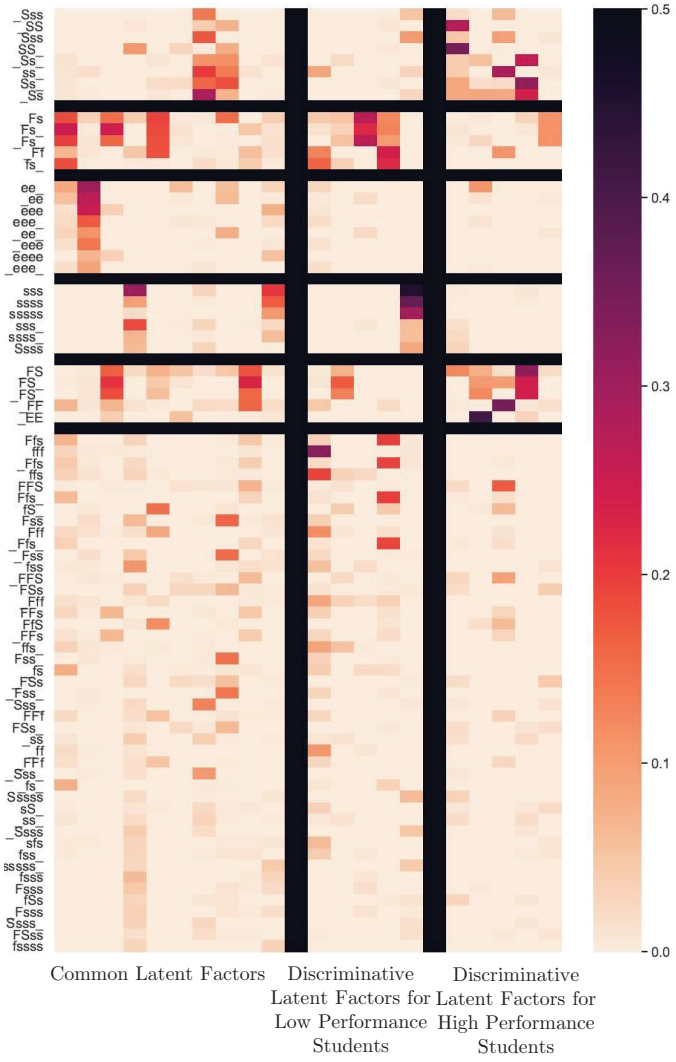

Figure 3: Heatmap shows the distribution of latent factors in common and discriminative parts.

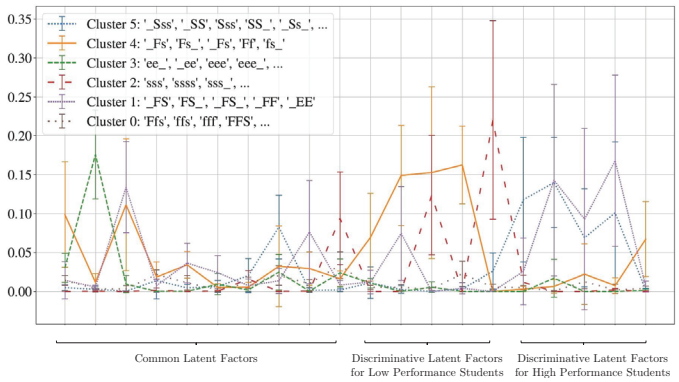

Figure 4: Latent factors for 6 clusters and respective patterns

performance students either repeat their success if they have achieved it by spending a longer 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. In the future, we would like to study the predictive power of the discovered latent factors.

References

Ameri, S.; Fard, M. J.; Chinnam, R. B.; and Reddy, C. K. 2016. Survival analysis based framework for early prediction of student dropouts. In *the 25th ACM International on Conference on Information and Knowledge Management*, 903–912. ACM.

Boyer, S., and Veeramachaneni, K. 2015. Transfer learning for predictive models in massive open online courses. In *International conference on artificial intelligence in education*, 54–63. Springer.

Chunqiao, M.; Xiaoning, P.; and Qingyou, D. 2017. An artificial neural network approach to student study failure risk early warning prediction based on tensorflow. In *International Conference on Advanced Hybrid Information Processing*, 326–333. Springer.

Fournier-Viger, P.; Gomariz, A.; Campos, M.; and Thomas, R. 2014. Fast vertical mining of sequential patterns using co-occurrence information. In *Advances in Knowledge Discovery and Data Mining*, 40–52. Springer.

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

Guerra, J.; Sahebi, S.; Lin, Y.-R.; and Brusilovsky, P. 2014. The problem solving genome: Analyzing sequential patterns of student work with parameterized exercises. 153–160.

Jovanović, J.; Gašević, D.; Dawson, S.; Pardo, A.; and Mirriahi, N. 2017. Learning analytics to unveil learning strategies in a flipped classroom. *The Internet and Higher Education* 33(4):74–85.

Kim, H.; Choo, J.; Kim, J.; Reddy, C. K.; and Park, H. 2015. Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization. In *the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 567–576. ACM.

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

Maldonado, R. M.; Yacef, K.; Kay, J.; Kharrufa, A.; and Al-Qaraghuli, A. 2010. Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In *Educational Data Mining 2011*.

Mirzaei, M.; Sahebi, S.; and Brusilovsky, P. 2019. Annotated examples and parameterized exercises: Analyzing students' behavior patterns. In *International Conference on Artificial Intelligence in Education*, 308–319. Springer.

Mouri, K.; Suzuki, F.; Shimada, A.; Uosaki, N.; Yin, C.; Kaneko, K.; and Ogata, H. 2019. Educational data mining for discovering hidden browsing patterns using non-negative matrix factorization. *Interactive Learning Environments*.

Pauca, V. P.; Shahnaz, F.; Berry, M. W.; and Plemmons, R. J. 2004. Text mining using non-negative matrix factorizations. In *SIAM International Conference on Data Mining*, 452–456.

Sahebi, S.; Lin, Y.-R.; and Brusilovsky, P. 2016. Tensor factorization for student modeling and performance prediction in unstructured domain. In *the 9th International Conference on Educational Data Mining*, 502–505.

Shahnaz, F.; Berry, M. W.; Pauca, V. P.; and Plemmons, R. J. 2006. Document clustering using nonnegative matrix factorization. *Information Processing & Management* 42(2):373–386.

Wen, X.; Lin, Y.-R.; Liu, X.; Brusilovsky, P.; and Barría Pineda, J. 2019. Iterative discriminant tensor factorization for behavior comparison in massive open online courses. In *The World Wide Web Conference*, 2068–2079.

Whitehill, J.; Williams, J.; Lopez, G.; Coleman, C.; and Reich, J. 2015. Beyond prediction: First steps toward automatic intervention in mooc student stopout. In *the 8th International Conference on Educational Data Mining*.

Xing, W.; Guo, R.; Petakovic, E.; and Goggins, S. 2015. Participation-based student final performance prediction model through interpretable genetic programming: Integrating learning analytics, educational data mining and theory. *Computers in Human Behavior* 47:168–181.

Xu, W.; Liu, X.; and Gong, Y. 2003. Document clustering based on non-negative matrix factorization. In *the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 267–273. ACM.

Zhang, Z.; Qin, Z.; Li, P.; Yang, Q.; and Shao, J. 2018. Multi-view discriminative learning via joint non-negative matrix factorization. In *International Conference on Database Systems for Advanced Applications*, 542–557. Springer.