Independent Study Report
Problem Solving Genome: modeling students by their common patterns on problem solving

Julio Guerra

ABSTRACT

Parameterized exercises present an interesting opportunity and challenge for educational data mining. Since this kind of exercise can be repeated many times with different parameters, it is possible to examine student's problem solving behavior on a deeper level. Preliminary analysis exposed unexpected patterns such as repeated successful and failed attempts to solve the same problem. To understand these behaviors we propose the Problem Solving Genome analysis framework. We first characterize attempts in terms of correctness (correct/incorrect) and time (short/long) and use pattern mining techniques to extract attempts' patterns that we call problem solving genes. Then, we build profiles of students, called problem solving genome, based on the frequencies of these patterns. By analyzing the similarity of problem solving genes within students, we show that the problem solving genome reflects consistent student behavior, even across exercises of different complexity level. Surprisingly, we find that the students' genomes persist across different performance groups. A detailed look at the genes within clusters of students having similar genomes suggests that a particular set of patterns are responsible for the difference of performance among students. The results reveal considerable potential of using problem solving genome analysis to understand student behavior at the macroscopic level and suggest new directions for the user modelling and adaptation.

1. Introduction

Parameterized exercises have recently emerged as an important tool for online assessment and learning. A parameterized exercise is essentially an exercise template that is instantiated at runtime with randomly generated parameters. As a result, a single template is able to produce a large number of similar, but distinct questions. While parameterized questions are considerably harder to implement than traditional "static" questions, the benefits offered by this technology make this additional investment worthwhile. During assessment, a reasonably small number of question templates can be used to produce online individualized assessments for large classes minimizing cheating problems [14]. In a selfassessment context, the same question can be used again and again with different parameters, allowing every student to achieve understanding and mastery.

The above mentioned properties of parameterized exercises made them very attractive for the large-scale online learning context. In turn, it made platforms that supported parameterized questions such as LON-CAPA [14] or edX very popular for college-offered online learning and MOOCs. At the same time, parameterized exercises as a learning technology have its own problems. Our experience with personalized exercises for SQL [21] and Java [9] in the self-assessment context demonstrated that the important ability to try the same question again and again is not always beneficial, especially for students who are not good in managing their learning. The analysis of a large number of student logs revealed some considerable number of unproductive repetitions. We observed many cases where students keep solving the same exercise correctly again and again with different parameters, well past the point when it can offer any educational benefit. While it might increase self-confidence, students time might be spent better by advancing to more challenging questions. We can also observe cases where students persist in failing to solve the same, too difficult exercise, instead of focusing on filling the apparent knowledge gap or switching to simpler exercises.

The work presented in this paper was motivated by our belief that the educational value of parameterized exercises could be increased by a personalized guidance mechanism that can predict non-productive behavior and intercept it by recommending a more efficient learning path. Main challenge with predicting unproductive behavior is a stability of behavior patterns in the problem solving process. If the patterns, such as specific unproductive sequences, appear at random, there is a little chance to predict and prevent them. If, on the contrary, specific patterns are associated with some student features (such as knowledge and individual traits), exercise complexity, or the learning process stage, there is a good chance to learn the association rules and use it for prediction. In this paper we performed an extended study of problem solving patterns in the context of parameterized exercises. We explored the connection between these patterns and the components of the learning process mentioned above. Our study produced a rather unusual result. While it was more plausible to expect that patterns are related to the current level of student knowledge, our data pointed that patterns are related to a student as a person. More exactly, we discovered that every student has a specific combinaion of micro-patterns, a kind of problem solving genome. This "genome" is relatively stable, distinguishing every student from his or her peers and changes very little with the growth of the student knowlegde over the course. We also discovered that genomes are not randomly distributed either, instead students with similar genomes form cohorts that perform relatively similarly in the problem solving process. We believe that that our discovery of problem solving genome is a very important step towards our goal of predicting and preventing unproductive behavior. Indeed, the stability of patterns on the personal level makes the task of pattern prediction feasible while the presence of cohorts opens the way to detect student problem-solving genome early in the learning process. In this paper we present our approach of detecting student problem-solving genome and report our exploration of the genome on the level of individual students and cohorts.

2. Related Work

2.1 Parametererized questions and Exercises

Recent studies in educational technology have demonstrated promising results by leveraging computer and Web abilities to deliver parameterized exercises worldwide, which has become one of the focusing topics in Web-enhanced education. One of the most influential system, CAPA [11], was evaluated in a number of careful studies [10, 11], providing clear evidence that individualized exercises can significantly reduce cheating while improving student understanding and exam performance. The CAPA technology has been later integrated into popular LON-CAPA platform [14] and its functionality defined the assessment architecture of eDX.

Due to the complexity of parameterized assessment, the majority of work on parameterized questions and exercises was done in physics and other math-related domains where a correct answer to a parameterized question can be calculated by a formula that includes one or more question parameters. There are, however, examples of using this technology in other domains. In particular, our team focused on parameterized exercises for teaching programming. We developed and explored QuizPACK platform for C-programming [3] and a similar QuizJET platform for Java programming [9]. This paper is based on our experience with both platforms and uses data obtained in 3 semesters classes using another system Progressor+ [7] that uses the set of parameterized exercises of QuizJET.

2.2 Sequential Pattern Mining

Mine patterns on students sequential actions has recently gain attention in educational data mining field. Using activity data collected from groups of students using interactive tabletops, Martinez et al [15], mined and clustered frequent patterns to analyze distinct behaviors between low and high achievement groups. The differential sequence mining method, introduced by Kinnebrew and Biswas [13] has been successfully used to differentiate behavioral patterns among groups of students (like low and high performance students.) The method uses SPAM [1] to find common patterns in the sequences of the whole dataset, and then statistical tests are applied to see differences on the frequencies of the discovered patterns among different groups. The same authors have applied this technique in data col- lected from the system Betty's Brain to discovered patterns that can distinguish self-regulated behaviors in successful and non-successful students [2], and to analyze the evolution of reading behaviors in high and low performance students during productive and non-productive phases of work [12]. Herold, Zundel and Stahovich [5] have used the differential sequence mining on sequences of actions on handwritten tasks and proposed a model to predict performance on the course based on pattern features. Our work extends this prior work by utilizing and aggregating the mined sequence patterns to construct student activity profiles. Such profiles enable us to evaluate the statistical differences at the student, exercise, and group levels.

2.3 Clustering in EDM

Clustering techniques has been widely used in educational data mining, especially for the purpose of grouping students [18]. Spectral clustering is a graph based clustering technique widely used in machine learning [20]. Trivedi et al [22] applied spectral clustering in educational data and demonstrated how the technique performs better, compared with other popular clustering methods like k-means, in understand global similarities among data points in complex education datasets.

2.4 Problem Solving Repetition

Problem solving repetition behaviors has been studied by psychologists in different ways, providing evidence that repetition behaviors have roots in cognitive, metacognitive and motivational aspects and explaining why some students quit and some persist when facing challenging problems [16]. Schunk [19] shows the positive correlation between persistency in repeating and self-efficacy (believe on self capabilities/skills to solve a problem). The attribution theory [24] describes how students that attribute performance outcomes (successes, failures) to effort tend to work harder than students who attribute to ability. This explanation is in the same line with the Growth and Fixed Mindset theory [4], which demonstrate how growth mindset people (believing that intelligence is malleable and that new things can be learned with effort) hardly get frustrated and tend to keep trying, compared with fixed mindset people (the intelligence is fixed and can not be change, no matter how much effort is put.) Grounded in the literature in educational psychology, we conjecture that patterns on problem solving repetition may be explained by individual learners' motivational traits that are part of learners' personality [17]. These theories provide insights into analyzing to which extent these behaviors are stable on students.

3. System and Dataset

We collected answers of students to a set of parameterized exercises of our system Progressor+ [7] over three semesters of a Java programming course in the School of Information Sciences in the University of Pittsburgh (Spring 2012, Fall 2012 and Spring 2013.) A parameterized exercise is an exercise that is generated using a template. When the student requests an exercise, it is generated setting some parameters which determine the correct answer. When the user answers, the system shows if it was correct or wrong, shows the correct response, and lets the student "try again". The next time, the exercise will be generated with other values and the correct answer will be different. In this way, the student can try the same exercise many times, leaving a trace of successes and failures. We describe such sequences of repetitions as sequences of 1s (correct answer) and 0s (incorrect answer) of one student within one exercise. We have observed many sequences of 1, 2 or 3 attempts like 0, 1, 01, 11, 011, etc, and also many other with several attempts which are more difficult to explain, like 001011011 or 11000. In the system, there are 103 different parameterized exercises organized in 21 topics (Variables, Objects, Arrays, etc.) Exercises are also labeled n terms of complexity as easy, medium and hard. There are 41 easy exercises, 41 medium exercises and 19 hard exercises. Overall, the students attempted 6489 and 14726 times giving incorrect and correct responses, respectively. Easy exercises were attempted 10620 times, medium complexity exercises were attempted 7876, and hard exercises were attempted 2719 times. The users usually go exercise by exercise repeating 1 or more times. There are 4212 sequences of only 1 attempt (no repetition) and 4758 sequences with more than 1 attempt. The frequency decreases following the power law shape: there are 2717 with more than 2 attempts, 1583 with more than 3 attempts, and 1016 sequences with more than 4 attempts.

4. Method

An overview of our analysis method is shown in Figure 1. The steps can be summarized as follows: First, we label students' attempts using time and correctness (Figure 1(a), Section 4.1). We then apply sequential pattern mining to extract sequential attempt patterns (Figure 1(b), Section 4.2) and further construct vector representation for each students based on the mined patterns' frequencies, which we called *Problem Solving Genome* (Figure 1(c), Section 4.3). Using the predefined performance groups (Figure 1(d), Section 4.4) we analyze distances between pairs of students' genome to determine how stable the genome is and to which extent the patterns depends on exercise complexity and performance (Figure 1(e), Section 4.5). Finally, we characterize different problem solving patterns by clustering the problem solving genomes (Figure 1(f), Section 4.6). We detail each step in the following subsections.

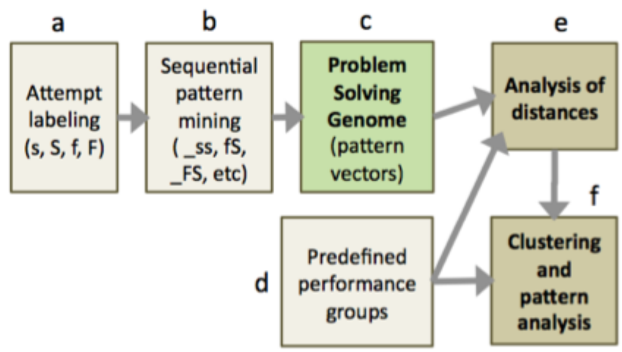

Figure 1. Method steps followed in this work.

4.1 Attempts labeling

We use both time and correctness of each attempt to label it for further use in sequential pattern mining analysis. In this way, each action will convey more information than using only correctness. As shown in the Figure 2, distribution of times for first attempts are different from other (non-first) attempts. This is reasonable if we consider that the user needs extra time the first time to read and understand the exercise. Additionally, time distribution is different for different exercises, as in general, complex exercises need longer times. Thus, for labeling the time factor, we used time information of historical records in our system to compute median times for each exercise for both first and other attempts. Then, we labeled the attempt as short or long depending on the time being lower or greater than the corresponding median (the median of the distribution for the specific exercise.) Combining correctness and time, we finally label the attempts using the letters 's' (lowercase s) for a short success, 'S' (uppercase S) for a long Success, 'f' for a short failure, 'F' for a long Failure.

The labeled attempts are organized in sequences by pairs student-question within a session in the system. Each sequence $s_{u,e}$ represent the sequential attempts of user u in the exercise e within a session. If the user attempted the same exercise in different sessions, there will be more than one sequence $s_{u,e}$.

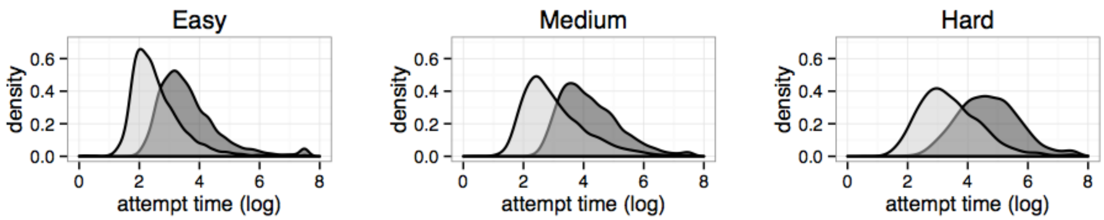

Figure 2: time distributions (log) for easy, medium and hard exercises. The right curve is always the first attempt time distribution, showing that first attempts usually take longer times.

4.2 Sequential pattern mining

To discover frequent patterns, we use PexSPAM algorithm [6], which extends the fast SPAM algorithm [1] with gap and regular expression constraints. Given a sequence database D = s_1, s_2, ..., s_n, the support of a pattern α is the number of sequences of D which contains α as a subsequence at least once. If the support of α is bigger than a threshold, then α is considered a frequent pattern. Support measure does not inform for multiple occurrences of the pattern within a sequence. In this work, we set a small minimum support in 1% because the sequences in our dataset tend to be short but many, i.e. a pattern that occurs in few sequences can still make a difference when looking at the aggregation of pattern occurrences by student. Additionally, and since we are interested in looking at patterns of 2 or more sequential attempts, we set the gap in 0 and considered only sequences with more than 1 attempt. After running the mining algo- rithm, we discover 102 common patterns occurring at least in 1% of the sequences. The top 20 patterns and the corresponding support can be seen in Table 1. Finally, we use '_' (underscore) to mark starting and/or ending patterns. For example _fS means start with a short failure and then make a long success.

Table 1: Top 20 frequent patterns with their sup- port.

	Pattern	Support		Pattern	Support
1	ss_	0.163	11	_FS	0.07
2	ss	0.107	12	FS	0.066
3	Ss	0.101	13	FS_	0.060
4	SS_	0.091	14	FF	0.059
5	_FS_	0.086	15	SS	0.058
6	_FF	0.083	16	_SS	0.054
7	Ss_	0.081	17	_ss_	0.053
8	_fS_	0.079	18	_SS_	0.052
9	_fF	0.077	19	sss	0.050
10	sss_	0.074	20	_fS	0.048

4.3 The problem solving genome: characterizing students with pattern vectors

Using the 102 patterns discovered by the sequential pattern mining, we compute frequency vectors of the patterns by student and by exercise. For further analysis of the relationship of the patterns among students, the performance levels and different complexity exercises, we split the data to compute different frequency vectors per student, as listed below.

overall pattern frequencies: counts the patterns' occurrence within all sequences of the student and divides by the total number of sequences. Since more than one pattern can occur in a sequence, and a pattern can occur more than one time in the same sequence, the frequencies in the vector may not sum 1.

early pattern frequencies: counts the patterns’ occurrences within the first half of the sequences of the student, and divides by the half number of sequences of the student.

late pattern frequencies: counts the patterns’ occurrences within the second half of the sequences of the student, and divides by the half number of sequences of the student.

random half frequencies: samples randomly half of the student’s sequences and computes the frequencies of the patterns within them. We compute two of such vectors per student, covering all student’s sequences.

random half frequencies within easy exercises: samples randomly half of the student’s sequences within easy exercises and computes the frequencies of the patterns within them. We compute two of such vectors per student, covering all student’s easy exercise sequences.

Additionally, for the sake of understanding pattern differences among complexity levels (i.e. easy, medium, hard exercises), we compute exercise pattern frequencies vectors grouping sequences by exercise and counting pattern occurrences.

4.4 Predefined performance groups and data filtering

Students were classified in predefined performance groups (PPG) based on scores on pre and posttest we collected in the 3 semesters. The pretest and posttest were highly similar among different semesters (small variation on questions) and the scores were further normalized as (score) / (max score) (having that min score is 0.) Additionally, using the normalized pre and posttest scores, we compute a normalized learning gain score as (normalized post score) - (normalized pre score.) For each of the pretest, posttest, and learning gain measures, students were classified in three groups using the percentiles 33.3 and 66.7: low, medium and high. For example, a student with pretest lower or equal than the percentile 33.3 in the pretest score distribution was classified as low pretest student. Summarizing, we have 3 PPG (low, medium, high) for each performance measure (pretest, posttest and learnig gain.) We collected a total of 97 pretest and 93 posttest results in the 3 semesters. We filter out students with few usage of the system setting a threshold of minimum 20 sequences and minimum 2 sessions. Additionally, we exclude one student that present a few sequences in the early stage with a very unusual repetition of short successes (67 ’ss’ patterns in 16 sequences). We consider this student as a clear outlier. We end up with 67 students in total having pretest, 65 of them having both pre and posttest. Table 2 shows the number of students in each PPG.

Table 2: Number of students in each predefined performance group (PPG).

	Pretest (total=67)	Posttest (total=65)	Learning gain (total=65)
low	24	22	22
medium	16	19	20
high	27	24	23

4.5 Analysis of distances

To see the value of the patterns as potential descriptors of behaviors, we seek to test to which extent students are stable on their patterns and to which extent the patterns depend on the predefined performance classification (knowledge), and the complexity levels of the exercises. We ask: Do pattern depend on individual differences of the students, student’s knowledge, or exercise complexity? To answer this question, we perform 3 different set of analysis based on distances between pattern vectors (described before). The first analysis Patterns among predefined groups aims to see how performance groups can be explained by the patterns students are using for solving parameterized exercises. Does students with similar performances have similar patterns for solving parameterized exercises? Is this similarity, between the students of the same predefined performance group, more than the similarity we can find between the students from different groups? For this analysis we contrast the overall pattern frequencies vectors on students classified in the performance groups. The second analysis Patterns among questions by complexity groups seeks for understanding the impact of exercise complexity on the patterns. For this, exercises are classified as easy, medium or hard and the distances between pattern vectors of the exercises are compared within and between groups. Finally, the third analysis Patterns as stable behavior of students looks for comparing students with themselves and to other students and answers the question: are students stable on their patterns? Here we use random half frequencies vectors, early and late pattern frequencies vectors, and random half frequencies within easy exercises vectors.

In each of the analyses, the distances are computed using Jensen-Shannon (JS) divergence between the pattern frequency distributions of a pair of pattern vectors. We use JS divergence as it is a symmetric version of Kullback-Leibler divergence and has been widely used for computing distance between frequency distributions. Since a pattern might occur more than once in a sequence, and more than one pattern may occur in a sequence, the frequency vectors are not summing to 1. Thus, we normalize the vectors before computing distances.

4.6 Clustering students based on the genome and pattern by pattern analysis

We aim to see individual patterns differences and their relationship with different behaviors and learning outcomes. For this, we cluster students using the overall pattern frequencies and characterize the clusters in terms of the distinguishable patterns. We use spectral clustering technique [23] as it gives a better separation of the students. We finally contrast pattern frequencies between performance groups in each cluster.

5. Results

5.1 Analysis of distances

5.1.1 Patterns among predefined groups

In this analysis, the goal is to find out if the students in the same predefined performance group (PPG) behave more similar to each other than to the students from other group. As we described before, the PPG are low, medium and high in 3 measures: pretest, posttest and learning gain. We sample 50% of all possible pairs of students within and between PPGs and compute the distances of all within and between group pairs. Then, we compare the average of distances within and between groups to see if students inside each group are more similar to each other than to students in other groups. Normality and homogeneity of variance is not met for all groups, thus we used Krustal-Wallis non-parametric mean rank test and Mann-Whitney test for single comparisons. We constrained the analysis to PPGs low and high to see extreme differences, and we use the overall pattern frequency vectors.

Table 3: Statistical tests on differences on distances between pairs of students within low, within high, and between low and high groups PPGs.

	low		high		low-high		Krustal-Wallis test			Mann-Whitney test		
	M	SE	M	SE	M	SE	Mean Ranks (low,high,low-high)	χ^2	*sig.*	Mean ranks	z	sig.
Pretest	.465	.014	.547	.017	.512	.010	294.68, 368.67, 341.51	11.926	.003	222.70, 258.21 low < low-high	-2.537	.011
Posttest	.486	.016	.516	.018	.511	.011	256.41, 271.97, 273.69	1.061	.588	-	-	-
L. Gain	.507	.019	.470	.018	.517	.013	242.32, 216.57, 251.35	5.276	.071	-	-	-

Results are shown in Table 3. Mann-Whitney comparison is reported only where significant differences among groups were found (pretest). For pretest groups, distances within the low group (mean rank = 222.70) are significantly smaller than distances between low and high groups (mean rank = 258.21), $z = -2.537$, $p = .011$. This suggests that student with no previous experience tend to behave differently than students with stronger background. There is no significant difference between high and low-high distances, though, meaning that high group behave more heterogeneously than low group. For posttest and learning gain groups there are no significant differences on distances within and between groups. Since we do not find clear differences on dis- tances within and between posttest and learning gain ex- treme groups, we hypothesize that pattern behaviors might not be due to performance factors, but because of other factors such as individual differences or exercises differences. We test for these hypotheses in the next analyses.

5.1.2 Patterns among questions by complexity groups

We now look for pattern differences among complexity levels of the exercises by computing distances between exercise frequencies vectors on pairs of questions within and between complexity groups. We filter out all questions with less than 20 sequences and perform comparisons between extremes groups, i.e. easy and hard complexity levels. Results of the Krustal-Wallis non-parametric test shows significant differences between distances within and between levels, $\chi^2 (2, N = 1596) = 160.359$, $p < 001$. Mean and standard error of distances within easy, within hard, and between easy and hard groups are shown in Table 4. Mann-Whitney test is performed to test differences among the levels. Distances within easy exercises (mean rank = 626.16) are significantly smaller than distances between easy and hard exercises (mean rank = 909.77), $z = -12.564$, $p < .001$. Similarly, the distances within hard exercises (mean rank = 277.20) are significantly smaller than distances between easy and hard exercises (mean rank = 383.13), $z = -4.733$, $p < .001$.

Table 4: Mean and standard error of distances within and between easy and hard exercises

	Mean	SE
within easy	.3311	.0031
within hard	.3478	.0085
between easy-hard	.4145	.0050

These results shows a clear dependency of the pattern behaviors with the complexity level of the questions. This is reasonable given that hard questions, that need more time, are expected to discourage repetitions. These results shows a clear dependency of the pattern behaviors with the complexity level of the questions. This is reasonable given that hard questions, that need more time, are expected to discourage repetitions. This suggests that for further analysis of students' individual differences, we need to control for the question complexity, as we will described in the next subsection.

5.1.3 Patterns as stable behavior of students

Using 50% random sample of the activity for each student, we compute distances of the student with herself, and between each student and all the others. We claim that if students are significantly closer (similar) to themselves than to others, this is a good evidence of the stability of the patterns. To perform a comparison between students we filter out all student with less than 60 sequences, limiting differences due to extreme differences on amount of activity. There were 32 students with at least 60 sequences. In this analysis we use paired samples t-test on the difference between the self and other distances. Normality assumption is met.

Results are shown in Table 5 first row (a). Students self- distances are clearly smaller ($M = .2370$, $SE = .0169$) than distances to other students ($M = .4815$, $SE = .0141$), $t = -15.224$, $p < .001$, Cohen's $d = 2.693$. To make the results stronger, we repeat this analysis with early and late pattern vectors to control for potential pattern change. We assume that if patterns depend on individual differences, they show stability even when students "evolve" in their behaviors. Results on Table 5 second row (b) confirm this idea: self distances ($M = .3211$, $SE = .0214$) are significantly smaller than between student distances (in early patterns) ($M = .4997$, $SE = .0164$), $t = -6815$, $p < .001$, Cohen's $d = 1.205$. This result is very promising, since it clearly shows that students are more similar to themselves in terms of pattern behaviors. Further more, we made a third version of this analysis considering random half frequencies within easy exercises vectors, to control for differences of students amount of activity on different complexity exercises, i.e. are differences among students explained because some students do more easy exercises and some students do more hard exercises? We performed this analysis with 39 students having at least 20 sequences in easy questions. Results showed in last row (c) in Table 5 confirm the stability of patterns: students are more similar to themselves (self distance $M = .3736$, $SE = .0214$) than to others (distances $M = .6065$, $SE = .0128$), $t = -10.352$, $p < .001$, Cohen's $d = 1.6569$. Since all effects are considerably high, these results strongly support the idea that pattern behaviors are due to personality rather than performance factors.

Table 5: Statistical tests comparing students with themselves and others

		self distances		dist. to others				
		M	SE	M	SE	t	*sig.*	Cohen's d
a)	random half frequencies	.2370	.0169	.4815	.0141	-15.224	$< .001$	2.693
b)	early/late pattern frequencies	.3211	.0214	.4997	.0164	-6.815	$< .001$	1.205
a)	random half frequencies in easy exercises	.3736	.0214	.6065	.0128	-10.352	$< .001$	1.657

5.2 Pattern by pattern differences on behavioral clusters

As we have seen in the Patterns as stable behavior of students analysis, student patterns are stable during time and among different performance groups. There are many other aspects that can result in pattern differences among students. One of these aspects can be the student's personality. To find out patterns differences of individual students and to see if specific type of students follows a special behavior in solving the problems, we use students' problem solving genome (pattern frequency vector), to cluster the students using spectral clustering technique with two clusters (K=2). We observe that two clusters give the largest eigen-gap, suggesting there are two intrinsic groups in the data. Figure 3 shows the top 30 frequent patterns in both of the clusters. Each point represents the average frequency of seeing a particular pattern in the cluster. Error bars are included to indicate significance. we order the patterns in x-axis by the differences between clusters 2 and 1. As we can see in this picture, some of the patterns (genes), such as _fS_ , _FS_ , _ss, Sss, etc., occur with significant frequency difference in the two clusters and some other patterns, such as _fS, fs_ , Ff, etc., do not show significant differences. If we look more closely, the sequences that start with failure are mostly related to the students in cluster 2 and the sequences that start with success are mostly related to the students in cluster 1. Also, we can see that the students in cluster 1 tend to repeat their successful attempts more and more frequently (e.g. the ssss pattern). In other words, even when they get the right answer to the question, they will insist on confirming knowing the question by repeating it again and again. Unlike students in cluster 1, the students in cluster 2 mostly stop after figuring out the right answer to the question, even if they have struggled for the correct answer in their previous attempts (e.g. _fS_ , _ss_ , and FS_ patterns.) Looking at this behavior, we can see that the students' genome can lead to different behaviors in solving parameterized exercises. Based on these findings, we call the first cluster of the students the confirmers and the second cluster the non-confirmers.

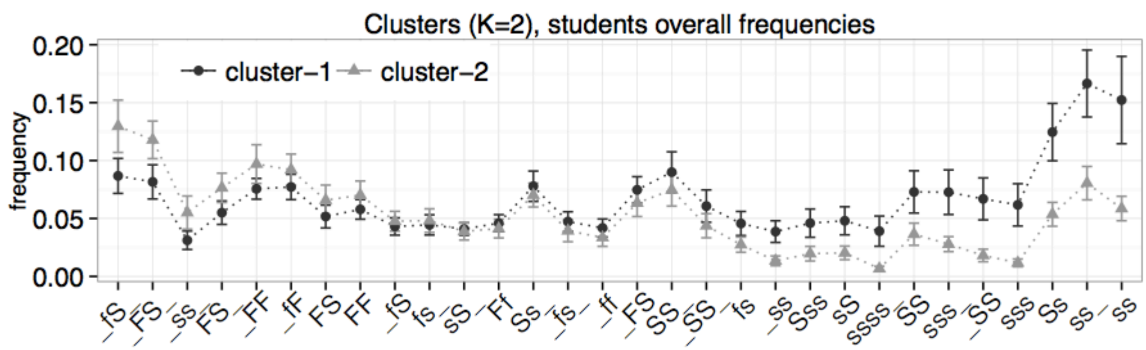

Figure 3: Top 30 patterns and their frequencies in each cluster. Patterns are ordered by the difference on frequencies between cluster 2 (non-confirmers) and 1 (confirmers).

Furthermore, to find out the clusters' relationship with different behaviors and learning outcomes, we compare the pattern by pattern differences for different PPGs within each cluster and describe the patterns that distinguish them. As we have mentioned earlier, both of the clusters have students from all PPGs. As a result, we cannot say that the student's genome has a direct impact on the performance of the student. Both confirmers and non-confirmers can have high or low performance. To look at the clusters deeply and to see if there are any differences in the patterns, within each cluster, that can drive students' performance, we repeat the first analysis within each cluster looking at the learning gain. For each of the clusters, we look at the patterns (genes) and the difference between their average frequencies for the students with low and high learning gain. The result is shown in Figure 4. The upper diagram shows the students in cluster 1 (the confirmers) and the lower diagram shows the students in cluster 2 (the non-confirmers). The red line with round markers show the pattern frequencies for low learning gain students and the blue line with the triangle marker is representative of high learning gain students. If we look at the patterns in cluster 1 (the confirmers), we can see that there are some patterns that show significant difference between the low and high learning gain students. These patterns all start with a failure: _FS_ and Ff have long failures in the beginning of the patterns and _fF, fs_ , and _ff, have short failures at the beginning of the patterns. Among these patterns, only _FS_ is practiced more by the high learning gain students. This indicates that, among the confirmer students, the ones that put a good amount of effort to answer a question right, after a long failure and stop repeating the same question learn more. The low learning gain group shows more frequent use of the Ff, _fF, fs_ , and _ff patterns. The common element of all of these patterns is short failure (f). If we look at Figure 4 for confirmers, we can see that all of the patterns that include a short failure, are practiced more by the low gain students. This can indicate that the low gain confirmer students do not spend enough time and thought on the questions that they do not know the answer of.

The non-confirmer students show more pattern differences between the low and high learning gainers. We can see that the high learning gain group follow the patterns of _FF, FS, _FS, SS_ , _SS, SS, and Ss more frequently. This means that the high learning gain, non-confirmer students tend to continue trying a non-parameterized exercise and spending time on it after they failed in it or it took them a long time to get to the correct answer for that exercise. In this sense, these students are closer to the confirmer group of students (cluster 1) but only at the times that they are not sure if they have learnt the solution to an exercise. On the other hand, the low learning gain group tend to develop the fs_ , _fs_ , and _ff patterns in their sequence. The first two indicates that they give up practicing the exercise after having a short success that comes after a short failure. Also, they tend to repeat short failures on the same exercise more often.

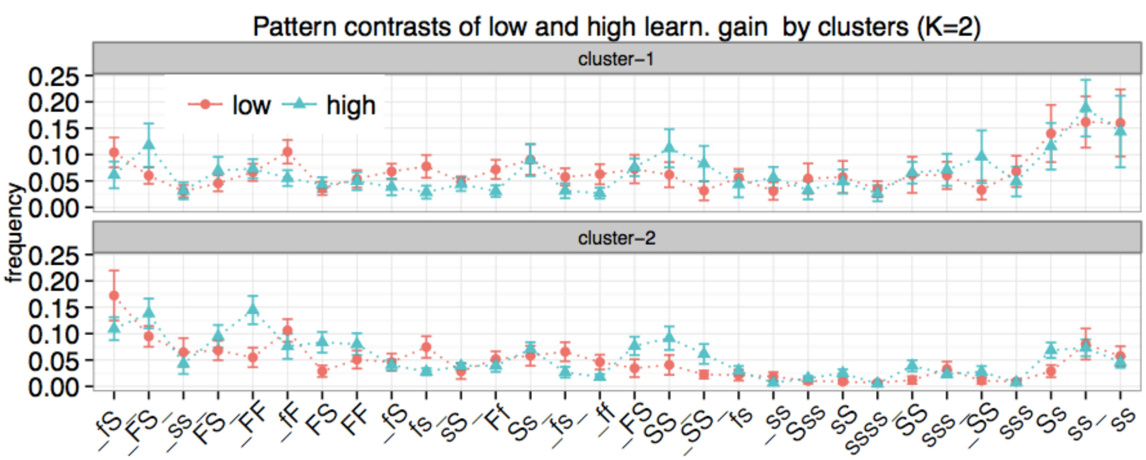

Figure 4: Top 30 patterns and their frequencies for low and high learnig gain PPG by cluster.

Another interesting observation here is that having repeated successes in the same parametrized exercise does not add to the learning gain of the students. We can see that none of the patterns having more than one short success make any significant differences between the low learning gain and high learning gain students.

The above analysis shows the specific patterns that can explain the differences between high and low learning gain students in each of the confirmers and non-confirmer clusters. In both of the clusters, short failures are more associated with low learning gain students. For the non-confirmers group, the students, who acted similar to the confirmers group in cases of having a hard time getting to the right answer, had higher learning gain. Also, repeating the short success did not add to the learning gain of students. These results are promising for the further guidance of the students in the correct use of the system and increasing their performance. Based on a student's pattern cluster, we can guide him/her to follow the sequences with the patterns associated with high learning gain (such as encouraging them to think longer on questions) and not following the patterns that do not affect their learning gain or having a negative effect (e.g. stopping the student from repeating short successes).

6. Acknowledgements

This Independent Study Report presents the work done in collaboration with Saghayegh Sahebi and Peter Brusilovsky. Full version of this report is published at EDM'2014.

7. References

[1] Ayres J, Flannick J, Gehrke J, Yiu T (2002) Sequential pattern mining using a bitmap representation. KDD 2002, 429-435

[2] Bouchet, F., Kinnebrew, J. S., Biswas, G., and Azevedo, R. (2012). Identifying Students' Characteristic Learning Behaviors in an Intelligent Tutoring System Fostering Self-Regulated Learning. In EDM (pp. 65-72).

[3] Brusilovsky, P. and Sosnovsky, S. (2005). Individualized exercises for self-assessment of programming knowledge: An evaluation of quizpack. ACM Journal on Educational Resources in Computing, 5(3):Article No. 6, 2005.

[4] Dweck, C. (2006). Mindset: The new psychology of success. Random House Digital, Inc

[5] Herold, J., Zundel, A., and Stahovich, T. F. (2013). Mining Meaningful Patterns from Students' Handwritten Coursework. In EDM 2013 (pp 67-73).

[6] Ho, Joshua, Lior Lukov, and Sanjay Chawla. Sequential pattern mining with constraints on large protein databases. In Proceedings of the 12th International Conference on Management of Data (COMAD 2005b), pp. 89-100. 2005.

[7] Hsiao, I-H. and Brusilovsky, P. (2012) Motivational Social Visualizations for Personalized E-learning, In: Proceedings of 7th European Conference on Technology Enhanced Education (ECTEL), ECTEL 2012, Saarbrücken, Germany, September 18-21, 2012, Springer-Verlag, Volume 7563/2012, pp.153-165.

[8] Hsiao, I. H., Sosnovsky, S., and Brusilovsky, P. (2010). Guiding students to the right

questions: adaptive navigation support in an E-Learning system for Java programming. Journal of Computer Assisted Learning, 26(4), 270-283.

[9] Hsiao, I. H., Sosnovsky, S. , and Brusilovsky, P. (2009). Adaptive navigation support for parameterized questions in object-oriented programming. In U. Cress, V. Dimitrova, and M. Specht, editors, 4th European Conference on Technology Enhanced Learning (ECTEL 2009), volume 5794 of Lecture Notes in Computer Science, pages 88-98. Springer-Verlag.

[10] Kashy, D. A., Albertelli, G., Ashkenazi, G., Kashy, E., Ng, H.-K., and Thoennessen, M. (2001). Individualized interactive exercises: A promising role for network technology. In 31st ASEE/IEEE Frontiers in Education Conference. IEEE, 2001.

[11] Kashy, E., Thoennessen, M., Tsai, Y., Davis, N. E., and Wolfe,S. L. (1997). Using networked tools to enhanse student success rates in large classes. In 27th ASEE/IEEE Frontiers in Education Conference, volume I, pages 233-237. Stipes Publishing L.L.C., 1997.

[12] Kinnebrew, J. S., and Biswas, G. (2012). Identifying Learning Behaviors by Contextualizing Differential Sequence Mining with Action Features and Performance Evolution. In EDM (pp. 57-64).

[13] Kinnebrew, J. S., Loretz, K. M., and Biswas, G. A. U. T. A. M. (2012). A contextualized, differential sequence mining method to derive students' learning behavior patterns. Journal of Educational Data Mining.

[14] Kortemeyer, G., Kashy, E. , Benenson, W., and Bauer, W. (2008). Experiences using the open-source learning content management and assessment system lon-capa in introductory physics courses. American Journal of Physics, 76(438), 2008.

[15] Martinez, R., Yacef, K., Kay, J., Al-Qaraghuli, A., and Kharrufa, A. (2011). Analysing frequent sequential patterns of collaborative learning activity around an interactive tabletop. In Proceedings of the 4th international conference on educational data mining (Vol. 13, No. 2, pp. 111-120).

[16] Mayer, R. E. (1998). Cognitive, metacognitive, and motivational aspects of problem solving. Instructional science, 26(1-2), 49-63.

[17] McAdams, D. P. (1995). What Do We Know When We Know a Person? Journal of Personality, 63(3), 365-396.

[18] Romero, C., and Ventura, S. (2010). Educational data mining: a review of the state of the art. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 40(6), 601-618.

[19] Schunk, D. (1991). Self-efficacy and academic motivation. Educational Psychologist 26: 207¸231.

[20] Shi, J., and Malik, J. (1997). Normalized cuts and image segmentation. In CVPR '97.

[21] Sosnovsky, S., Brusilovsky, P., Lee, D. H., Zadorozhny, V., and Zhou, X. (2008) Re-assessing the Value of Adaptive Navigation Support in E-Learning. In: W. Nejdl, J. Kay, P. Pu and E. Herder (eds.) Proceedings of 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2008), Hannover, Germany, July 29-August 1, 2008, Springer Verlag, pp. 193-203

[22] Trivedi, S., Pardos, Z. A., Sárközy , G. N., and Heffernan, N. T. (2011). Spectral Clustering in Educational Data Mining. In EDM (pp. 129-138).

[23] Von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17(4), 395-416.

[24] Weiner, B. (1986). An Attributional Theory of Motivation and Emotion. New York: Springer-Verlag.