Using Sequential Pattern Mining to Explore Learners' Behaviors and Evaluate Their Correlation with Performance in Inquiry-Based Learning

Conference Paper · September 2017

DOI: 10.1007/978-3-319-66610-5_21

5 authors, including:

Remi Venant
Institut de Recherche en Informatique de Toulouse

Kshitij Sharma
Norwegian University of Science and Technology

Philippe Vidal
Institut de Recherche en Informatique de Toulouse

Pierre Dillenbourg
École Polytechnique Fédérale de Lausanne

Using sequential pattern mining to explore learners' behaviors and evaluate their correlation with performance in inquiry-based learning

Rémi Venant[1], Kshitij Sharma[2], Philippe Vidal[1], and Pierre Dillenbourg[2] and Julien Broisin[1]

[1] Université de Toulouse, IRIT, 31400 Toulouse, France,
{remi.venant,philippe.vidal,julien.broisin}@irit.fr
[2] Computer-Human Interaction in Learning and Instruction (CHILI), Ecole Polytechnique Fédérale de Lausanne (EPFL), station 20, CH-1015 Lausanne, Switzerland
{kshitij.sharma,pierre.dillenbour}@epfl.ch

Abstract. This study analyzes students' behavior in a remote laboratory environment in order to identify new factors of prediction of academic success. It investigates relations between learners' activities during practical sessions, and their performance at the final assessment test. Based on learning analytics applied on data collected from an experimentation conducted with our remote lab dedicated to computer education, we discover recurrent sequential patterns of actions that lead us to the definition of learning strategies as indicators of higher level of abstraction. Results show that some of the strategies are correlated to the learners' performance. For instance, the construction of a complex action step by step, or the reflection before submitting an action, are two strategies applied more often by learners of a higher level of performance than by other students. While our proposals are domain-independent and can thus apply in other learning contexts, the results of this study led us to instrument for both students and instructors new visualization and guiding tools in our remote lab environment.

1 Introduction

Research on predictors of success in learning has been a hot topic for decades [1–4]. Many studies in that field focused on finding predictors of performance, which is commonly measured through academical assessment. Predictors are traditionally based on information about learners collected through past academic results, pre-course tests or questionnaires that include, among others, work style preference, self-efficacy [5], background or expectations [6]. However, the development of Technology Enhanced Learning (TEL), combined with the emergence of Educational Data Mining (EDM) and Learning Analytics (LA), provide new capabilities to explore learners' behavior in learning situations and to study its influence on their performance.

Remote or virtual laboratories (VRL) are learning environments designed to support inquiry learning through practical activities with the mediation of computers. Within these environments, learners develop inquiry and self-regulated skills through interactions with remote or simulated apparatus, but also collaborative skills through interactions with peers and instructors. With the tracking of these interactions, VRL may provide an insight of learners' behaviors at a high resolution that could lead to a better understanding of the learning process. While studying these actions through independent measures can be a first approach, the analysis of sequential patterns may provide another understanding of how learners act [7]. Sequential pattern mining, as a method to identify relevant patterns of actions within a set of sequences [8] is then to be considered.

In order to explore the potential links between learners' behavior and their performance, we conducted an experiment in a real class environment, with 85 students enrolled in a Computer Science program. We explore in this article the interactions between learners and the remote apparatus to study the potential correlations between the learners' performance score at the final assessment test, and both quantitative indicators and sequential action patterns. Our objective is to identify behavioural patterns for a practical session that lead to better learning outcomes, in order to predict learners' performance and to automatically guide students who might need more support to complete their tasks.

The next section presents the computational settings (i.e., our learning environment, with a focus on its tracking framework), and exposes the experimentation protocol together with the resulting dataset. While a first analysis exposed in section 3 covers engagement indicators such as the number of actions achieved by a student, or the time between two actions, section 4 proposes a methodology based on sequential pattern mining to discover sequences of actions that are representative of the learners' level of performance. These patterns allow for specification of abstract indicators, viewed as learning strategies and correlated with students' success. We then situate our research work among existing studies in the field of computer education and dedicated laboratories, and discuss about the impact of our study on new artificial intelligence features integrated into our remote lab environment.

2 Experimental Settings

The experimentation was conducted at the Computer Science Institute of Technology (CSIT) of the University of Toulouse (France). For the whole experimentation, learners used our web-based virtual laboratory environment dedicated to computer education, and especially to system and network administration, to complete the whole set of practical tasks they were asked to.

2.1 The Learning Environment: our Virtual and Remote Laboratory

Our learning environment is a web-based platform that relies on a cloud manager to offer on-demand remote laboratories made of virtual computers and networks, and that features advanced learning capabilities [9]. It has been designed to overcome the spatial limitations and restrictions of access to physical resources: it provides, for example, each learner with a set of virtual machines, routers and switches accessible from anywhere and without any limitation of use (i.e., students are granted with the administrator role).

Within this environment, instructors can create a practical activity by designing the topology of machines and networks needed by learners to achieve the pedagogical objectives; the activities achieved within that environment are up to the teacher, as the environment does not enforce any form of learning scenario. When a learner accesses a particular activity, the system automatically creates and sets the different virtual resources up. Learners can then manipulate the machines (i.e., start them up, put them to sleep, etc.) and interact with them through a web-based terminal similar to a traditional terminal.

At the time of the experimentation, the learning features accessible to learners (and instructors as well) included real-time communication (i.e., an instant messaging system), collaborative work (i.e., several learners can work together on the same machine and see what others are doing), awareness tools (i.e., learners can compare actions they are carrying out against the actions being carried out by their peers), as well as tools for replay and deep analysis of working sessions. Let us note that the system makes it possible for teachers to deactivate a given learning feature for a particular practical activity.

In addition to the above pedagogical facilities, our virtual lab environment integrates a learning analytics framework able to collect in the xAPI format [10] most of users interactions with the system. In this study, we focus on interactions between learners and the remote virtual resources they had to administrate, as this kind of activity can be considered as almost fully representative of the learning tasks completed by learners.

Such interactions rely on the Shell commands executed within the web terminal. These commands include a name and, sometimes, one or more arguments (e.g., *ls -a -l* is the command name *ls* with the arguments *-a* and *-l*). Also, once a command is executed, the machine may return a textual answer (e.g., the execution of the command *ls -a -l* returns the list of all files and folders stored in the current directory). Thus, the xAPI statements at the basis of the pattern analysis suggested further in this paper consists of the 8 following elements: (i) the timestamp, (ii) the id of the laboratory, (iii) the learner's username, (iv) the id of the machine, (v) the name of the command, (vi) its arguments, (vii) the output the machine produced, and (viii) the technical rightness of the command. That last element is a boolean value inferred on the basis of the elements (v), (vi) and (vii) to indicate whether the command was executed successfully [11].

2.2 Experimentation Protocol and Learning Scenario

The experiment took place for an introductory course on Shell commands and programming; it involved 107 first year students, with a gender repartition that reflects the distribution of CSIT students. While the first objective here is to understand the concepts of the Shell itself (e.g.: standard output, redirection...), learner also deal with previous learned concepts on system architecture in depth through the manipulation of their relative commands. A third learning outcomes target programming skills applied to Shell. With prior competencies on basic algorithmic students acquired previously, they must understand how to automate administration of computing systems.

We conducted the experiment at the beginning of the course for three weeks, during which students had a 24-7 access to their own virtual machine deployed within our remote lab environment. Each week, a face-to-face practical session of 90 minutes was given. For that three weeks, the course targeted three main learning outcomes: understanding of a Shell command, file system management in Linux using some Shell commands, and understanding of several basic concepts of Shell programming. For each session, learners had to achieve a list of tasks involving a set of Shell commands. They first had to understand what the commands do, how they work (i.e., what arguments must/may be used), and then to execute them to achieve the given tasks. The last session required learners to reuse the commands they discovered during the first two sessions to build simple Shell scripts made of conditional statements or loops.

Finally, the pedagogical material provided to students only comprised, as PDF files, a textual description of the tasks to achieve and the name of the commands to use, along with few simple examples. For a full understanding of a certain command, learners had to consult the matching manual available in the Shell of their virtual machine.

2.3 The Resulting Dataset

Once outliers have been removed, the dataset comprises 85 students which submitted a total of 9183 commands. Then the mean number of commands by learner is 108.00 with a standard deviation $\sigma = 66.62$. The minimum of command submitted for a learner is 22 while the maximum is 288.

2.4 Measure of Academic Performance

We defined in this study the assessment score (AS) as a continuous variable between 0 and 20 that denotes the score learners got when they took the test at the end of the course. The distribution of AS in the experiment presents qualitative cutpoints that make clearly appear three categories of AS (AScat): low (named L; number of students (N) within this category = 22), medium (M, with N=27) and high (H, with N=36).

In the next two sections, the dataset resulting from the experimentation is analyzed against the AS and/or the categories of AS. The following section defines some quantitative indicators as independent variables and investigates their correlation with these two above mentioned dependent variables, before we go into deeper pattern mining analysis in Section 4.

3 Study of Quantitative Indicators

Starting from the records of the dataset, we first studied the four following quantitative indicators: (1) the number of commands submitted by a learner (*#submissions*); (2) the percent of commands executed successfully (*%success*); (3) the average time spent between two submissions of commands of the same working session ($\Delta Time$); and (4) the number of commands submitted by a learner that refer to help seeking (*#help*). The first three indicators can be found in other research works [4, 17] and allow quantifying learners' production. The last indicator identify help access. While it can be more complicated to compute it in other contexts, where help resources depends on other systems, or are gathered through the web, remote or virtual labs often come with their own assistance material, whose access can be easily tracked [12].

In order to identify working sessions, we applied a time series clustering algorithm and checked for each learner that their class schedule was consistent with the algorithm (i.e., the list of working session a learner did includes at least the sessions she/he had in class). The *#help* indicator is based on well-known patterns such as the command *man* that provides a complete manual of a certain command, or the arguments *–help* and *-h* that give a lightweight manual. Table 1 shows the Pearson correlation analysis between the four indicators defined above and the assessment score.

The indicators *#submissions* and $\Delta Time$ do not appear to be correlated with the assessment score, as the p-value for both indicators is greater than 0.05. Also, even if *%success* and *#help* present a weak significant correlation with AS, they only roughly reflect how students behaved during practical learning: *%success* is an indicator of production that does not take into account learners' progress, so as *#help* which does not reflect the way students sought for help (i.e., after a command failure, before testing a new command, etc.).

In order to go further in the analysis of learners' behavior, we explore in the next section how they carried out their activities in terms of sequences of commands; let us note that the word *instructions* may also be used in the remaining of the paper to designate such Shell commands.

Table 1. Pearson correlation between quantitative indicators and AS

	r	p-value
#submissions	0.193	0.076
%success	0.248	0.022
ΔTime	-0.127	0.247
#help	0.226	0.037

4 Pattern Mining Analysis

A pattern mining analysis was applied on the experimentation dataset to identify the significant sequences of actions carried out by learners during practical activities, and to analyze whether these sequences are related to the two dependent variables AS and AScat.

4.1 Nature of Actions

First, we propose to go further the restriction of the learning context by applying a pattern mining analysis not on commands themselves, but on their nature, their relationships, and the result of their execution. Hence, we define a generic *action* submitted by a learner on a resource in the context of a practical session as a structure of three components: its *type*, its *parameters* and its *nature*. The type and the parameters depend on the learning domain; for instance, *to supply a RLC electrical circuit* with a nominal tension of *12V* represent a type and a parameter of an action carried out for a practical work in Physics. In our context, the *type* is the *command name*, whereas the *parameters* represent its *arguments* (see end of Section 2.1). Regarding the *nature*, it provides semantic about the relation between an action and the action that has been submitted just before.

According to the above definition, we specified eight exclusive natures of actions: *Sub_S*, *Sub_F*, *ReSub_S*, *ReSub_F*, *VarSub_S*, *VarSub_F*, *Help* and *NewHelp*. The natures *Sub_** refer to an action whose type is different from the type of the previous action, and which has been executed successfuly (*Sub_S*) or not (*Sub_F*) by the resource. The natures *ReSub_** address an action that is identical to the previous one (i.e., same type and parameters), while the natures *VarSub_** represent an action of the same type than the previous one, but with different parameters. Finally, *Help* depicts an action of help seeking about the type of the previous action, while *NewHelp* indicates a help access without relations with the previous action. For instance, if the previous command is *ls -al*, the next command *rm* will belong to *Sub_F* (as *rm* has a different command name, and is technically wrong because that command requires at least one argument), *ls -al* to *ReSub_S*, *ls -alRU* to *VarSub_S*, while *man ls* will be classified with the nature *Help* and *man rm* with the nature *NewHelp*.

4.2 Patterns of Actions

To discover which sequences of actions were statistically significant, we analyzed two-length and three-length sequences only, as no sequences of length four or more were used by enough learners to be significant. The statistical tests applied for each sequence were a Pearson correlation test for AS, and an analysis of variance (i.e., one-way ANOVA) for AScat. The patterns appearing in Table 2 are those whose p-value is lower than 0.05 for at least one of the two tests. Also, the column "Trend of use" of Table 2 depicts the order of use of a pattern among the categories of AS, with its significance given in the column "ANOVA p-value". For instance, high-level students used the pattern #2 more than the low-level students, and medium level students also used this pattern more often than the low-level students; however, no ordered relation is given between high- and medium-level students for this pattern.

Table 2. Analysis of action patterns

#	Pattern	Test with AScat		Test with AS	
		Trend of use	ANOVA p-value	r	cor. p-value
1	Sub_S, VarSub_S	$H, M > L$	< 0.001	0.335	0.002
2	Help, ReSub_S	$H, M > L$	0.003	0.293	0.006
3	VarSub_S, NewHelp	$H, M > L$	0.007	0.210	0.053
4	VarSub_S, Sub_S	$H, M > L$	0.021	0.264	0.014
5	ReSub_S, NewHelp	$H, M > L$	0.026	0.361	< 0.001
6	VarSub_S, VarSub_S	$H, M > L$	0.031	0.203	0.062
7	Sub_S, VarSub_S, VarSub_S	$H, M > L$	0.002	0.286	0.008
8	VarSub_S, VarSub_S, Sub_S	$H, M > L$	0.003	0.294	0.006
9	Sub_S, VarSub_S, NewHelp	$H, M > L$	0.007	0.250	0.020
10	NewHelp, Sub_S, VarSub_S	$H, M > L$	0.009	0.243	0.025
11	Sub_S, ReSub_S, NewHelp	$H, M > L$	0.020	0.335	0.002
12	Sub_F, VarSub_F, VarSub_S	$L > H, M$	0.021	-0.217	0.046
13	Sub_S, NewHelp, ReSub_S	$H, M > L$	0.047	0.244	0.024

As shown in Table 2, 13 patterns appeared to be statistically significant. Most of them present both a significant trend of use between performance levels, and a significant weak (i.e., $0.1 < |r| < 0.3$) or medium (i.e., $0.3 < |r| < 0.5$) correlation with AS. It appears that most of these patterns are used by high- and medium-level students at a higher frequency than by low-level students, and positively correlated with the performance at the academic test; only one pattern of actions (i.e., pattern #12) is used more often by low-level students than by others, where students unsuccessfully submit a particular action by modifying its parameters until the submission succeeds. Nonetheless, no patterns make it possible to clearly distinguish high- and medium-level students.

Also, the patterns reveal common semantics depicting the students' behavior. For instance, the patterns 1, 6, 7, 8 and 9 show a sequence of a successful action (i.e., *Sub_S*, *ReSub_S* or *VarSub_S*) followed by another successful action characterized by the same type (i.e., *VarSub_S*). We make here the hypothesis that these patterns illustrate learners building a complex action progressively.

The set of patterns we identified can thus be viewed as approaches applied by learners to carry out a task or solve a problem. Some of them refer to a common methodology we define as *learning strategy*. In the next section, we identify these strategies from the patterns of Table 2, and analyze their relation with the academic performance.

4.3 Learning Strategies

The 13 patterns highlight eight strategies: *confirmation*, *progression*, *success-then-reflexion*, *reflexion-then-success*, *fail-then-reflexion*, *trial-and-error*, and *withdrawal*. *Confirmation* is the successful resubmission of the same action (i.e., command and arguments remain unchanged), while *progression* depicts a sequence of successfully executed actions of the same type, but whose parameters get more complex from one to another. *Success-then-reflexion* expresses a successful action, followed by access to the help related to the matching type. Conversely, *reflexion-then-success* appears when students first access the help of a certain type of action, and then submit the matching action successfully. *Fail-then-reflexion* shows an access to a help related to an action that failed. *Trial-and-error* expresses a sequence of trial of the same action with a variation of its parameters until the submission succeeds. Finally, *withdrawal* matches with an action of a different type than the previous one whose submission failed.

Table 3 shows the regular expressions we used to detect the above strategies within the learning paths followed by learners (i.e., within the sequences of natures of actions carried out by learners). For instance, the regular expression related to the *progression* strategy matches patterns of successfully executed actions of the same type but with different parameters, while help accesses to this type of action may appear between submissions.

Table 3. Regular expressions used for detection of learning strategies

<table>
<tr><th>Strategy</th><th>Regular expression</th></tr>
<tr><td>Confirmation</td><td>(?:Sub|ReSub|VarSub)_S,(?:Sub_S,)*(?:Sub_S)</td></tr>
<tr><td>Progression</td><td>(?:Sub|ReSub|VarSub)_S,(?:Help,)?VarSub_S</td></tr>
<tr><td>Success-then-reflexion</td><td>(?:Sub|ReSub|VarSub)_S,(?:Help|NewHelp)</td></tr>
<tr><td>Reflexion-then-success</td><td>(?:Help|NewHelp),(?:Sub|ReSub|VarSub)_S</td></tr>
<tr><td>Fail-then-reflexion</td><td>(?:Sub|ReSub|VarSub)_F,(?:Help|NewHelp)</td></tr>
<tr><td>Trial-and-error</td><td>(?:Sub|ReSub|VarSub)_F,
(?:(?:ReSub|VarSub)_F,)*(?:ReSub|VarSub)_F</td></tr>
<tr><td>Withdrawal</td><td>(?:Sub|ReSub|VarSub)_F,(?:Help,)*(?:NewHelp,Sub_)</td></tr>
</table>

4.4 Results

We studied the relationships between each of these strategies and the academic performance with the same tests than in section 4.2 (i.e., an ANOVA for AScat, and a Pearson correlation test for AS). Table 4 shows the results for that study. The significant values are highlighted in bold, while the strategies whose at least one result is significant appear in italic.

Progression, *success-then-reflexion*, *reflexion-then-success* and *fail-then-reflexion* are the strategies that present significant results. The first three ones allow to

Table 4. Analysis of learning strategies

Strategies	Test with AScat: Trend of use	Test with AScat: ANOVA p-value	Test with AS: r	Test with AS: cor. p-value
Confirmation	∅	0.745	0.108	0.321
Progression	$\boldsymbol{H, M > L}$	**0.001**	**0.294**	**0.006**
Success-then-reflexion	$\boldsymbol{H, M > L}$	**0.010**	**0.282**	**0.008**
Reflexion-then-success	$\boldsymbol{H, M > L}$	**0.015**	**0.242**	**0.026**
Fail-then-reflexion	∅	0.020	**0.273**	**0.011**
Trial-and-error	∅	0.341	-0.050	0.670
Withdrawal	∅	0.457	-0.004	0.968

cluster students in a category of performance and seem to be traits of behavior of students of high- and medium-levels of performance, while all of them present a significant positive weak correlation with the academic score.

Also, significant strategies are all positively correlated to the AS: the results do not reveal any particular behaviors of learners of low-level of performance. The trial-and-error strategy does not present any significant results in this experimentation. This may be explained by the experimental settings mentioned before (see Section 2): students were beginners in Computer Science, and the learning tasks they were assigned to relied on exploratory learning where learners had to discover by themselves the Shell commands. In this form of learning, doing multiple trials to discover and understand how the machine reacts is an expected behavior [13], no matter the performance level of the student.

Another interesting result is the *withdrawal* strategy which does not seem to be related with the assessment score. This strategy, applied homogeneously by all students, whatever their performance level is, does not express that students fail at achieving a particular task. Different hypothesis can explain the fact a learner suspends the realization of an action, such as the curiosity or the discovery of new actions. This strategy thus does not seem to be relevant to predict performance or to make a decision.

This analysis of learning strategies mainly reveals behaviors of high- and medium-level students that are positively correlated to the assessment score. With the *progression* strategy, high-level students seem to decompose their problem in steps of increasing complexity. The three others strategies used by high-level students are related to reflexion through the use of help; this result is in line with the findings of Section 3, where the indicator *#help* (i.e., the number of help accesses) is weakly and positively correlated with the academic performance.

5 Discussion

5.1 Results exploitation

The outcomes of this study gave us the opportunity to enrich our remote lab environment with new analytics providing insights of learners' behaviors to teachers and students as well. Figure 1 represents a set of visualizations illustrating the occurrences of both the success-then-reflexion (in green) and the reflexion-then-success (in purple) strategies followed by four different learners, for the whole duration of the experiment; each graph comes with the academic score and category of the matching student. The different visualizations strengthen the findings of the previous section: the more these strategies are used, the better score the student obtained at the assessment.

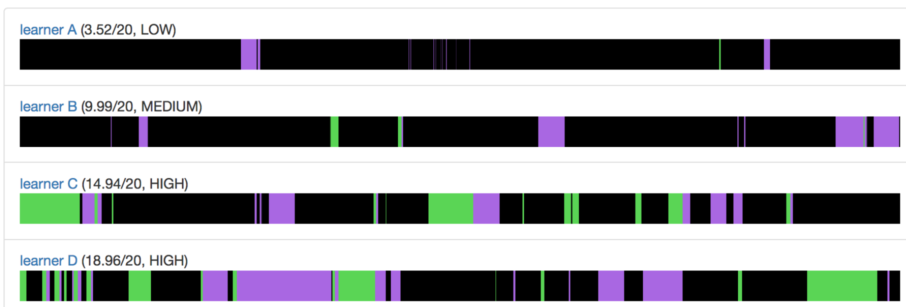

Fig. 1. Success-then-reflexion and reflexion-then-success strategies used by learners

While these visualizations are of interest to understand how learners act, the results of our analysis allow for on-the-fly detection of their behaviors and open the door for new opportunities. Indeed, the continuous improvement of TEL-based systems, according to experimental findings resulting from their usage, is a critical part of the re-engineering process [14]. Applied to learning analytics, this enhancement cycle makes it possible to discover new design patterns and to generate new data for research about and improvement of TEL [15].

Thus, with respect to this methodology, we integrated into our remote lab environment two new features built on two distinct design patterns. The first feature relies on an intelligent tutoring system (ITS) able to guide learners during their practical sessions according to the learning strategies they are currently engaged in. For instance, when a learner fails several times to execute a command, the ITS suggests the learner to read the matching manual or to seek help from a peer that has successfully used that command, so that the learner becomes engaged in the reflexion-then-success strategy leading to better performance. The second design pattern we implemented is an awareness system intended for teachers and highlighting, based on the learning strategies followed by learners, students that seem to present weaknesses. For instance, if several learners follow the withdrawal strategy on the same command, the system notifies the teachers so they can make a collective intervention. These new features are already implanted into our system and will be evaluated in the near future through different axis: their usability, their reliability to guide learners and notify teachers, and the impact they may have on both learners' and teachers' behaviors.

5.2 Related work

In computer education, several studies have been conducted to find out what characteristics of learners' profile may predict their success or failure in a given learning activity; such characteristics include pre-activity properties like personality traits and past academic achievement [5, 16], or demographic factors and learners' expectations [6]. To take into account such indicators is useful, for example, to identify learners that may require more attention and for which a personalized tutoring would be beneficial. However, this approach restrict learners' data to information that cannot evolve during the activity: the learning activity is seen as an object that does not impact learning outcomes. Instead, the approach we adopted, based on learning analytics about learners' interactions occurring all along the practical activity, tends to overcome this issue since it considers learners' interactions as a potential variable of performance prediction.

In Computer Science, other research works also adopt a learning analytics approach to predict performance. For instance, Blikstein [3] and Watson & al. [17] rely on the source codes produced by learners to analyze various indicators such as the code size, the number of compilations, the time between two compilations, or the score students got at the post-experimental test. In another way, Vihavainen [4] presents a quantitative study in an introductory programming course where snapshots of students' code are regularly logged during practical sessions to detect good practices (i.e., code indentation or variables shadowing) or compilation results (i.e., success or failure). In these works, indicators are tightly coupled to the programming activity. In the LaboRem [18] or Ironmaking [19] systems dedicated to physics education, students have to input values of several parameters of different devices before launching a simulation whose output is used to analyze different physical phenomenons. The notions of actions and variation of parameters we introduced in our study apply here as well, and allow to analyze learners' behaviors by reusing both the nature of actions and learning strategies we defined. Our learning strategies thus allow to monitor learners' behaviors in a homogeneous way across different disciplines, and thus to strengthen and generalize the results we found out in our specific context.

With the constant increase of traces a system is able to collect at a higher resolution, data mining methods become salient. In particular, the sequential pattern mining we adopted, and which is used to determine the most frequent action patterns occurring among a set of action sequences [8], is becoming a common approach to better understand learners' behaviors, especially in the MOOC domain. Very closed to our works, [20] suggests a topical N-gram Model applied to two Coursera MOOCs to extract common session topics (e.g., "Browse

Course", "Assignment and Forum"), to cluster learners according to these topics, and eventually to study the difference of apparition of the topics between high- and low-grade students. Still on the dataset of Coursera MOOCs, [21] studied patterns of actions at a higher level of abstraction to distinguish between high- and low-achieving users. The authors proposed a taxonomy of exclusive MOOC user behaviors (i.e., viewer or collectors, solver, all-rounder, and bystanders) based on the observation of the number of assignments and lectures they completed, and explored their distribution through different dimensions such as engagement, time of interaction, or grades. In this research, the sequential pattern mining allowed the authors to conclude, for instance, that the population of high-achievers was mainly composed of two subgroups: solvers, that primarily hand in assignments for a grade without or poorly watching lectures, and all-rounders who diligently watch the lectures, finish the quizzes and do assignments.

Also closed to our methodology, [22] suggests an algorithm based on a combination of sequence mining techniques to identify differentially frequent patterns between two groups of students. They aimed at identifying and comparing high- and low-achievers' behaviors during productive and counter-productive learning phases. Their methodology includes (i) an algorithm based on Pex-SPAM [23] to find out a set of patterns, and (ii) the use of a piecewise linear representation algorithm to identify productive and counter-productive phases. They identified differentially frequent sequential patterns of actions that are more used by one group of learner than by the other, according to the performance learning phase. While they propose an abstract representation of actions composing the patterns, the vocabulary they employ is dedicated to MOOCs and cannot apply to remote or virtual laboratory, as in [20]. However, their abstraction approach is comparable to ours, since we used regular expressions to define learning strategy as they add specific suffix to their alphabet to express multiplicity of occurrence and relevance/irrelevance to express the relation between an action and its previous one. Also, their proposal aims at finding out patterns that tend to be significantly used by one group of students more than the other, while in our methodology, we filtered patterns based on their direct correlation with the learners' performance. Their study of relation between patterns and performance, achieved afterwards, is only applicable for performance or progress that is measured as a scalar metric and periodically assessed by the environment.

6 Conclusion

The study presented in this paper, based on data collected from an experimentation conducted in an authentic learning context, aimed at revealing relationships between learners' behaviors during practical learning situations, and their academic performance. We adopted a sequential pattern mining approach to identify correlations between several learning strategies and performance, the most significant strategies being: (i) the *progression*, when learners successfully perform actions of the same nature but more and more complex; the *reflexion* (through the consultation of help manuals) before (ii) or after (iii) the execution of a related action. These strategies seem to be representative of students of high-level performance. The data analyzed in this study only relate to interactions between learners and the resources required to achieve the practical work; some works are in progress to extend our analysis model to other data collected by the system in order to deeper investigate learners' behaviors.

While we focused here on the relations between learners' behavior and their performance, we must now deal with these links in depth, in order to analyze their causal nature, but also to compute a predictive model to help reducing failing rate. Moreover, the learning strategies depicting learners' behaviors have been defined based on analysis, but a lack of formal representation is obvious. Thus, consistent taxonomy and definitions of these strategies have to be investigated, especially by educational sciences experts, in order to provide a solid basis for behavioral studies within different learning situations. While the ITS we developed may be used to study causal relationship between learning strategy and performance, we first have to analyze its impact on learners' behavior, as much as we have to validate the visualization tool dedicated to teachers.

Finally, our remote laboratory environment also includes features dedicated to cooperative and collaborative learning [9]. Activities based on collective tasks would allow to study new research questions about learners' behavior in practical work situation, in a socio-constructivism context. The influence of learning strategies on interactions between learners, or the evolution of the strategies learners apply as they go along the learning path, are some of the research questions we plan to address in a near future.

References

1. Bunderson, E.D., Christensen, M.E.: An analysis of retention problems for female students in university computer science programs. Journal of Research on Computing in Education vol. 28(1), pp. 1–18 (1995)
2. Workman, M.: Performance and perceived effectiveness in computer-based and computer-aided education: do cognitive styles make a difference? Computers in Human Behavior vol. 20(4), pp. 517–534 (Jul 2004)
3. Blikstein, P.: Using learning analytics to assess students' behavior in open-ended programming tasks. In: Proceedings of the 1st international conference on learning analytics and knowledge, pp. 110–116. ACM (2011)
4. Vihavainen, A.: Predicting Students' Performance in an Introductory Programming Course Using Data from Students' Own Programming Process. In: 13th International Conference on Advanced Learning Technologies, pp. 498–499. IEEE (2013)
5. Wilson, B.C., Shrock, S.: Contributing to success in an introductory computer science course - a study of twelve factors. ACM SIGCSE Bulletin vol. 33(1), pp. 184–188 (2001)
6. Rountree, N., Rountree, J., Robins, A., Hannah, R.: Interacting factors that predict success and failure in a CS1 course. ACM SIGCSE Bulletin vol. 36(4), pp. 101–104 (2004)
7. Aleven, V., Mclaren, B., Roll, I., Koedinger, K.: Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. International Journal of Artificial Intelligence in Education vol. 16(2), pp. 101–128 (2006)

8. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proceedings of the Eleventh International Conference on Data Engineering., pp. 3–14. IEEE (1995)
9. Broisin, J., Venant, R., Vidal, P.: Lab4ce: a remote laboratory for computer education. International Journal of Artificial Intelligence in Education vol. 27(1), pp. 154–180 (2017)
10. Taamallah, A., Khemaja, M.: Designing and eXperiencing smart objects based learning scenarios: an approach combining IMS LD, XAPI and IoT. In: TEEM '14: Proceedings of the second International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 373–379. ACM, New York, New York, USA (Oct 2014)
11. Venant, R., Vidal, P., Broisin, J.: Evaluation of learner performance during practical activities: An experimentation in computer education. In: 16th International Conference on Advanced Learning Technologies, pp. 237–241. IEEE (2016)
12. Orduna, P., Almeida, A., Lopez-de Ipina, D., Garcia-Zubia, J.: Learning analytics on federated remote laboratories: Tips and techniques. In: IEEE (Ed.) Global Engineering Education Conference (EDUCON), pp. 299–305. IEEE (2014)
13. de Jong, T., Linn, M.C., Zacharia, Z.C.: Physical and Virtual Laboratories in Science and Engineering Education. Science vol. 340(6130), pp. 305–308 (Apr 2013)
14. Corbière, A., Choquet, C.: Re-engineering method for multimedia system in education. In: Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on, pp. 80–87. IEEE (2004)
15. Inventado, P.S., Scupelli, P.: Data-driven design pattern production: a case study on the ASSISTments online learning system. In: EuroPLoP '15: Proceedings of the 20th European Conference on Pattern Languages of Programs. Carnegie Mellon University, ACM (Jul 2015)
16. Hostetler, T.R.: Predicting student success in an introductory programming course. ACM SIGCSE Bulletin vol. 15(3), pp. 40–43 (Sep 1983)
17. Watson, C., Li, F.W.B., Godwin, J.L.: Predicting Performance in an Introductory Programming Course by Logging and Analyzing Student Programming Behavior. In: 13th International Conference on Advanced Learning Technologies, pp. 319–323. IEEE (2013)
18. Luthon, F., Larroque, B.: LaboREM—A Remote Laboratory for Game-Like Training in Electronics. IEEE Transactions on Learning Technologies vol. 8(3), pp. 311–321 (2015)
19. Babich, A., Mavrommatis, K.T.: Teaching of Complex Technological Processes Using Simulations. International Journal of Engineering Education vol. 25(2), pp. 209–220 (2009)
20. Wen, M., Rosé, C.P.: Identifying latent study habits by mining learner behavior patterns in massive open online courses. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 1983–1986. ACM (2014)
21. Anderson, A., Huttenlocher, D., Kleinberg, J., Leskovec, J.: Engaging with massive online courses. In: Proceedings of the 23rd international conference on World wide web, pp. 687–698. ACM (2014)
22. Kinnebrew, J.S., Loretz, K.M., Biswas, G.: A contextualized, differential sequence mining method to derive students' learning behavior patterns. Journal of Educational Data Mining vol. 5(1), pp. 190–219 (May 2013)
23. Ho, J., Lukov, L., Chawla, S.: Sequential pattern mining with constraints on large protein databases. In: Proceedings of the 12th International Conference on Management of Data (COMAD), pp. 89–100 (2005)