Smart Learning Environments

RESEARCH Open Access

Recognizing patterns of student's modeling behaviour patterns via process mining

Yu Wang[1], Tong Li[1,2*], Congkai Geng[1] and Yihan Wang[1]

*Correspondence:
litong@bjut.edu.cn
[1]Beijing University of Technology, Beijing, People's Republic of China
[2]Postal address: South Information Building 412, Pingleyuan 100, Chaoyang District, Beijing, People's Republic of China

Abstract

Evaluating student learning effect plays an essential role in education, which is typically done by assessing student's final deliverables. However, the student's learning process has not been properly explored in the past.

In this paper, we propose an interactive student learning effect evaluation framework which focuses on in-process learning effect evaluation. In particular, our proposal analyzes students modeling assignment based on their operation records by using techniques of frequent sequential pattern mining, user behavior analysis, feature engineering, and process mining. A comprehensive online modeling platform has been developed to enable data collection. We have carried out a case study, in which we applied our approach to a real teaching scenario, consisting of student online modeling behavior data collected from 24 students majoring in computer science. We also associate our process mining results with the numeric evaluation values. The preliminary result of case analysis has shown good potential to mine student modeling patterns and interpret their behaviors, contributing to student learning effect evaluation.

Keywords: Student behavior analysis, Learning effect evaluation, Frequent sequential pattern mining, Feature engineering

Introduction

With a growing concern in student learning effect evaluation, traditional evaluation methods like paper exam, oral presentations, and practical experiments are no longer showing good performance (Struyven et al. 2005). Final grade can not meet the current demand in learning effect evaluation, so we are focusing on the in-process learning data analysis in this paper. A comprehensive and impactful in-process student learning effect evaluation method enables a more precise and reasonable evaluation.

The first challenge in this topic is the data collection. The in-process data has been overlooked by traditional evaluation methods for a long time, so we need to first obtain useful in-process data efficiently. The other challenge is to find the proper in-process learning data analysis methods, where identifying meaningful user behavior patterns is the core problem.

Regarding the research topic, relevant researchers mainly use user behavior analysis and frequent sequential pattern mining to help analyze how well the student is performing in the learning process. User behavior analysis focuses on web log analysis, which is extracting user behavior elements and selecting useful parameters from log data. Capturing characteristics of normal user behaviors is also an important work of user behavior analysis (Morita and Shinoda 1994). If we consider about the sequential timing characteristic of in-process information, algorithms in frequent sequential pattern mining are showing excellent performance in finding useful insights from data. A suitable and effective algorithm can show both long-term changing trend and short-term violent incident (Cao et al. 2005).

We have previously proposed a comprehensive framework for evaluating student learning effect, which focuses on analyzing in-process data. We proposed the framework based on process mining techniques, and we also developed an online modeling platform *BJUTModeling* which can automatically record students' modeling operations. Besides, we carried out a case study based on real in-class student data in order to evaluate the effectiveness of our proposal.

This paper extends our previous work (Yu et al. 2019) which improve analyzing the framework for better accuracy by carrying out deeper data analysis. In particular, the contribution of this paper can be concluded as below:

- Apply process mining methods to action sequences with the purpose of revealing general characteristics.
- Associate analysis between process mining results and numeric evaluation values in order to understand student's online modeling behavior habit.
- Carry out a comprehensive case study and figure out insightful conclusions.

The rest of paper is organized as follows. "Related work" section reviews related work. Related techniques and methodology are presented in "An interactive and comprehensive framework for student learning effect evaluation" section. The newly developed online modeling platform is introduced in "Online modeling platform" section. We do a case study on real data from students in "Case study" section. At last, we conclude our research in "Conclusions" section.

Related work

User behavior analysis is widely used in e-commerce marketing strategy evaluation, user experience improvement, website anomalies detection (Jansen et al. 2000). It collects various types of data, ranges from user demographic information to actual operation parameters. Behavior event analysis model, user retention analysis model, funnel analysis model, user behavior path model, user segmentation, and click analysis model are main models in user behavior analysis (Agichtein et al. 2006). In this paper, we need to apply existing models and methods of user behavior analysis to student learning effect evaluation, in order to realize student behavior analysis.

User logs are often indexed in time order, which is called sequential data. Thus, frequent sequential pattern mining algorithms help us mine the sequential database, looking for repeating patterns that can be used to find associations between different items in the sequential dataset.

Frequent sequential pattern mining algorithms can be sorted into three categories, namely, apriori-based, pattern-growth, and early-pruning algorithms (Mabroukeh and Ezeife 2010). AriorиAll (Agrawal and Srikant 1995) and GSP (Srikant and Agrawal 1996) algorithms are apriori-based algorithms, with great capability in exploring the complete set of patterns due to its multiple times of scan of the databases. A potentially huge set of candidate sequences may cause great difficulties in mining long sequential patterns in this kind type of algorithm. FreeSpan (Han et al. 2000) algorithm is based on pattern-growth, which recursively project sequence databases into smaller projected databases by using frequent items. It then grows subsequences fragments in each projected database. PrefixSpan (Pei et al. 2001) is also a pattern-growth algorithm, more specifically, a prefix-projected sequential pattern mining algorithm. It makes less projections and quickly shrinking sequence compared with the FreeSpan algorithm.

According to the domain knowledge of this topic, we will use an enhanced PrefixSpan algorithm called PreSeqPat to figure out the useful repeating patterns in the sequential database. This enhanced algorithm can reduce the complexity of the PrefixSpan, details of which will be shown in "An interactive and comprehensive framework for student learning effect evaluation" section.

Along with the data explosion today, more and more complicated models are invented to solve the real-world problems with increasing amount of data. Unfortunately, even there are advanced algorithms emerging everyday, most computer scientists and data analysts are spending 60 to 80% time on data preparation (Turner et al. 1999). The status quo calls for the rapid development of feature engineering methods, which can significantly reduce the time and cost in the early phase of data projects, leading to a shorter and more controllable modeling process. Feature engineering is playing a vital role in machine learning and artificial intelligence, where big data is seating in the center of the research process. Currently, feature engineering has been applied in various fields like chemistry (Li et al. 2017), keyphrase extraction (Berend and Farkas 2010), fraud detection, and knowledge base construction (Ré et al. 2014).

Considering the traditional student learning effect evaluation methods we have, in-course assignments, oral presentations, and written exams are widely used to assess student's transferable skill, delivery skill, and understanding skill (Jimaa 2011). For a long time, in-process evaluation is ignored when the final grade of a course is solely consisted of the final written exam or a group project report (Black and Wiliam 1998). With the development of information techniques, using advanced techniques to obtain formative data of student learning process is achievable nowadays.

Process mining has been used in educational data mining (EDM) by some researchers. Some researchers mainly focused on the supporting data in student learning like version control system data, team wiki data, and issue tracking system data (Bogarín et al. 2018). Some researchers aimed at using process mining techniques to explain educational phenomena in order to improve educational outcomes (Romero and Ventura 2013). The method of applying process mining techniques on direct data in the field of educational data mining has great potential in evaluating student learning effect (Cairns et al. 2015). Our research will show the framework of how to apply process mining techniques to modeling data in order to recognize student's modeling behavior patterns, leading to an effective student learning effect evaluation method.

An interactive and comprehensive framework for student learning effect evaluation

Our proposed student learning effect evaluation framework consists of four modules, data collection module, data mining module, behavior analysis module, and process mining module. Details are shown in Fig. 1. Each module is laying the groundwork for later stages, giving the framework a clear data flow and reasonable structure. Main techniques for each module are be shown below in the order of actual use.

Data collection

In order to get first-hand data, we build up an online modeling platform, whose details will be discussed in "Online modeling platform" section. Data are collected from the server of the online modeling platform, consisting of two parts, log data and model data. Log data is recording all user operations on the online modeling platform, including user information and all operation parameters. Model data is recording all elements and links in the diagram, fully representing main structure of the diagram.

Model data mainly shows the final view of the model, and log data contain a great number of in-process information. Thus, we are delivering this research focusing on the log data. Along with the log data, data mining methods and user behavior analysis can be applied to model student behavior, which can bring insightful observations and conclusions from the raw data.

Data mining

Data preprocessing

Data Cleaning and Operation Separation Raw log data contains plenty of formatting characters, which are meaningless in further data processing. So we re-organize the raw log data and make it as a highly structured dataset. By this data cleaning process, operations are separated from the raw data, resulting in individual operations. This brings great convenience in later data processing.

Noise Removal Each operation recorded in log data has many columns, including user information, operation information, and operation parameters. Some of the columns are regarded as obviously redundant features, which should be removed as noise to simplify later processes. Remaining columns are vital and useful, avoiding annoying extra information.

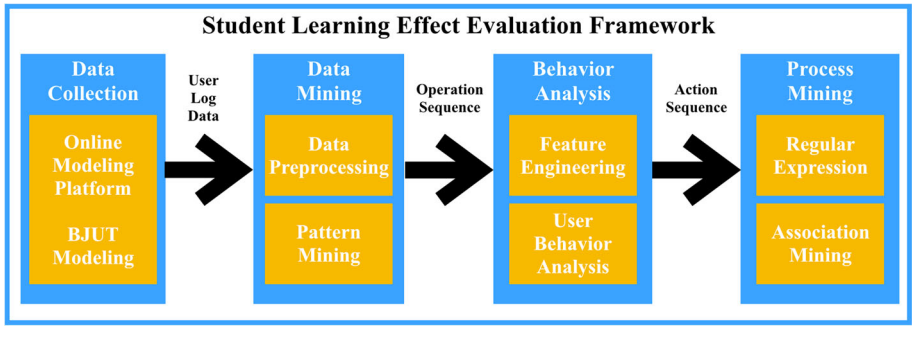

Fig. 1 Student learning effect evaluation framework

Pattern mining

Operation Symbolization According to the design of the online modeling platform, it has several types of operations, such as adding an element, moving an element, adding a link, and editing text of a link.

These operations cover all element operations and linkage operations which are essential in modeling and building a diagram. These operations are telling the user behavior in a sequential sequence, so we can symbolize these operations to make it easier in later sequential pattern mining.

The symbolized operation sequences only takes operation types into consideration and can divide the whole dataset by users. By this step, the detailed information of each operation is not considered, because we need to find the general frequent sequential patterns in the dataset first and then do the user behavior analysis individual by individual.

FreSeqPat, an enhanced PrefixSpan algorithm According to the common sense of sequential pattern mining and the specific need in this paper, we develop an algorithm called FreSeqPat based on PrefixSpan, willing to realize the frequent sequential pattern mining in this topic.

PrefixSpan algorithm owns great capability in dealing with items with more than one element inside, which means it needs to consider in-item prefix and suffix. This is vital in online shopping cart analysis and marketing strategy decision process, but not useful in our sequential database. Our sequential dataset does not have any item with more than one element inside. Thus, we improve the PrefixSpan algorithm as the FreSeqPat algorithm to reduce the size of the projected database, resulting in lower space complexity.

FreSeqPat algorithm is able to find the complete set of sequential patterns in the given sequential database. These patterns are with the appearance frequency no less than the preset threshold *min_support*. FreSeqPat checks all patterns in the sequential database with a reasonable size of projections without information loss.

The pseudo code of algorithm FreSeqPat is shown in Table 1. The difference between FreSeqPat and PrefixSpan is in the first part. FreSeqPat algorithm does not need to check whether an element is the last one in an item or not, because the sequential database only

Table 1 Pseudo code of FreSeqPat

Algorithm FreSeqPat : Frequent Sequential Pattern Mining
Input: A sequence database S, and the minimum support threshold *min_sup*.
Output: The complete set of frequent sequential patterns.
Parameters: α : a sequential pattern; l : the length of α;
$S\|\|\alpha$: the α-projected database, if $\alpha \neq <>$; Otherwise the sequence database S.
Method: Call *FreSeqPat (<>, 0, S)*
Subroutine: *FreSeqPat (α, l, $S\|\|\alpha$)*
Scan $S\|\|\alpha$;
If item $< b >$ can be appended to α to form a larger sequential pattern, **then**
put item b into set B;
For each item b in B **do**
Append b to α to form a sequential pattern α';
Put α' into set A';
For each pattern α' in A' **do**
Construct α-projected database $S\|\|\alpha'$;
Call *FreSeqPat (α', l, $S\|\|\alpha'$)*

takes one element per item. Therefore, FreSeqPat can effectively reduce the complexity of PrefixSpan but not lose any frequent pattern from the sequential database.

Behavior analysis

Feature engineering

The quality of features is of great concern in the data processing procedure. Better features mean better flexibility, simpler data processing models, and better results. Thus, feature engineering methods applied in this paper are of significant value.

Feature engineering results in two kinds of new features. Some features are directly calculated from existing features and the others are derived from features based on domain knowledge. Both kind of new features are with higher value than the original features of the raw data. New features contain more information along with higher information capacity, without losing important values. These new features can significantly reduce the complexity of modeling and analysis.

User behavior analysis

User behavior on online modeling platform are recorded as operations user log, and user operations have been symbolized as sequential symbols. Along with the pattern mining result of this sequential database, we are now able to perform user behavior analysis on it.

Due to the characteristics of each user, we realize that huge differences are existing among users. So we need to do case-by-case user behavior analysis in this paper, where each user is regarded as an individual case. In this paper, the behavior event analysis model is used to figure out how each student performs when using the online modeling platform. Also, user behavior path model is also needed to draw a general overview of online modeling behavior for each student.

Process mining

Regular expression of action sequence

With the action sequences on hand, we're now aware of the processing order for each student during his whole run-time. However, the detailed processing order can not give us more useful information about the specific student. We can then extract the general pattern from the plain sequence, which can be concluded as a process mining procedure.

In this paper, action sequences consist of capital letters. We can convert the action sequences into regular expressions of for each student using the process mining techniques. This manipulation not only keeps the independence of each student, but also helps us see the general look of the dataset from a higher level of abstraction.

Regular expressions of action sequences show a formal and compressive representation of the action sequences. Action sequences with various length can be normalized as nearly same-length regular expressions, which help us do the later work.

Association process mining with numeric evaluation value

Association process mining techniques give us great power to figure out what is the relation among items in the dataset. Nowadays, numeric evaluation values are being widely used. They have a consistent measuring standard when facing different occasions, so they own an irreplaceable position in student learning effect evaluation. But it comes with a big problem, zero possibility will occur in carrying out an in-process analysis. In this

paper, we associate our regular expressions of action sequences with numeric evaluation values. Thus, an analysis focusing on in-process data is also taking the numeric value into account. The association process mining techniques will give us insightful results considering both formative data and consequence.

Online modeling platform

In this project, we develop BJUTModeling [1], an online modeling platform. User's behavior is recorded as the log data in BJUTModeling, consisting of operation time, operation type, and operation parameters. Table 2 shows an example of the log data.

The log data serves as the data source, which will be used in the later case study. With log data on hand, we can recognize frequent sequential patterns from the data. We also need to bring in the domain knowledge, so that we can carry out the quanlitative analysis of the patterns, leading to insightful findings.

Platform design

BJUTModeling is designed in the Client/Server mode, and developed in Html, css, and JavaScript. BJUTModeling works as a online modeling website where people can sign-up, log-in, log-out, draw diagrams, edit diagrams, save diagrams, and print diagrams. The server of BJUTModeling records all user operations, including basic user information, operation parameters and other details.

The user interface (UI) design of BJUTModeling is clearly shown in Fig. 2. This figure shows a modeling page of the platform, where the shown data flow diagram (DFD) is only a sample diagram without any practical meaning. The three main components are element pool on the left, diagram canvas in the middle, and function buttons on the upper-right corner.

Platform function

On the online modeling platform BJUTModeling, users can successfully draw a diagram and build up a model. The usability and accessibility of BJUTModeling is guaranteed by test, and its main functions are listed as below:

- User sign-up and log-in.
- Data flow diagram, UML class diagram, and use case diagram.
- Clear view of available elements in diagrams.
- Easy-to-use actions of drag and drop.
- Save and print diagrams.
- Automatically record user modeling operations.

Case study

Participants

Under the design of this student learning effect evaluation method, we apply this method in a real world case analysis. This case analysis is set up in the course *Introduction to Software Engineering*. Users of BJUTModeling are 24 junior students majoring in Computer Science from Beijing University of Technology.

[1] http://bjutmodeling.com

Table 2 An example of log data on BJUTModeling

Date	Time	Operation Type	Object	Old Location	New Location
2019-04-17	21:16:19	!d Location:	-3	old:Point(86.6399,0)	new:Point(29.9531, -276.3906)

This case analysis is using the log data of the course assignment in structured analysis, requiring students to draw a data flow diagram to model a problem. It uses a real in-class dataset, consisting of 24 real student online modeling log data. The reliability and practicality of the student learning effect evaluation method is therefore confirmed under this real data case analysis.

Instruments

With the raw log data of BJUTModeling on hand, we first analyze all the 16 columns of it. We realize that 6 columns in one operation, which is an entry in log data, is obviously redundant. Thus, we drop these six noise columns in order to get a cleaned log data.

Columns left are renamed as Date, Time, Student Number, Op, Type, OpLevel, and four other operation parameters. These columns are important for data mining procedure later on, which can bring insightful observations and conclusions from this case analysis.

Methods

Symbolization of operations

According to the raw user log data of the online modeling platform BJUTModeling, we can conclude the user operations into 16 types based on columns Op, Type, and OpLevel. In order to analyze the student modeling behavior in a clear and logical way and find the frequent user pattern first, we only take the operation type into consideration now. Thus, we symbolize the 16 different user modeling operations as 16 letters, where each letter is representing a specific user modeling operation.

Re-symbolization of grouping operations

It is clear that the length of individual operation sequence is not equally distributed. Some students' operation sequence length is more than 2000, while some students' length is relatively short as 20.

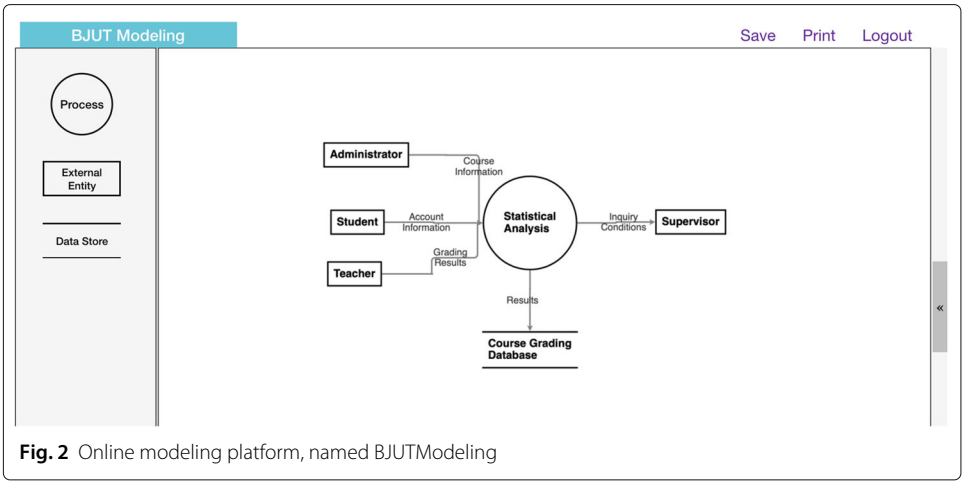

Fig. 2 Online modeling platform, named BJUTModeling

Also, under the design and construction of BJUTModeling, we figure out that some operations can be grouped to form a meaningful action in the platform. So we group these highly correlated operations in order to get much shorter sequences of action symbols without losing any information. In this case, we can group these 16 operations by their practical meanings in BJUTModeling. This re-symbolization process results in 8 new action symbols. The corresponding relationship between 16 original operations and 8 re-symbolized action symbols, along with the meaning of 8 new action symbols are listed in Table 3.

Frequent sequential pattern mining of actions

Along with the re-symbolized sequence database, doing frequent sequential pattern mining of actions is now possible. In this case, we need to set a maximum length threshold to the FreSeqPat algorithm to avoid sequence explosion. Otherwise, we will result in a too-long pattern, which is meaningless in this topic. Considering the characteristics of data flow diagram and structured analysis, the maximum length of actions in a single frequent pattern can be set as 20. With this reasonable threshold, we will get interpretable patterns but not overlong patterns.

Also, along with the modeling rules of data flow diagram and domain knowledge in modeling behavior, the parameter min_sup in the FreSeqPat algorithm is set as 4. This is to guarantee the completeness of the frequent patterns, and prevent from pattern explosion. These two constraints can avoid counting patterns with overlong length and exclude anomalies from the action sequences.

Feature engineering

By the domain knowledge of software engineering, more specifically, the data flow diagram of structured analysis, we conclude the following new features from the sequential dataset of user log:

1. Operation Time:
   Due to the characteristics of website operations, we can easily find that the operation time is really important in analyzing user behavior. Several observations can be derived from this new feature, such as whether a student is more likely to do online modeling in small steps, or a user needs to think for a long time between steps.
2. Shifting Distance of Element:
   From the overview of the operation sequence dataset, the importance of calculating a new feature of shifting distance is clear. The raw operation data only records old

Table 3 Re-symbolization of grouping operations as actions

Action	Corresponding Operations	Meaning of Action
A	mkhcfcd	Add element
M	ecd	Move element
T	fg	Edit text
D	yw	Delete element
L	nvkhab	Add link
C	ab	Change link
E	zvw	Delete link
X	x	Add diagram menu

and new locations of the element shifting operations, but not the shifting distance, so it is in great need to regard this calculation result as a new feature.

3. Proportion of moving immediately after adding:
   Thinking about the real situation when drawing a data flow diagram, it is significant to consider the fraction that a user moving the newly added element right after the element adding operations. This feature is also a great indicator of a user's online modeling habit.

Results

Proportion of non-operational log-in manipulation

The log data, recorded in the online modeling platform, contains all log-in manipulations. A log-in manipulation without any operation is regarded as a non-operational log-in manipulation. In order to see the proportion of non-operational log-in manipulations among students, we hereby define a proportion P_{non_op} as the number of non-operational log-in manipulations over the number of all log-in manipulations.

$$P_{non\_op} = \frac{\#\ non - operational\ log - in\ manipulations}{\#\ all\ log - in\ manipulations}$$

Figure 3 shows a histogram about the frequency of proportion P_{non_op}, where x-axis is given as proportion P_{non_op} and y-axis as frequency of each proportion.

P_{non_op} can show the student behavior habit. As we can read from the histogram, 18 out of 24 students, which is 75%, have operations in every log-in manipulation. This is a relatively high percentage, showing us that only a few students have the non-operational log-in manipulation. Students with non-operational log-in manipulations

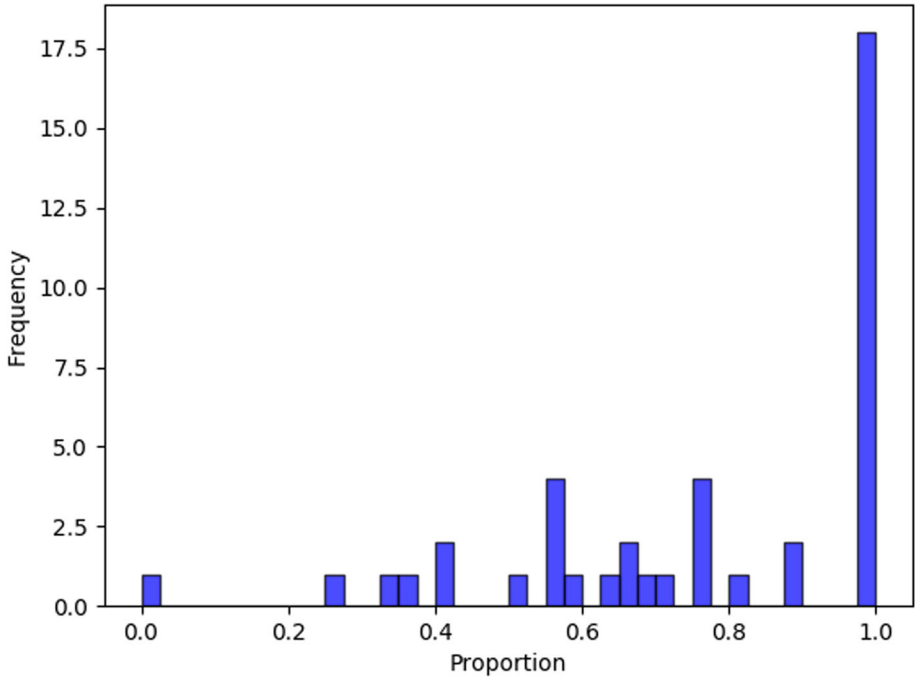

Fig. 3 Proportion of non-operational log-in manipulation

may have trouble in drawing diagrams or understanding the problem description because they sometimes leave the platform without any operations.

Types of operation

According to the structure of the online modeling platform, we use 16 letters (a, b, c, d, e, f, g, h, k, m, n, v, w, x, y, z) to represent the 16 different types of operation in the log data. These letters are called operation symbols. A single operation symbol can't represent a complete action in the platform, so we continue using the symbols here but not their practical meanings.

Figure 4 shows a histogram about the frequency of each operation symbols in the whole dataset, where x-axis represents operation types and y-axis represents the frequency of each operation type.

From the absolute frequencies of each operation symbol shown in Fig. 4, type a, b, c, d, and e are important operations. Also we can know that operation a and b are in the same frequency, while operation c, d, and e are almost in the same frequency.

Aside from the absolute values, we can also get some observations from the relative values. Some operations happen 10 to 20 times more than other operations, while some other operations only appear within 100 times. The huge range (maximal frequency - minimal frequency) gives us the necessity to group operations and analyze their actual meanings.

Types of action

In the re-symbolization procedure, highly correlated operations can be grouped into actions with practical meaning. The re-symbolization procedure results in a shorter sequences of action symbols, which can significantly decrease the processing difficulties.

The frequency of each action is a great indicator of the importance of that action. Figure 5 takes meaning of action symbols as x-axis, which can be looked up from Table 3, and frequency of each action as y-axis.

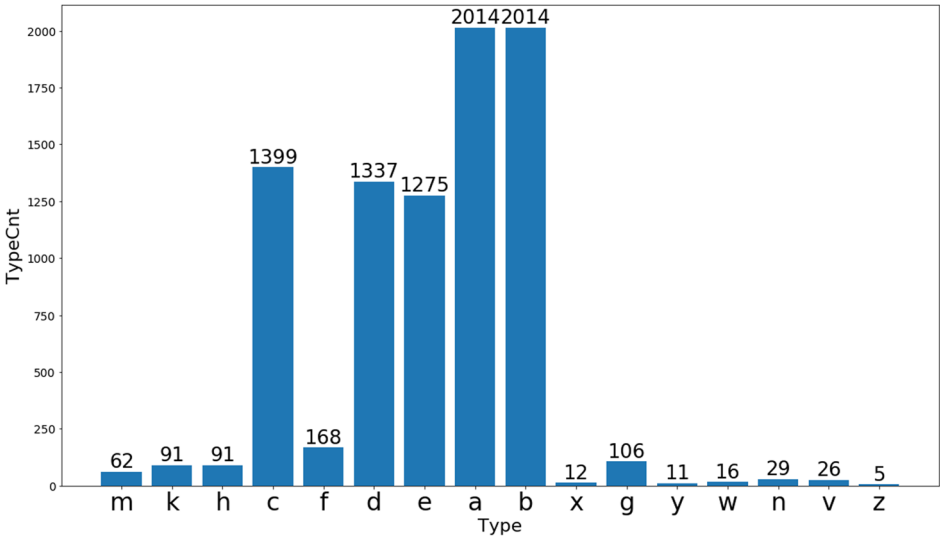

Fig. 4 Frequency of each operation

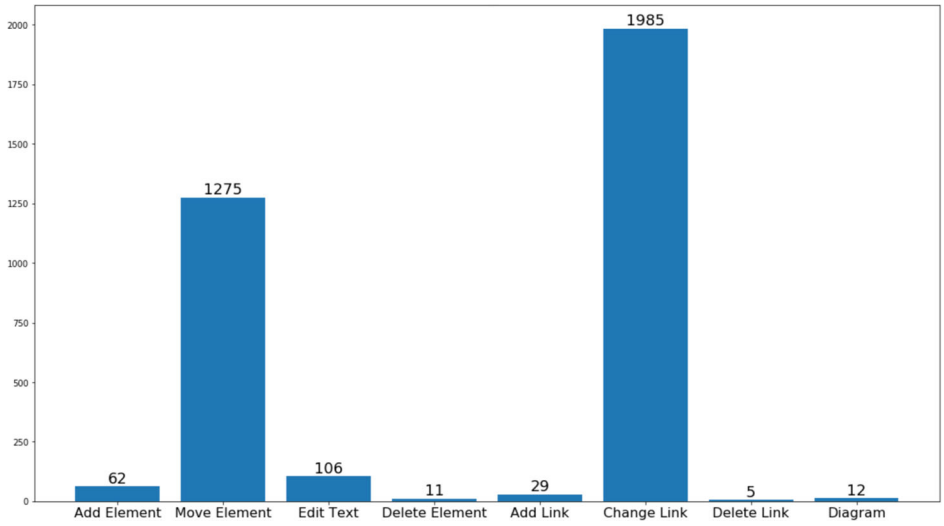

Fig. 5 Frequency of each action

As shown in the figure, action C (Change link) and M (Move Element) are in the highest frequency, which are 12 to 20 times more than other action symbol. According to the proportional value of each action symbol, link changing action (action C) takes the biggest proportion, which is of 56.96%. Element moving action (action M) takes the second largest proportion of actions as 36.59%.

The action sequences can be used in later user behavior analysis.

User behavior analysis on action sequence

According to the user log of each student, a statistical value of the total operation counts can be calculated. It is true that some of the students only own tens of operation, while some students are doing thousands of operations on the online modeling platform.

The range of this operation number per student is 2061, which is too large to be regarded as a convincing statistical value. Also, with this extremely large range, the median, mean, and mode of operation counts are of no use in this case. Thus, we need to do case-by-case analysis in this case analysis to figure out specific user habits for various kinds of students.

Take the action sequence of the student with student number 16074123 shown in Fig. 6, several insightful user behavior habits can be concluded from the action sequences, which are shown as below:

1. Type by type actions:
   Students are more likely to finish all actions in one type first, and then do all actions of another type. As shown in Fig. 6, this student firstly did action sets of changing links and changing elements (shown as action C and M), then did action sets of changing links and editing text (shown as action C and T). These sequential patterns are showing that this student prefers doing the same type of action at one time, then shifting to another type of action.
2. More actions on adjustment:

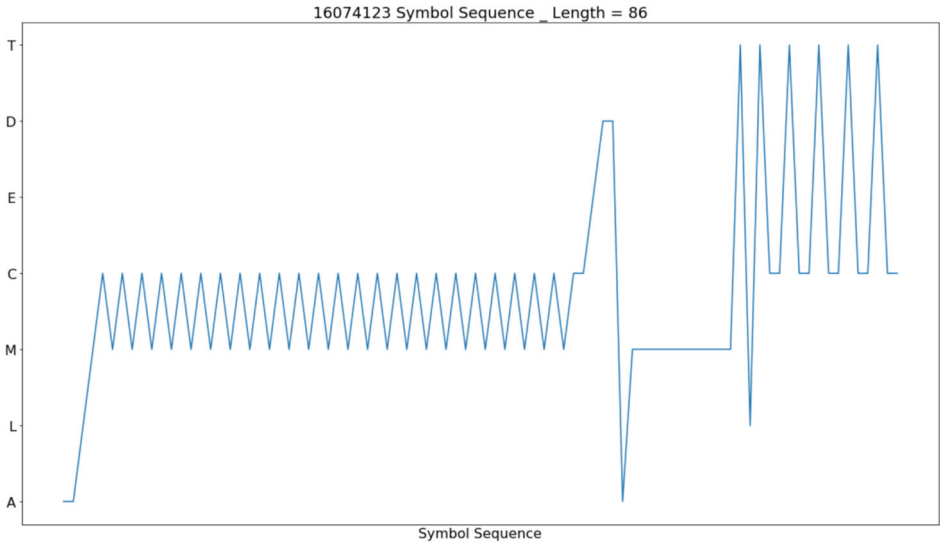

Fig. 6 Action Sequence of Student 16074123

Extremely high values are indicating another student modeling behavior pattern. The average value of changing element divided by adding element among the students is 21.2, and the average value of changing link divided by adding link is 21.4. These extremely high values confirm the observation that students tend to do much more adjustment actions than adding an element or a link.

Process mining

In order to apply process mining techniques on the action sequences, we should consider both time order and correlations between actions. According to these concerns, the regular expression is a proper way to show their general pattern in a higher level without losing their detailed information. As shown in the process mining results, some students' action sequences can be normalized into extremely short pattern in regular expression format, while some other students' action sequences are extraordinarily hard to normalize and shorten. Some typical process mining results are shown in Table 4 in the form of regular expression based on the action sequences data. As we can seen from Table 4, different students are acting in different patterns.

Two important observations can be obtained from the regular expression sequences, which are listed below:

Table 4 Regular Expressions of Action Sequences

Action Sequence Length	Student Number	Regular Expression
426	16013223	(MCC+)+
538	16041505	(MC+)+(MMMMCCCC)+C+(MC+)+
86	16074123	AAL(MC+)+EDDAM+TL(TCC)+
33	16074119	(MCCCC)+C+TC+
109	16074306	X(MC+)+
232	16074313	(MC+)+
30	16074321	(TCC+)+
912	16521210	(MC+)+

1. MC+ pattern:
   As shown in Table 4, the pattern MC+ appears in such a considerable frequency and scale among all the students. The regular expression MC+ represents the pattern that an element moving action (action M) is followed by at least one link changing action (action C). By looking into all the regular expression sequences, nearly half of the students are having MC+ pattern, so the proportion of MC+ pattern is too large to be overlooked. The MC+ pattern shows that student usually do several link changing actions right after they move an element.
2. Regularity of actions:
   Some students are acting very orderly, while some others are acting randomly. Take student number 16013223 as an example of orderly action. Even if its action sequence length is 426, which is a relatively large number in the whole dataset, its regular expression is really simple, only the repetition of pattern MCC+ (one element moving action followed by at least two link changing actions). However, if we take a look at the student number 16074123, we can see another side of the regularity. 16074123 owns a relatively short action sequence with length 86, but its regular expression is much more complicated compared with student number 16013223.

Analysis associated with student assignment scores

As we explained before, associating process mining results with numeric evaluation values is of great practical meaning. In this case, we associate the process mining results with student assignment scores. The student assignment scores is a direct measurement of the quality of the online modeling assignment, which can show the student's learning effect to some extent. For each student's action sequence, various MC+ patterns are comparing their proportion in his/her total MC+ patterns. From one aspect, the number of action C right after action M can show the student's understanding level of the modeling knowledge, which can be correlated to their assignment score.

Part of the MC+ pattern proportions and assignment scores are listed in Table 5. As an observation from the whole dataset, students with higher assignment score tend to have higher proportion of MC or MCC action pattern. At the same time, students with relatively more MCCC and MCCCC+ action pattern are more likely to get lower assignment scores in the end. This tells us that in this modeling behavior case, smaller amount of link changing action after one element moving action is more likely to lead to a better understanding of the diagram elements and the modeling background knowledge.

Threats to validity

As an empirical study, many factors may influence the validity of our case study. So we need to take these threats to the validity into consideration.

Table 5 Percentage of MC+ Patterns and Assignment Scores

Student Number	Assignment Score	MC	MCC	MCCC	MCCCC+
16013223	90	0%	77.19%	20.18%	2.63%
16041505	85	61.02%	29.66%	6.78%	2.54%
16074107	95	95.12%	4.88%	0%	0%
16074123	90	96%	4%	0%	0%
16074306	100	96.23%	3.77%	0%	0%
16074313	95	98.26%	1.74%	0%	0%

- External validity : In this case study, we're using the student data from BJUT. However, the case study may derive some different results when we collect the data from another university or from another program.
- Construction validity : In this case study, we're using data collected from our own-developed online modeling platform. However, if some sudden bugs occur on the server, the recorded data may have some fault.

Conclusions

This paper proposes an interactive and comprehensive framework focusing on in-process student learning data to evaluate student learning effect. Based on process mining methods, it uses several advanced techniques including feature engineering, frequent sequential pattern mining, and user behavior analysis. Moreover, we build up an online modeling platform which supports three types of online diagram drawing. It can automatically record user modeling operation data. We also carry out a case study on a real course Introduction to Software Engineering with data collected from the platform. It gives out insightful conclusions about student learning effect and behavior.

In the future, the framework will use more advanced process mining techniques. Higher level formalization method will be applied in this project, leading to a representative pattern for student online modeling behavior. Besides, this framework will be applied in various fields to evaluate student learning effect including business project management, big data engineering, and software testing. Last but not least, more real in-process student learning data will be gathered from the online modeling platform, so that the case study can derive more convincing and meaningful conclusions and improve the stability and robustness of the framework.

Acknowledgements
Not applicable.

Authors' contributions
The second author proposed the student learning effect evaluation framework and set up its structure. The first author developed the detailed methodology of the framework, carried out a case study based on the real-class in-process data, and was a major contributor in writing the manuscript. The third and fourth author built up the online modeling platform for data collection, and kept maintaining the platform for long-term use. All authors have read and approved the final manuscript.

Funding
This work is supported by Beijing Education Science Planning Funding (No. CCHA18148), International Research Cooperation Seed Fund of Beijing University of Technology (No. 2018B2), and Basic Research Funding of Beijing University of Technology (No. 040000546318516).

Availability of data and materials
The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests
The authors declare that they have no competing interests.

Received: 19 September 2019 Accepted: 15 November 2019
Published online: 23 December 2019

References

Agichtein, E., Brill, E., Dumais, S. (2006). Improving web search ranking by incorporating user behavior information, In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 19–26). New York: ACM.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns, In *International Conference on Data Engineering* (pp. 3–14). New York: IEEE.

Berend, G., & Farkas, R. (2010). Sztergak: Feature engineering for keyphrase extraction, In *Proceedings of the 5th International Workshop on Semantic Evaluation* (pp. 186–189). Stroudsburg: Association for Computational Linguistics.

Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, *5*(1), 7–74.
Bogarín, A., Cerezo, R., Romero, C. (2018). A survey on educational process mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(1), 1230.
Cairns, A.H., Gueni, B., Fhima, M., Cairns, A., David, S., Khelifa, N. (2015). Process mining in the education domain, In *International Journal on Advances in Intelligent Systems*, 8. USA: IARIA.
Cao, H., Mamoulis, N., Cheung, D.W. (2005). Mining frequent spatio-temporal sequential patterns, In *Fifth IEEE International Conference on Data Mining (ICDM'05)* (pp. 8–11). New York: IEEE.
Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U. (2000). Freespan: frequent pattern-projected sequential pattern mining, In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 355–359). New York: ACM.
Jansen, B.J., Spink, A., Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, *36*(2), 207–227.
Jimaa, S. (2011). The impact of assessment on students learning. *Procedia-Social and Behavioral Sciences*, *28*, 718–721.
Li, Z., Ma, X., Xin, H. (2017). Feature engineering of machine-learning chemisorption models for catalyst design. *Catalysis Today*, *280*, 232–238.
Mabroukeh, N.R., & Ezeife, C.I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, *43*(1), 3.
Morita, M., & Shinoda, Y. (1994). Information filtering based on user behavior analysis and best match text retrieval, In *SIGIR'94* (pp. 272–281). Heidelberg: Springer.
Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H. (2001). Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth, In *Proceedings 17th International Conference on Data Engineering* (pp. 215–224). New Yrok: IEEE.
Ré, C., Sadeghian, A.A., Shan, Z., Shin, J., Wang, F., Wu, S., Zhang, C. (2014). Feature engineering for knowledge base construction. CoRR abs/1407.6439. *arXiv preprint arXiv:1407.6439*.
Romero, C., & Ventura, S. (2013). Data mining in education. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *3*(1), 12–27.
Srikant, R., & Agrawal, R. (1996). Mining sequential patterns: Generalizations and performance improvements, In *International Conference on Extending Database Technology* (pp. 1–17). Heidelberg: Springer.
Struyven, K., Dochy, F., Janssens, S. (2005). Students' perceptions about evaluation and assessment in higher education: a review. *Assessment & Evaluation in Higher Education*, *30*(4), 325–341.
Turner, C.R., Fuggetta, A., Lavazza, L., Wolf, A.L. (1999). A conceptual basis for feature engineering. *Journal of Systems and Software*, *49*(1), 3–15.
Yu, W., Tong, L., Congkai, G., Yihan, W. (2019). Evaluating student learning effect based on process mining, In *ICAI2019*. Heidelberg: Springer.

Publisher's Note