



# Evaluating Student Learning Effect Based on Process Mining

Yu Wang, Tong Li<sup>(✉)</sup>, Congkai Geng, and Yihan Wang

Beijing University of Technology, Beijing, China  
sallywangyu42@gmail.com, litong@bjut.edu.cn,  
{gengcongkai789123,17071121}@emails.bjut.edu.cn

**Abstract.** As education is taking an increasingly significant role in society today, efficient and precise evaluation of student learning effect is calling for more attention. With recent advances of information technology, learning effect can now be evaluated via mining student's learning process. This paper proposes 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, and feature engineering. In order to enable effective student learning evaluation and deliver practical value, we have developed a comprehensive online modeling platform to collect operation data of modelers and to support the corresponding analysis. We have carried out a case study, in which we applied our approach to a real dataset, consisting of student online modeling behavior data collected from 24 students majoring in computer science. The results of our analysis show that our approach can effectively and practically mine student modeling patterns and interpret their behaviors, contributing to assessment of their learning effect.

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

## 1 Introduction

Student learning effect evaluation has become a great concern today as education is playing a more and more significant role in the new era of information technology. Traditional student evaluation methods regard paper exams, practical experiments, and oral presentations as components of the final grade. However, such traditional methods only focus on evaluating students' final deliverables, but cannot capture and analyze the in-process learning data. With the rapid development in computer science and data analytic techniques [1] like process mining, we argue that a comprehensive and impactful in-process student learning effect evaluation method is of great need to enable a more precise evaluation.

A first challenge to this topic is how to efficiently obtain useful in-process data, which has been ignored by traditional learning effect evaluation methods.

Another challenge is concerning how to effectively analyze such in-process data, e.g., identification of meaningful behavior patterns.

Regarding this 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 is focusing 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 [2]. Considering 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 shows both long-term changing trend and short-term violent incident [3].

In this paper, we propose a comprehensive framework for evaluating student learning effect, which focuses on analyzing in-process data. For one thing, we propose to record detailed student modeling behaviors by developing an online modeling platform. For another, we apply and customize process mining techniques [4] to figure out reasonable student behavior patterns and useful insights about student learning effect evaluation. In particular, the contribution of this paper can be concluded as below:

- Propose a student learning effect evaluation framework based on process mining techniques.
- Develop an online modeling platform *BJUTModeling* which can automatically record students' modeling operations.
- Carry out a comprehensive case study based on real in-class data from students in order to evaluate the effectiveness of our proposal.

The rest of paper is organized as follows. Section 2 reviews related work. Related techniques and methodology are presented in Sect. 3. The newly developed online modeling platform is introduced in Sect. 4. We do a case study on real data from students in Sect. 5. At last, we conclude our research in Sect. 6.

## 2 Related Work

User behavior analysis is widely used in e-commerce marketing strategy evaluation, user experience improvement, website anomalies detection [5]. 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 [6]. 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 [7]. ArioriAll [8] and GSP [9] 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 [10] 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 [11] 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 Sect. 3.

### 3 An Interactive and Comprehensive Framework for Evaluating Student Learning Effect

#### 3.1 Framework Design

Our proposed student learning effect evaluation framework consists of four modules, online modeling platform, data mining module, feature engineering module, and user behavior analysis module. Details are shown in Fig. 1.

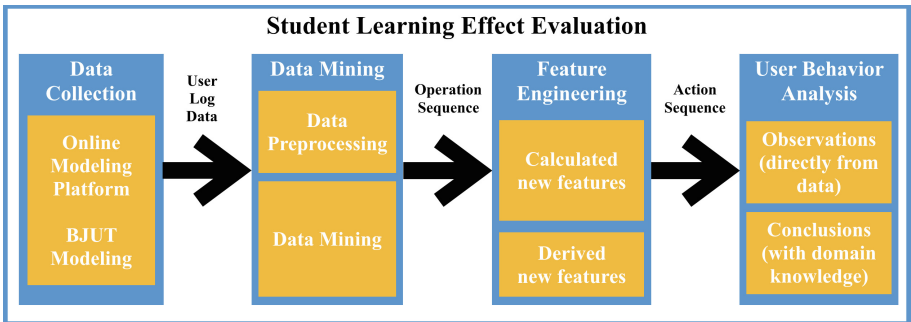


Fig. 1. Student learning effect evaluation framework.

#### 3.2 Data Collection

In this paper, we build up an online modeling platform, whose details will be discussed in Sect. 4. 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.

### 3.3 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.

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

**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:  $\alpha$  : a sequential pattern;  $l$  : the length of  $\alpha$ ;  
 $S||\alpha$  : the  $\alpha$ -projected database, if  $\alpha \neq \langle \rangle$ ; Otherwise the sequence database  $S$ .

---

Method: Call *FreSeqPat* ( $\langle \rangle$ ,  $0$ ,  $S$ )  
Subroutine: *FreSeqPat* ( $\alpha$ ,  $l$ ,  $S||\alpha$ )  
Scan  $S||\alpha$ ;  
**If** item  $\langle b \rangle$  can be appended to  $\alpha$  to form a larger sequential pattern, **then**  
    put item  $b$  into set  $B$ ;  
**For** each item  $b$  in  $B$  **do**  
    Append  $b$  to  $\alpha$  to form a sequential pattern  $\alpha'$ ;  
    Put  $\alpha'$  into set  $A'$ ;  
**For** each pattern  $\alpha'$  in  $A'$  **do**  
    Construct  $\alpha$ -projected database  $S||\alpha'$ ;  
    Call *FreSeqPat* ( $\alpha'$ ,  $l$ ,  $S||\alpha'$ )

---

### 3.5 Generate New Features from Feature Engineering

As the machine learning and data mining algorithms are being used in more fields, the quality of features is of great concern. It is clear that better features mean better flexibility, simpler data processing models, and better results [12]. Thus, feature engineering methods applied in this paper are of significant value.

Feature engineering results in two kinds of new features, features directly calculated from existing features and features 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. These new features can significantly reduce the complexity of modeling and analysis.

### 3.6 User Behavior Analysis

User behavior on online modeling platform are recorded as operations user log, and user operations have been symbolized as sequential sequences. 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 there huge differences are existing among users. So we need to do case-by-case user behavior analysis in this paper, where each user are regarded as an individual case. In this paper, the behavior even 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 in this paper to draw a general overview of online modeling behavior for each student.

## 4 Online Modeling Platform

### 4.1 Platform Design

In this project, we develop BJUTModeling<sup>1</sup>, an online modeling platform. The modeling page of BJUTModeling is shown in Fig. 2, using data flow diagram as an example. The user interface is clearly shown in the figure, and we can easily see how it works. This platform is the data source of this paper, from where we get the log data of students.

BJUTModeling is designed in the Client/Server mode, where the server records all user operations, including basic user information, operation parameters and other details. It is coded with Html, css, and JavaScript, realizing the online modeling function. It works as a website, where people can sign-up as a user and then log-in to the platform to finish the online modeling behavior.

### 4.2 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.

---

<sup>1</sup> <http://bjutmodeling.com>.

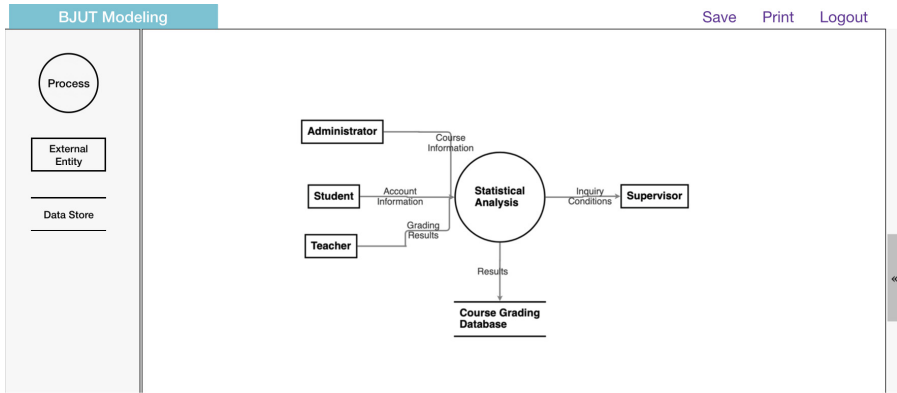


Fig. 2. Online modeling platform BJUTModeling.

## 5 Case Study

### 5.1 Case Study Design

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.

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. The case analysis is using 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.

### 5.2 Data Preprocessing

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.

### 5.3 Pattern Mining

**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 logic way and find the frequent user pattern first, we only take the operation type into consideration now. Thus, we symbolize these user modeling operations into 16 letters, where each letter is representing a 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.

Also, under the design and construction of BJUTModeling, we figure out that some operations can be grouped. Grouping operations are representing one particular action on the online modeling platform. So we can group these highly correlated operations to get much shorter sequences without losing information. In this case, we can group these 16 operations into 8 groups, which are re-symbolized as 8 new actions. The detailed information of grouping operation, actions, and meaning are list in Table 2.

**Table 2.** Re-symbolization of grouping operations.

Actions	Grouping operations	Meaning
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

**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 results 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.



## 5.4 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, whether a user needs to think for a long time between steps.

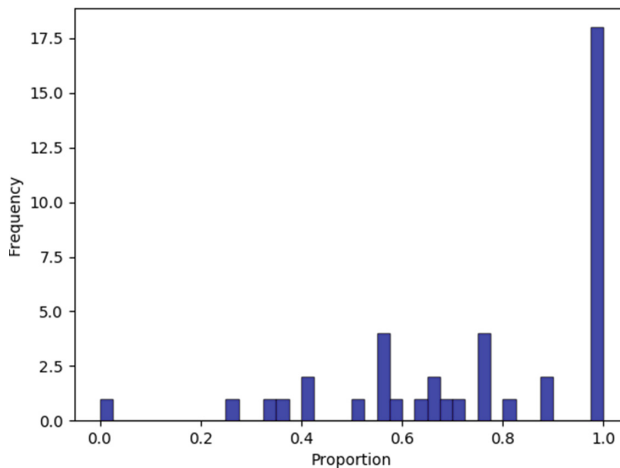
### 2. Shifting Distance of Element:

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 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. Fraction of moving right 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.

## 5.5 Data Analysis Result and Knowledge Interpretation



**Fig. 3.** Proportion of non-operation log-in manipulation.

**Proportion of Non-operation Log-in Manipulation.** Every log-in manipulation is recorded in raw user log data. Thus, a proportion of non-operation log-in manipulations divided by all log-in manipulations can be calculated. Figure 3 clearly shows the statistical histogram of this proportion.

As we can read from the histogram, 75% of the students has operations in each log-in manipulation. This is a relatively high proportion, showing us that only a few students do the log-in manipulation without any operations. This can be concluded as a user behavior habit of this case analysis focusing on student online modeling behavior.

**Types of Operation.** According to the 16 types of operation on the online modeling platform, a counting value of appearance frequency of each operation type can be derived to show the importance of each type of operation.

From result shown in Fig. 4, the operation type a, b, c, d, 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 are happening 10 to 20 times more than others, while other operations are only appearing within 100 times. This huge gap between types of operations gives us the necessity to analyze the actual meaning of operations.

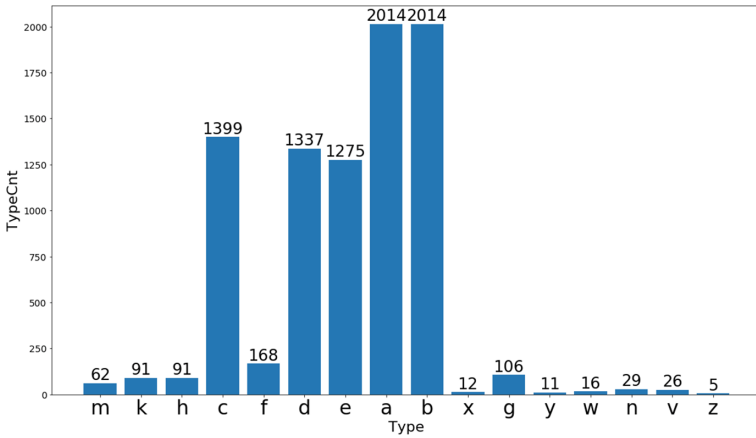
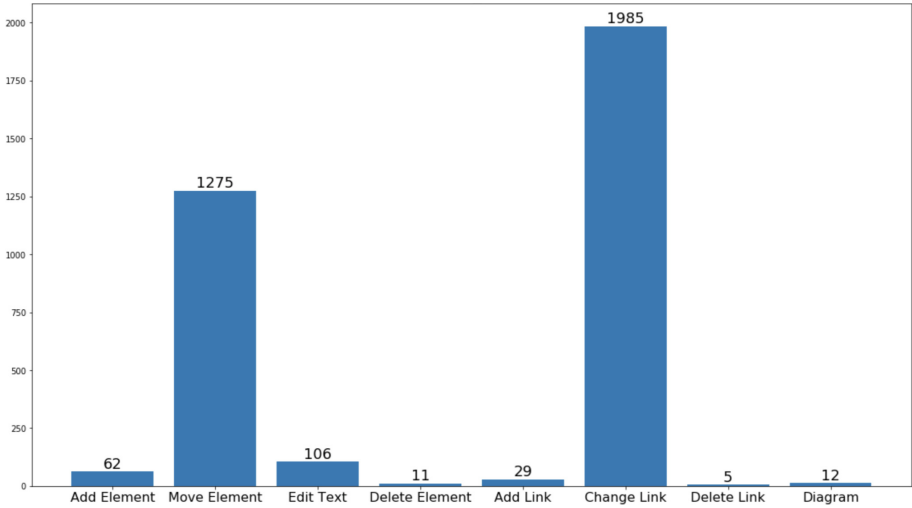


Fig. 4. Frequency of each operation type.

**Types of Action.** In the re-symbolization procedure, highly correlated operations are grouped as actions. A counting value of appearance frequency of each type of action is a great indicator of its importance. These 8 actions can significantly decrease the processing difficulties and improve the quality of data analysis result.

From Fig. 5 we can know that actions of moving element and changing link are in the highest frequency, which are 12 to 20 times more than other types of action.



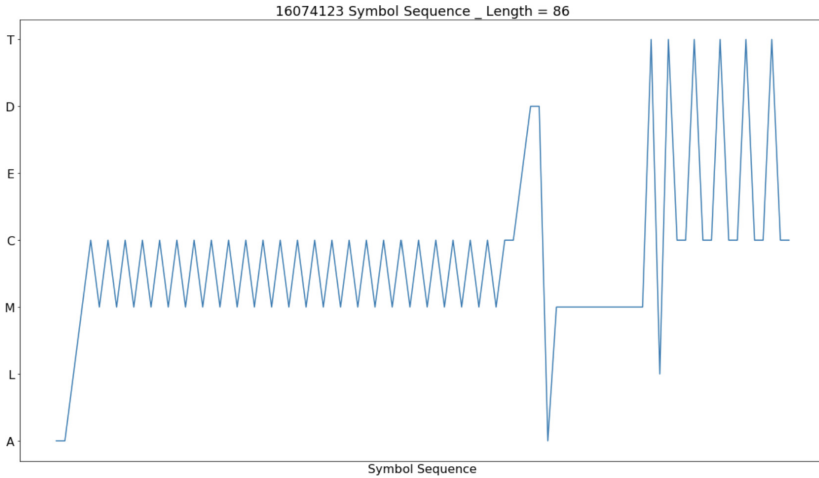
**Fig. 5.** Frequency of each action type.

According to the proportion value of each type of action, link changing action takes the biggest proportion, which is of 56.96%, and element moving action takes the second largest proportion of actions as 36.59%.

It can be concluded that students are taking more steps in adjusting elements and links rather than creating elements and links. This seems to be a common sense, but the actual user data tells us the truth that the frequency of adjusting elements and links is nearly 13 times more than the frequency of creating and deleting elements and links. This number is extremely high, giving us a great indication that students are actually not so that sure about their modeling actions.

**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.



**Fig. 6.** Action sequence of student 16074123.

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:

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 after adding an element or a link.

## 6 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, 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 behavior.

In the future, the framework will use more advanced process mining techniques and be applied in various fields to evaluate student learning effect. In addition, more real in-process student learning data will be put into the case study to derive useful conclusions and improve the stability and robustness of the framework.

**Acknowledgements.** 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).

## References

1. Baghaei, N., Nehring, N., Dacey, S.: Improving students' performance through gamification: a user study. In: Proceedings of the 10th International Conference on Computer Supported Education, vol. 1, pp. 213–218 (2018)
2. Morita, M., Shinoda, Y.: Information filtering based on user behavior analysis and best match text retrieval. In: Croft, B.W., van Rijsbergen, C.J. (eds.) SIGIR 1994, pp. 272–281. Springer, London (1994). [https://doi.org/10.1007/978-1-4471-2099-5\\_28](https://doi.org/10.1007/978-1-4471-2099-5_28)
3. Cao, H., Mamoulis, N., Cheung, D.W.: Mining frequent spatio-temporal sequential patterns. In: Fifth IEEE International Conference on Data Mining (ICDM 2005), pp. 8–11. IEEE (2005)
4. Van Der Aalst, W.: Process Mining: Discovery, Conformance and Enhancement of Business Processes, 2nd edn. Springer, Heidelberg (2011). <https://doi.org/10.1007/978-3-642-19345-3>
5. Jansen, B.J., Spink, A., Saracevic, T.: Real life, real users, and real needs: a study and analysis of user queries on the web. *Inf. Process. Manag.* **36**(2), 207–227 (2000)
6. Agichtein, E., Brill, E., Dumais, S.: 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. ACM (2006)
7. Mabroukeh, N.R., Ezeife, C.I.: A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv. (CSUR)* **43**(1), 3 (2010)
8. Agrawal, R., Srikant, R.: Mining sequential patterns. In: International Conference on Data Engineering, pp. 3–14. IEEE (1995)
9. Srikant, R., Agrawal, R.: Mining sequential patterns: generalizations and performance improvements. In: Apers, P., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 1–17. Springer, Heidelberg (1996). <https://doi.org/10.1007/BFb0014140>
10. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U.: 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. ACM (2000)

11. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H.: PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. In: Proceedings 17th International Conference on Data Engineering, pp. 215–224. IEEE (2001)
12. Turner, C.R., Fuggetta, A., Lavazza, L., Wolf, A.L.: A conceptual basis for feature engineering. *J. Syst. Softw.* **49**(1), 3–15 (1999)