



Dropout Prediction in MOOC Combining Behavioral Sequence Characteristics

Xiaoxuan Ma¹, Huan Huang^{2(✉)}, Shuai Yuan³, and Rui Hou¹

¹ School of Computer Science, South-Central University for Nationalities, Wuhan, China

hourui@mail.scuec.edu.cn

² School of Education, South-Central University for Nationalities, Wuhan, China

huanghuan@mail.scuec.edu.cn

³ School of Computer, Hubei University of Education, Wuhan, China

Abstract. In the past decade, online education platforms led by MOOC have developed rapidly around the world, bringing great changes to the education industry. MOOC aim to provide high-quality, free and open courses for global learners. However, different from the traditional classroom education, MOOC suffers from a significant high dropout rate due to its online mode. In previous studies, researchers mostly use some well-designed features by handcraft. Such methods can be time-consuming and complicated. In this paper, we combine the unsupervised algorithm with machine learning algorithm to solve the problem of dropout prediction in MOOC. Our model use the sub-sequences identified in the participant's behavior sequence as features, which simplifies the complexity of the features design. And a large number of experiments have been carried out on a public datasets, the experimental results show that the performance of the proposed method can be compared with the method using the high-dimensional and complex features used by other researchers.

Keywords: MOOC · Dropout Prediction · Behavioral Sequence

1 Introduction

Massive Open Online Courses (commonly known as MOOC) are open courses designed to provide world-wide educational courses to a large number of learners through an online platform [1]. Most MOOC courses are free and of high quality, which are usually recorded at great universities in various countries. Since 2010, MOOC have gradually come into the public view, and the rapid development of online learning platforms led by MOOC has become a new popular trend. Relies on the real-time and high-quality courses offered by the MOOC platform, learners can freely choose the learning content they are interested in and the learning time more suitable for them. At the same time, learners can constantly adjust their learning progress or choose different learning content in time according to their

learning habits. These are advantages that MOOC online learning platform have that traditional classroom learning does not have. The most obvious feature of MOOC is that they offer a variety of courses, and is not limited by the number of learners, so the same course can accommodate more participants compared to traditional classroom learning. It is also these factors that have caused a widespread problem in MOOC platforms, that is only a small number of people finish the course. MOOC has high dropout rates [2]. And it is reported that the completion rate on Coursera is only 7–9% [3].

We know that an unavoidable problem of online education platforms is that learners have too much autonomy in learning, which leads to their failure to complete the course. This is also a common problem faced by many online education platforms. In recent years, it has attracted wide attention from scholars.

Generally, these studies about dropout prediction can be divided into two categories. In the first category, the traditional machine learning model is used for modeling to predict the dropout rates. Researchers select the datasets they are interested in, implement feature engineering to extract the effective features, and use these features for modeling to predict the dropout rates.

In traditional machine learning approach, researchers use handcrafted features. Researchers need to construct complex features to get a good prediction result, It can also be understood intuitively that more features can better represent the underlying rules of the data. However, manual feature design is a time-consuming and laborious task. And an effective feature extraction strategy is not universal for different datasets. Some researchers use the number of single behavior features in click-stream data to represent the data. In this way, the extracted features is independent and lose the temporal relation in behavior sequence to some extent. The following studies prove the validity of our point. In [12]. They use raw click-stream data without this feature engineering, claiming that it removes an important sequential pattern of the click-stream. Still, these end-to-end models may overlook important patterns in the data by taking only a single objective into account. In [13], this naturally leads researchers to consider unsupervised methods to capture meaningful patterns.

The second category is the method of neural networks, it can automatically extract features during training. Using neural networks approach is the hot spot of recent years, researchers often use multi-layer network or combination of different network model. The method of neural networks improve the performance of the prediction. However, it still has problems in this way. That is, the complex network structure and the poor interpretability of the model, what scale of deep learning network model should be used for different datasets, and also a lot of uncertainty about parameter tuning of the model.

For those two reasons, this paper try to construct a more effective dropout prediction model from the perspective of traditional feature engineering. In order to achieve this purpose, this paper only adopts the most widespread records as features in feature engineering. These features are the data that users interact with the MOOC platform. Therefore, our feature engineering can be easily implemented on similar datasets based on user behavior.

The structure of the paper is as follows. Section 2 provides a related work on what has been researched about dropout prediction in MOOC. Section 3 introduce the datasets used in this paper. Section 4 introduces the proposed method in this paper. Results and discussion are in Sect. 5. Finally, the conclusions are described in Sect. 6.

2 Related Work

In this section, we briefly review some significant results of MOOC dropout prediction research in recent years.

In [1], they present an approach that works on click-stream data. And their algorithm takes the weekly history of student data into account and thus is able to notice changes in student behavior over time. In [4], they build predictive models weekly, over multiple offerings of a course. Based on logistic regression, they propose two transfer learning algorithms to trade-off smoothness and accuracy by adding a regularization term to minimize the difference of failure probabilities between consecutive weeks. In [5], a decision tree was used to predict dropout and perform feature analysis. In [6], a survival model was developed to measure the influence of factors related to student behavior and social positioning within discussion forums using standard social network analysis techniques. In [14], they extract features mainly from discussion forums and video lectures, and employ Hidden Markov Models (HMMs) to predict student retention.

Also, there are some researchers using ensemble methods to improve the prediction ability of the model, such as in [7], a composition and ensemble of the naive Bayes (NB), multilayered perceptron (MLP), SVM, and decision table (DT) was used to give the final dropout prediction. Some researchers have used more information about participants to help improve the model's predictive ability, such as In [8], they used students related variables (gender, age, grade in high school and so on) collected during the admission of student and pre-university information (high school marks, family income of students, parents' qualification), and they tested a variety of decision tree models, and the results show that using students' relevant information can improve the performance of the model.

Neural network has become popular in recent years, with the proposed and application of CNN and RNN network models, many researchers have built prediction models based on these neural networks, such as: In [9], by regarding dropout prediction as a sequence classification problem, they propose some temporal models for solving it. And they propose a recurrent neural network (RNN) model with long short-term memory (LSTM) to predict dropout. Through extensive experiments on a public dataset, they show that the proposed model can achieve results comparable to feature engineering based methods. In [10], their model is a deep neural network, which is a combination of Convolutional Neural Networks and Recurrent Neural Networks. In their model, features are extracted automatically from raw records by convolutional and pooling layers in the lower part of the model. Characteristics of time series data are considered by recurrent layer in the upper part of the model. Experimental results show that their

model can achieve comparable results to those obtained by feature engineering based methods. In [11], they propose a deep neural network model, which is a combination of Convolutional Neural Network, Long Short-Term Memory network and Support Vector Machine in a bottom-up manner. Also their model can automatically extract features from the raw data, moreover they takes into account the impact of the sequential relationship of student behavior and class imbalance on dropout, and the model they proposed reinforce the performance of dropout prediction. In [9–11], researcher used deep learning model to extract features automatically, and their model really achieve good results compared to feature engineering based methods.

3 Dataset

This section introduces the dataset used for the experiment in this article. In order to prove the effectiveness of our proposed method, we select a public dataset as the research data: the dataset of KDD Cup 2015.

This dataset contains information about 39 courses in the online Platform Xuetangx and these courses all last one month. And it directly give the label of each participant whether the participant is dropout or not, which simplifies our discussion of how to define dropout and allows us to focus more on the problem solving itself.

This dataset contains three csv files: enrollment_train.csv, log_train.csv (see Table 1), and truth_train.csv, where, enrollment_train records the student’s participation with the course, with enrollment_id as the unique identifier; Log_train records the behavior record of participants interacting with the MOOC system. Here are seven types of behavioral data (see Table 2). This table is also uniquely identified by enrollment_id; Truth_train is the real label for the dataset, also uniquely identified with enrollment_id. So enrollment_id represents the different participant, our task is to give the dropout prediction for every participant based on their behavior records.

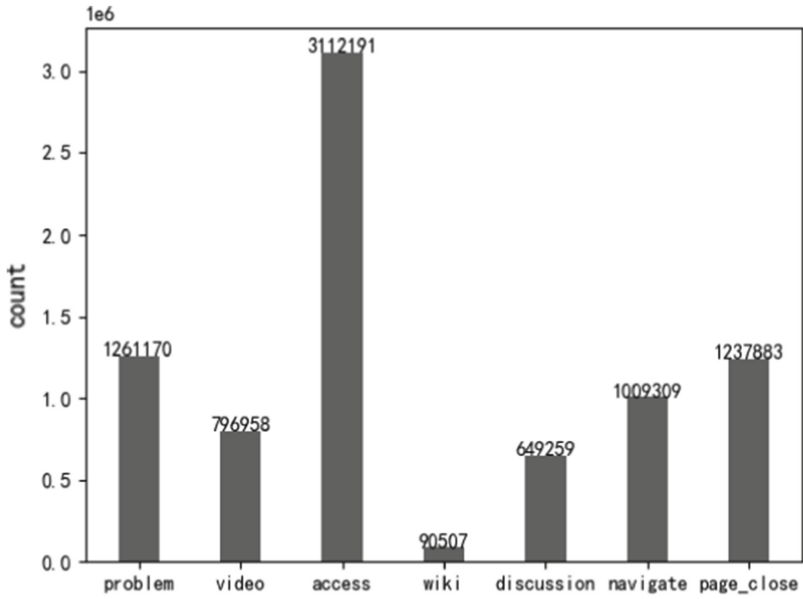
The size of this dataset is as follows: 39 courses, 120,542 enrollment_ids, 815,7277 events. The distribution of behavior in the events is shown in Fig. 1.

Table 1. Table in log_train.csv.

Attribute	Meaning
enrollment_id	id of the participant
time	the time of the event
source	event source (“server” or “browser”)
event	behavior record for participant
object	object participant access or navigate to

Table 2. Behavior data of students

Attribute	Meaning
problem	working on course assignments
video	watching course videos
access	accessing other course objects except videos and assignments
wiki	accessing the course wiki
discussion	accessing the course forum
navigate	navigating to other part of the course
page_close	closing the web page

**Fig. 1.** Distributions of seven behavior records.

4 Proposed Method

4.1 Overview of the Method

From the previous work, we can know that the dropout prediction problem is actually a sequence prediction problem. Our task is to take a sequence of the participant, which is generated by participant's learning behaviors in time order, and then use classification model to predict whether the participant will eventually dropout. This can be summarized as a time series binary classification problem. The following is the overview of proposed method:

First of all, we divide the whole datasets into training datasets and testing datasets by 4:1. Therefore, 80% of the data is taken as training datasets and 20% of the data is taken as testing datasets. Then, we do preprocess on dataset. The preprocessing includes two parts. The first part is carried out on the whole datasets to aggregate the behavior records of participants from raw datasets into behavior sequences, and each participant corresponds to one behavior sequence. The second part is only carried out on the training datasets. Based on the time of the each behavior record, the behavior record are aggregated into different sub-sequences, each sub-sequence corresponds to one learning process, so that each participant will correspond to several sub-sequences. The whole sequence of learning behaviors obtained in this part are taken as the input of the Sequence Pattern Mining Algorithm, and the frequent sub-sequences are obtained from the output. These frequent sub-sequences will be used as the important features in our model. Then, we use four common classification algorithms to model, train the model through the training datasets, and verify the performance of the model on the testing datasets (Fig. 2).

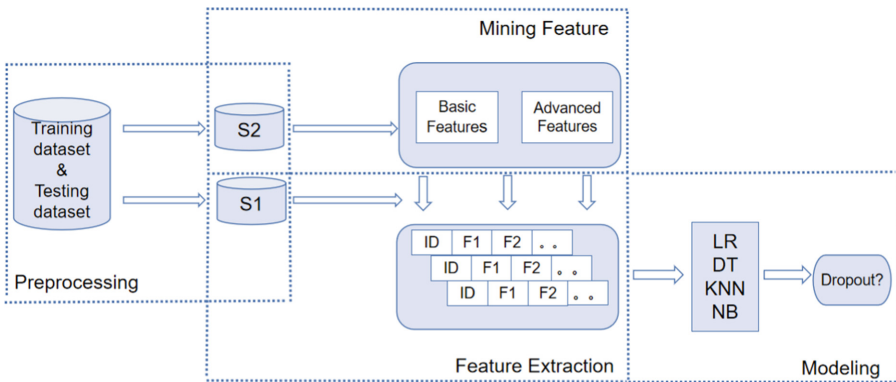


Fig. 2. Distributions of seven behavior records.

4.2 Data Preprocessing

First of all, we extract enrollment_id, time, event from log_train.csv file. Then the extracted data is cleaned to prepare for the next step of processing. Among them, “event” contains seven behavior types, and each type of behavior corresponds to one type of operation of the participant interacting with the learning system, and for the convenience of processing, we map these seven learning behaviors into 1 to 7 in the following order: problem, video, access, wiki, discussion, navigate, and page_close. The following is the main process of preprocessing, which is divided into two parts:

Part I: In this part, we directly aggregated the behavior records of each participant into an behavior sequence in chronological order, so that the learning activity of each participant was corresponding to one behavior sequence, and we could obtain the behavior sequence of all participants in this way. The experimental results generated in this part can be expressed as follows: The behavior sequence of the participant with Enrollment_id 1 is, and all behavior sequences are denote as dataset S1. For example, the behavior sequence corresponding to the participant with Enrollment ID 1 was 63331..., the length of this behavior sequence is 314.

Part II: In this part, we aggregate the participants' behavior records to varying degrees. We also draw some lessons from the previous studies, the researchers aggregate the behavior records of the participants in a day or a week's time, which make the behavior sequence more tidy on the format, but still not enough to find more underlying rule because the time granularity is not small enough. Therefore, with the time of each behavior record takes place, we aggregate the records with a time interval of no more than 30 min between adjacent behaviors in a day into one sequence. In this way, each participant's learning activity corresponds to several sequences, and each sequence represents one learning process. (The time interval used to divide is an empirical value adopted on the basis of fully studying the rules of the dataset.) The experimental results generated in this part can be expressed as follows: the behavior sequence corresponding to the participant with enrollment_id i is, where i is participant's total number of learning processes. All behavior sequences are denote as dataset S2. And both the S1 and S2 datasets will be used in Sect. 4.3.

4.3 Mining Features

In this section, we describe the feature engineering in detail. The features adopted in this paper are mainly composed of two parts, one is the basic features, the other is the advanced features obtained through the Sequence Pattern Mining Algorithm. There are nine basic features, seven of them are the frequency of seven basic learning behaviors, which have been introduced in Part 3, including problem, video, access, wiki, discussion, navigate, and page_close. The other two basic features can be obtained through the processing in the second part of Sect. 4.2. One is the total number of the participant's learning process, represented by t_i , where i is the enrollment_id. The other is the total time of participants' learning activities, represented by T_i , $T_i = \sum T_{it_i}$, where T_{it_i} is the duration of the t_i th learning process for the participant whose enrollment_id is i .

Advanced features are obtained through the Sequence Pattern Mining Algorithm. Sequence pattern mining is a kind of association analysis algorithm in data mining. Different from ordinary association analysis, inputting sequence of this algorithm is ordered, and the output sub-sequences are also ordered. Sequence pattern mining refers to the knowledge discovery process, which aim to find frequent sub-sequences as patterns from the original sequence dataset, that is, inputting a sequence dataset and outputting the sub-sequences that

are not less than the minimum support degree. Therefore, the sub-sequences obtained through sequence mining can represent more valuable information contained in the original sequence. In this paper, the dataset adopted for sequence pattern mining is the S2 dataset obtained in Sect. 4.2, which contains all behavior sequences of all participants in the training dataset. In this paper, we use the idea of Apriori algorithm to design a simple sequence mining algorithm. Starting with seven single learning behaviors as a candidate item set, the whole S2 dataset is searched and count the support degree for each candidate item. The items whose support degree is greater than the minimum support threshold are entered into the next iteration as candidate items. In the process of generating the candidate binomial set, we only combine the items whose support degree is greater than the minimum support threshold to generate the candidate binomial set, and then iterate the above steps. Finally, the resulting frequent sub-sequence is used as the advanced features. Where, the support degree is the proportion of sub-sequences occurring in the entire dataset. The formula is as follows:

$$\text{Support}(seq) = P(seq) = \frac{\text{number}(seq)}{\text{num}(\text{AllSamples})} \quad (1)$$

4.4 Feature Extraction and Modeling

Firstly, we introduce the details of feature mining. The basic features obtained in Sect. 4.3 include the frequency of the seven basic behaviors, as well as the number of participants' learning processes and the total time of participants' learning activities. In the training stage, the frequency of the seven basic behaviors needs to be obtained from the training dataset in S1, and the other two basic features are obtained from the training dataset in S2. The pattern sub-sequences used as the advanced features are obtained from the training dataset in S2, which are a series of frequent sub-sequences of participants' behavior sequences. We calculate the frequency of these pattern sub-sequences occurred in S1, then we get the advanced features values. The extracted basic features and advanced features are taken as the features of participants, and all of these features are used for training. In the test stage, similar to the processing in the training stage, the frequency of the seven basic behaviors needs to be obtained from the testing dataset in S1, and the other two basic features are obtained from the testing dataset in S2. The advanced features are obtained by using the sub-sequence extracted in the training stage to get their frequency from the testing dataset in S1. The basic features and advanced features obtained in this way are used as features of participants, and all of these features are used for testing.

In modeling, we adopted four common classification algorithms, including Logistic Regression, Decision Tree, K-Nearest Neighbor and Gaussian Naive Bayes. These four models are briefly introduced as follows: Logistic Regression: Logistic Regression is a commonly used classification algorithm in machine learning. Its principle is to classify different data by fitting a decision boundary, which can be expressed as: $w_1x_1 + w_2x_2 + \dots + w_nx_n + b = 0$, Suppose $h_w(x) = w_1x_1 + w_2x_2 + \dots + w_nx_n + b < 0$ represents that sample X belongs to

category 0, and then when $h_w(x) > 0$, means that sample X belongs to category 1. Logical regression algorithm adds a layer of sigmoid function on this basis, so that $0 \leq h_w(x) \leq 1$. The final logistic regression calculation formula is:

$$y = \frac{1}{1 + e^{-(w^T x + b)}} \quad (2)$$

Decision Tree: Decision Tree is a commonly used classification algorithm. A decision tree is a tree-shaped structure in which each internal node represents a judgment on an attribute, each branch represents the output of a judgment result, and finally each leaf node represents a classification result. The decision tree classifies the samples by their different judgment results on each attribute.

K-Nearest Neighbor: K-nearest Neighbor algorithm is one of the simplest machine learning algorithms. The idea of this method is as follows: In the feature space, if most of the nearest k samples near a sample belong to a certain category, then the sample also belongs to this category.

Gaussian Naive Bayes theorem: Naive Bayes theory hypothesis that each input variable is independent of each other. And it build the model through the calculation of the probability of each category denoted as $P(C_j)$, and conditional probability of each attribute denoted as $P(A_i | C_j)$.

5 Experiment

The experiment of this paper mainly includes dataset preprocessing, feature mining, feature extraction, modeling, training and testing with four classification algorithms. All the codes used in the experiment were coded in Python language, and the four classification algorithms were coded with the package of Scikit-Learn Algorithm. In the training process, we use the GridSearchCV module provided in Scikit-Learn to tune the model's parameters.

Section 5.1 is divided into three parts. Section 5.1.1 introduces feature mining, experimental details in the modeling, and parameter tuning strategies in the modeling; Sect. 5.1.2 introduces the criteria for model performance evaluation adopted in the experiment. Section 5.2 is the analysis of the experimental results in this paper and the comparison with the experimental results we referenced.

5.1 Experiment Settings

Experiment Settings and Details. This section describes the setup of the experiment, including details of the feature mining and modeling. In Sect. 4.3, we have introduced the feature mining method proposed in this paper, including the mining of basic features and advanced features, in which the basic features can be obtained only by further processing on the basis of S1 and S2 datasets, and the process is not complicated. However, advanced features, that are frequent pattern sub-sequences contained in participants' learning activity sequences, need to be obtained by repeated iteration of sequence pattern mining algorithm on the basis

of S2 dataset. Therefore, in feature mining experiments, we mainly introduce the details of this part of the experiment.

The key step in the sequence pattern mining algorithm is calculating the support degree of candidate sub-sequences and select the candidate for the next round according to the support degree result. Here we start from seven individual behavior as the initial candidates, according to the output of the algorithm, we make heuristic choices. And this selection method is very simple, that is to divide the candidate sequence into a group with high support and a group with low support according to the experimental result. The threshold for dividing needs to be determined according to the specific results of the experiment.

Here we directly present the experimental results obtained according to the advanced features mining method in Sect. 4.3, and the results are shown in the following three figures:

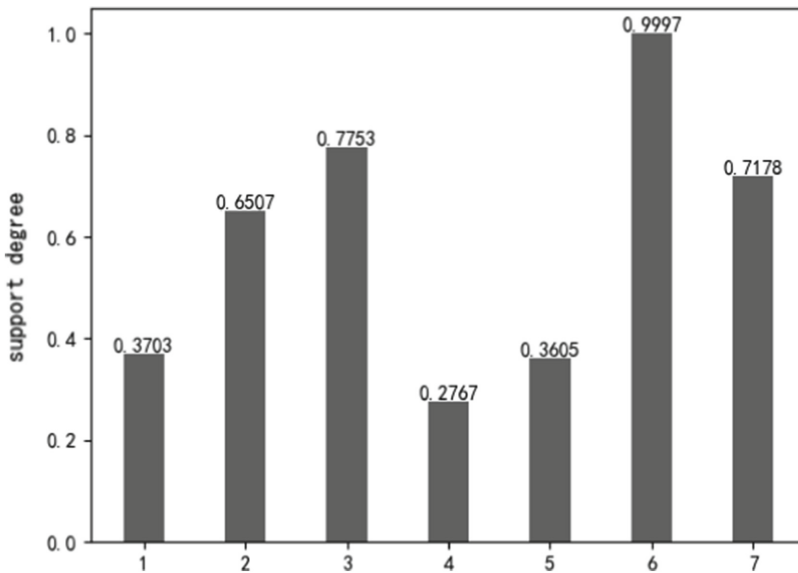


Fig. 3. Support degree of one element item set.

It can be clearly concluded from Fig. 4 that, in one element item set, we should select 2, 3, 6 and 7 to enter the next iteration of the algorithm. Then, we combine them into the binomial sequences: 23, 26, 27, 32, 33, 36, 37, 62, 63, 66, 67, 77, 72, 73, 76, 76, 77. These sequences are fed into the algorithm to calculate their support degree, and the same heuristic rules mentioned above are used to select the binomial sequences as the next round candidates. Then do the same iteration. According to the results in Fig. 3, 4 and 5, we set the minimum support threshold as 50% to obtain the pattern sub-sequence that meets our requirements. Finally, we obtained two sets of advanced features:

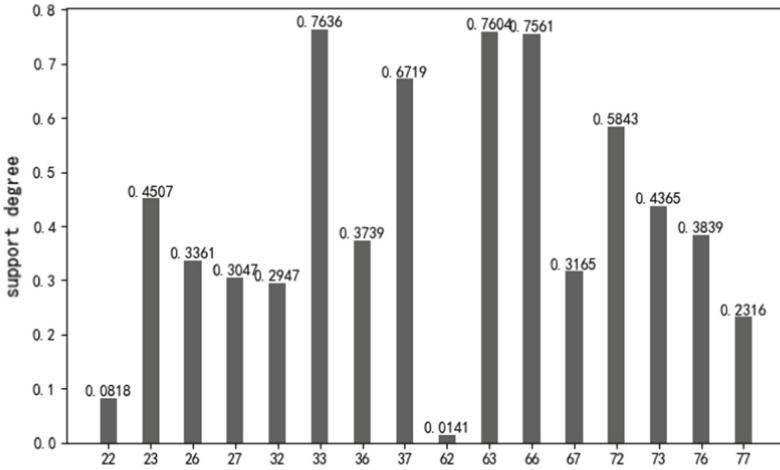


Fig. 4. Support degree of two elements item.

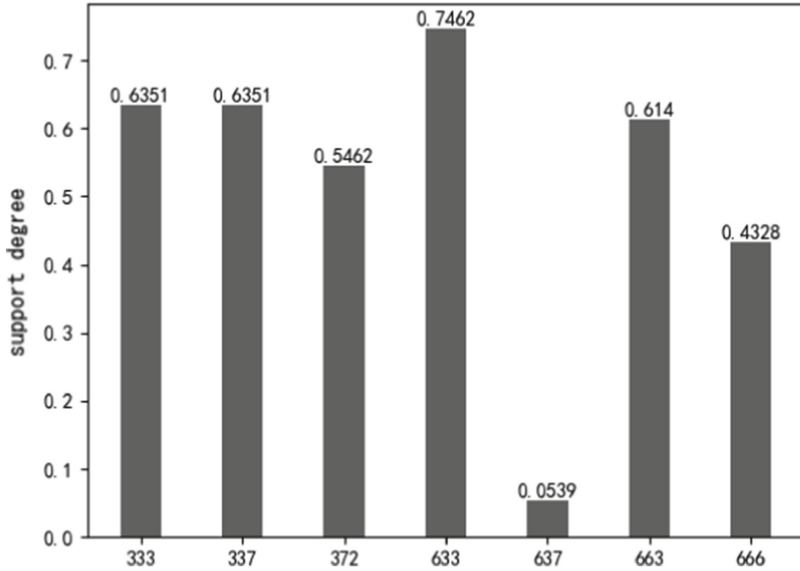


Fig. 5. Support degree of three elements item.

the two elements sub-sequences are 33, 37, 63, 66, 72; The three elements sub-sequences are 333, 337, 372, 633, 663. We refer the nine basic features mentioned above as feature set A, the five binomial sequences as feature set B, and the five trinomial sequences as feature set C.

In the experiment above, we get three groups of progressive feature sets, which are A, A+B and A+B+C respectively. The three feature sets are respec-

tively used to build the model on the basis of the four classification algorithms, so that 12 models are finally obtained. The performance of the 12 prediction models will be verified on the testing datasets. The experimental results are given in Sect. 5.2 (Table 3).

Table 3. Parameter tuning.

Model	Parameter	Meaning	Range
LR	C	the reciprocal of the regularization coefficient λ	(0.1–0.5,0.1)
DT	Max_depth, Criterion	the maximum depth, creation strategy	(1,21),(entropy, gini)
KNN	Weights, N_neighbors	consider the weight of distance or not, number of neighbors	(uniform, distance), (1–11,1)
NB	none	none	none

Evaluation Metrics. The datasets used in our experiment showed obvious class imbalance, and the single indicator of Accuracy was no longer sufficient to accurately measure the performance of the model. Therefore, multiple indexes are used together as the prediction indicators of the model, Including Precision, Recall, F1-score and area under the Receiver Operating Characteristic (ROC) curve (AUC).

5.2 Results and Discussion

In order to prove the performance of the method proposed in this paper, we refer to the experimental results of this papers in [10] for comparison. The datasets used in these their papers is the same as ours, which makes our comparison more meaningful. In their paper, the author extracted 186 features by manual feature extraction method, and used these features to train the machine learning model as their baseline. Moreover, the author proposed a novel deep learning method. We cited some of their results as comparison.

From our experimental results, the method of feature engineering proposed in this paper has played a certain effect on the performance improvement of the

Table 4. Some experimental results in [10].

Method	Precision	Recall	F1-score	Auc
LR	89.17	96.12	92.52	87.70
DT	84.57	97.87	90.74	80.03
NB	89.89	92.52	91.18	77.40
CNN-RNN	88.62	96.55	92.41	87.42

Table 5. Experimental results in our experiment.

Method	Precision	Recall	F1-score	Auc
LR(A)	85.11	95.71	90.10	69.91
LR(A+B)	85.40	95.70	90.25	70.55
LR(A+B+C)	85.41	95.69	90.26	70.57
DT(A)	91.04	95.24	93.09	81.98
DT(A+B)	91.21	95.53	93.32	82.41
DT(A+B+C)	91.29	95.50	93.35	82.54
KNN(A)	93.07	96.40	94.70	86.22
KNN(A+B)	93.50	96.76	95.11	87.16
KNN(A+B+C)	93.60	96.89	95.21	87.38
NB(A)	85.31	94.82	89.82	70.18
NB(A+B)	86.38	93.99	90.02	72.26
NB(A+B+C)	86.77	93.64	90.07	72.99

four models. Taking the experimental results of DT model as an example, the Auc-score is 81.98% when only the basic feature set A is used. When feature sets A and B are used, the Auc-score is 82.41%. When all feature sets A, B, and C are used, the Auc-score is 82.54%. Similarly, this trend of improvement appears on the other three models, which indicates that the advanced features that we extract can improve the performance of the model (Tables 4 and 5).

By comparing the experimental results of the same model in the paper we quoted, it can be found that the performance of our proposed method on DT and KNN models is comparable to its best results, in which our Auc-score on DT and KNN model is higher, and our performance on LR model and NB model is poor. However, we should point out that the experiment in this paper adopts a simple feature extraction scheme, and the maximum number of features used in the experiment is only 19. Compared with the 186 features manually extracted in the citation paper, it is obvious that the feature engineering scheme in this paper is more efficient. At the same time, the experiments in this paper have shown good results on the KNN model, which is comparable to the performance of the depth model proposed in the citation paper.

6 Conclusion

In this paper, the Sequence Pattern Mining Algorithm is used to extract the subsequences contained in the long sequence, which can better represent the hidden rules in the sequence of learning behaviors. At the same time, the features used in this paper are all from the behavioral data and do not include other types of additional data recorded by the platform. Through extensive experiments, we can find that the method proposed in this paper has a good performance on DT and KNN based models. In future work, we hope to verify the performance of

the model on more datasets. In addition, we will further study and analyze the performance of our feature mining method on different classification algorithms.

References

1. Kloft, M., Stiehler, F., Zheng, Z., Pinkwart, N.: Predicting MOOC dropout over weeks using machine learning methods. In: Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs (2014)
2. Khalil, H., Ebner, M.: MOOCs completion rates and possible methods to improve retention - a literature review. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications 2014 (ED-Media) (2014)
3. Peng, D., Aggarwal, G.: Modeling MOOC dropouts. *Entropy* **10**(114), 1–5 (2015)
4. Bailey, J., Zhang, R., Rubinstein, B., He, J.: Identifying at-risk students in massive open online courses. In: AAAI (2015)
5. Sharkey, M., Sanders, R.: A process for predicting MOOC attrition, pp. 50–54 (2014). <https://doi.org/10.3115/v1/W14-4109>
6. Yang, D., Sinha, T., Adamson, D., Rose, C.P.: “Turn on, tune in, drop out”: anticipating student dropouts in massive open online courses. In: NIPS Workshop on Data Driven Education (2013)
7. Manhães, L., Cruz, S., Zimbrão, G.: Evaluating performance and dropouts of undergraduates using educational data mining (2014)
8. Pal, S.: Mining educational data to reduce dropout rates of engineering students. *Int. J. Inf. Eng. Electron. Bus.* **4**(2), 1–7 (2012)
9. Fei, M., Yeung, D.Y.: Temporal models for predicting student dropout in massive open online courses. In: IEEE International Conference on Data Mining Workshop (2015)
10. Wei, W., Han, Y., Miao, C.: Deep model for dropout prediction in MOOCs. In: The 2nd International Conference (2017)
11. Wu, N., Zhang, L., Gao, Y., Zhang, M., Sun, X., Feng, J.: CLMS-Net: dropout prediction in MOOCs with deep learning, pp. 1–6 (2019). <https://doi.org/10.1145/3321408.3322848>
12. Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., Tingley, D.: Delving deeper into MOOC student dropout prediction (2017)
13. Sinha, T., Li, N., Jermann, P., Dillenbourg, P.: Capturing “attrition intensifying” structural traits from didactic interaction sequences of MOOC learners. In: EMNLP Workshop on Modelling Large Scale Social Interaction in Massive Open Online Courses [SHARED TASK WINNER] (2014)
14. Balakrishnan, Eecs, G.: Predicting student retention in massive open online courses using hidden Markov models (2013)