Feature Engineering for Predicting MOOC Performance

Nadirah Mohamad[1], Nor Bahiah Ahmad[2], Dayang Norhayati Abang Jawawi[3] and Siti Zaiton Mohd Hashim[4]
[1,2,3,4]School of Computing, Faculty of Engineering,
Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia

E-mail: nadirah23@live.utm.my[1]

Abstract. Increasing data recorded in massive open online course (MOOC) requires more automated analysis. The analysis, which includes making student's prediction requires better strategy to produce good features and reduces prediction error. This paper presents the process of feature engineering for predicting MOOC student's performance utilizing deep feature synthesis (DFS) method. The experiment produces features which all the top features selected using principal component analysis (PCA) are the features that are generated from method. In terms of prediction comparing both based features and generated features, the result shows better accuracy for generated features proposed using k-nearest neighbours technique which shows the method potential to be used for future prediction model.

1. Introduction

In the last few years, academician and learning analyst used to carefully observes the file and data log to investigate the students' online activities which usually consist of small groups [1]. However, traditional method resulted in increasing workloads and consuming time [2] which is no longer suitable and required more automated tools for learning analytic especially for online learning with large students' records, massive open online course (MOOC). One of the main reasons applying analytic is to monitor and predict student's performance to provide appropriate intervention program for students. Various factors can affect student's performance in MOOC such as student's frequency accessing activities, student's interaction level with peers or instructor, or student's time management [3-5]. The factors become the basis for the features constructed for prediction. However, mostly, the features being used are limited, thus underutilized the other features that have potential to be generated to produce better prediction [6]. Meanwhile, feature engineering as one important component of data pre-processing has been implemented in various fields such as health, biology, financial and transport to cater the issue [7]. The aim is to produce more quality features that can support better and less error prediction and can reduces feature exploration time to deliver results of analytic on time. Therefore, in this study, automate features is generated through feature engineering to support better prediction.

2. Related studies

Feature engineering is the process of constructing features that clean and transform raw data, involving the feature transformation and feature aggregation and has been known to be performed by domain experts [8]. For student's prediction, several works [9-11] that have discussed on feature engineering discussed more on generating based features. In educational data mining, most of the features used are based features which uses hand-craft strategy is known to be high quality features as the features are constructed based on the pedagogical theory [12,13]. However, there is still limited

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

research on MOOC prediction that use generated features which have the potential to complement existing explored features. Meanwhile, Kanter and Veeramachaneni [14] presented deep feature synthesis (DFS) which generates features based on historical information using the list of chosen functions according to level. The higher level of feature, the deeper synthesis applied. Apart from improving prediction, the project aimed at reducing the workload on data pre-processing with more automate process for raw data and provides more effective works which have been applied for various fields. A study [15] has used DFS for fraud detection which helped them to reduce the false positive percentage and saved 190K euros. Meanwhile, in medical, the use of DFS has potential to aid analysis of microelectrode recording (MER) data [16]. Therefore, based on performance reported using DFS, the method is chosen for this study. Apart from that, feature engineering process involves generation of features derived from the raw data points which is mostly relevant with educational data, where most online platform stored user's raw data on online activities in the form of events. The events are commonly used for constructing proxy feature or based feature which represent pedagogy theory. This reason makes DFS covers both human-craft strategy and learned strategy. Thus, this paper adopted DFS for feature engineering in effort to enhance student's performance prediction whether assisting the process of the prediction itself or improving the result of prediction.

3. Methodology

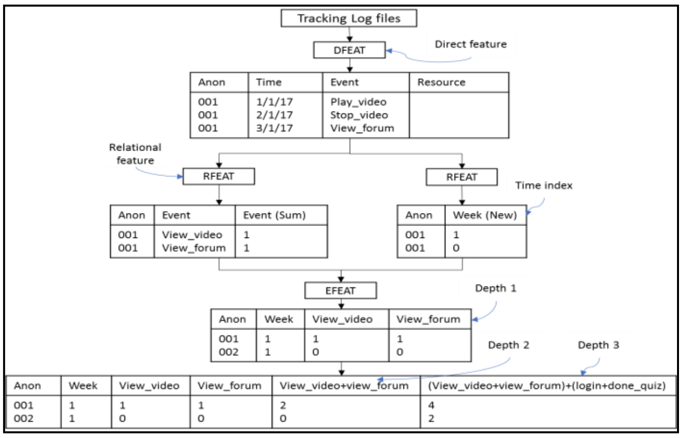
All the files will go through pre-processing before the feature engineering process. Then, the study proceeds with feature reduction and treating imbalance data. Lastly, the prediction comparing both based features and generated features using the DFS method is implemented.

3.1. Data collection and data pre-processing

The model for student's prediction is applied on online tracking log from Center for Advanced Research through Online Learning, Stanford that granted access to the MOOC data which based on edX (one of the popular MOOC platforms). Four main files are selected which are students' events file, frequency of login file, student's quizzes activities file and students with grade file. Two datasets for this study is named as x_course and y_course. Further explanation on the process can be referred to our previous work [17]. After setting up the datasets, the data is loaded where the four files are matched up to produce final list of candidates. All records with missing values are removed. For x_course, the events file left with 66665 records and final grade file shows 20 students' records. Meanwhile, for y_course, the events file left with 812696 records and final grade file shows 583 students' records. There are 10 students who have received certificates and 10 students did not receive certificates for x_course. Meanwhile, for y_course there are 570 students who have received certificate and 13 students did not received certificates. After finalizing records, formatting instance is implemented such as changing the date format for standardization among files.

3.2. Deep feature synthesis

After the pre-processing phase, feature engineering is implemented using DFS. The process of constructing generated features is using available DFS package which built for python integrated development environment (IDE). DFS aimed at producing features using three types of features namely *direct features (dfeat)*, *relational features (rfeat)* and *entity features (efeat)* [15]. The process for MOOC dataset is depicted in figure 1 and is explained next.



Anon	Time	Event	Resource
001 001 001	1/1/17 2/1/17 3/1/17	Play_video Stop_video View_forum	

Anon	Event	Event (Sum)
001 001	View_video View_forum	1 1

Anon	Week (New)
001 001	1 0

Anon	Week	View_video	View_forum
001 002	1 1	1 0	1 0

Anon	Week	View_video	View_forum	View_video+view_forum	(View_video+view_forum)+(login+done_quiz)
001 002	1 1	1 0	1 0	2 0	4 2

Figure 1. Example of feature engineering process using events file.

Dfeat is needed other than *rfeat* in order to produce *efeat* later. Time delineation is applied where the learning period is divided into weekly. Therefore, an instance week is constructed for every file which act as a time index. This instance is applied using *dfeat* concept where the week instance is based on time which is taken from *event* file. Other instances derived from the four main files are all the based features required for prediction. The based features constructed in this study is based on the most produced features from previous studies [16]. The features include number of times student log in to the course *(login_freq),* number of times student has done quiz *(done_quiz)*, number of times student load a video (*view_video),* number of times student view forum *(view_forum)* and number of weeks student spent time on the course (*weeks_active)*. *Rfeat* is applied on weekly dataset which utilizing time index to reach deep feature 2. The weekly dataset that consist of *student id, week, login_freq, done_quiz, view_video, view_forum* and *weeks_active*; which are all the features for each student for each week. Summation is applied to produce the final dataset for based feature that represent each student's online activities. *Efeat* produces new generated features which will reach deep feature 3. For *efeat*, two types of processes involved which is aggregation and transform processes. In this study, the aggregation process chosen is *mean, max* and *min* while for transform process, subtract and divide is chosen based on previous study and its practicality [14]. At the end, 257 new generated features with 5 based features has been generated for *x_course*. And the same process applies with *y_course.*

3.3. Feature reduction and treating imbalanced data

Next, feature reduction is implemented to select only the best features among the 262 features to be included in the model for prediction. The features combine both based features and generated features. The technique used for reducing the number of features is the principal component analysis (PCA) which is known as one of the techniques that able to propose several combinations with the best features that presented through correlations among dataset. As the result, the study observed that *y_course* has imbalanced dataset. Therefore, one of the popular techniques which is Synthetic Minority Over-sampling Technique (SMOTE) [10] has been applied with oversampling strategy to increase *no certificate* class which is found to be very low. After implementation, *y_course* is now consisting of 570 with *yes certificate* class and 533 for *no certificate* class.

3.4. Prediction and validation

For prediction, to identify the best technique, top classification techniques are chosen which include decision tree (J48 and DecisionStump), OneR and Lazy technique. The techniques are then validated based on accuracy or specifically correctly classified instance (CCI) and root mean square error (RMSE) to measure their accuracy. The datasets are analysed with class label of whether the student will receive the certificate or not. For the x_course chosen in this study, the pass marks for certificate is 70% while for y_course, the pass mark is 60%. Then, using the best technique, based features is compared with generated features using DFS method.

4. Result and discussion

In this section, two important parts is explained. First, the study discusses on DFS implementation for MOOC. Second, the study explains the result of comparison between using based features with generated features using DFS method to expose potential of using generated features for MOOC.

4.1. Deep feature synthesis for MOOC

In terms of feature engineering, compare to certain datasets in fields like business, retail, or banking, the available main file for online learning especially MOOC data that records students' events or online hits, contains no instance with numerical value such as number of orders made, or amount of income. Commonly available instances are student id, date and time, type of events and type of resources. Instead of deriving feature from available instance, the structure requires a whole new space to construct new feature.

4.1.1. Construction of based features

Based features utilize *dfeat* and *rfeat*. Instead of directly call data from a file, MOOC feature is based on selected activity that need to be extracted first. The activities are listed under event_type instance as shown in table 1. Therefore, the process of features construction needs to depend on activities list to calls required data. Example of activities that can be constructed are viewing video, viewing forum or done quiz.

Table 1. Example of feature engineering process using events file.

Anon_screen_name	Event_type	Time
b90558eb	/courses_x/Introduction.pdf	2015-12-01 00:49:20
9c88c5dc	load_video	2015-12-03 18:12:55
9c88c5dc	pause_video	2015-12-03 18:17:00
b681782b	/courses_x/courseware	2015-12-01 00:07:30
b681782b	page_close	2015-12-01 00:07:34
b681782b	/course_x/discussion/forum	2015-12-01 00:07:56

4.1.2. Time index

For time index, instead of the simply summarizing the date or time instance into week or month from a file, time index for MOOC dataset in this study is based on duration call from several files. This is because, each activity has different time frame. To get the duration of learning, all activities with the time frame need to be combined. Other than that, type of resources is different among courses which require a list or dictionary to be developed for references. Therefore, the process requires more space and detail investigation on available resources prior analysis.

4.2. Comparison of method

From feature reduction technique, the highest rank features selected among all the features constructed for *x_course* are shown in table 2. All the features selected are generated features. The same result applied for *y_course* which the highest rank consists of generated features which most of the features with depth 3. The study concludes that the feature engineering process which has been implemented accomplished the objective to produce generated features especially *no_events_in_week* feature that can be used for MOOC prediction.

For prediction, the result of accuracy is used to show if the techniques are able to predict the label correctly using the various top classification techniques include decision tree (J48 and DecisionStump), OneR and k-nearest neighbors (lazy) technique. From the result, all techniques show inconsistent result of CCI using combination of based features and generated features. However, the result is consistent using k-nearest neighbors. Therefore, k-nearest neighbors is used for comparing the result of DFS method using based features and generated features for its stability. In terms of RMSE, *x_course* reported 0.5 error on average. The result of both strategies shows almost similar result which both are able to make prediction at around 80% and 90% using one of the selected techniques. *y_course* also reported around 0.5 error on average for RMSE. Meanwhile, using k-nearest neighbors technique, figure 2, shows prediction result using based features, generated features and combination of features for both *x_course* and *y_course*. For both courses, combination of features shows better result of accuracy with increase of 15 percent for *x_course* and 5 percent for *y_course.* Meanwhile, using only generated features, *x_course* improved by 5 percent and *y_course* improved at least one percent. Overall, the generated features can improve the prediction accuracy percentage by 6 percent on average. The study concluded that by using k-nearest neighbors technique, the generated features can produce better accuracy than based features. Also, the result from both datasets reflects the potential and competency of generated features to achieve almost the same quality with based features which is known to be good features as the features derived directly from data point.

Table 2. Generated features ranked by PCA

x_course	**y_course**
no_events_in_week - view_forum sum	(done_quiz - no_events) / (no_events_in_week - view_video_sum)
view_video / login_freq	(done_quiz - no_events_in_week) / no_events_in_week sum
view_video / (login_freq -view_forum sum)	(no_events_in_weeks - view_forum) / (no_events_in_week - view_video_sum)
(done_quiz - view_video) / (done_quiz - login_freq sum)	no_events_in_week / (no_events_in_week - view_video_sum)
(view_forum - view_video) / (login_freq - view_forum sum)	(no_events_in_week - view_forum) / no_events_in_week sum

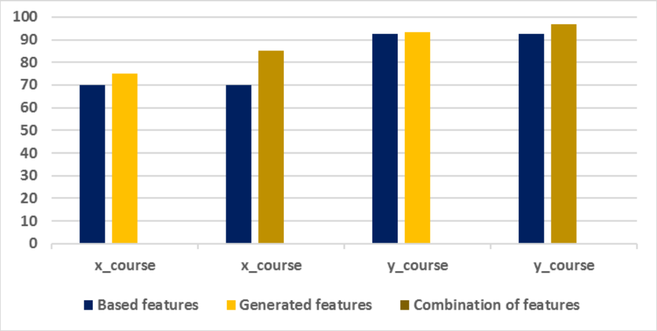


Figure 2. Comparison of accuracy between based features with learned and combined features

5. Conclusion

This paper has presented the process of feature engineering for predicting MOOC student's performance which adopting deep feature synthesis (DFS) method. For feature engineering, this study contributes on the part of construction of based features and time index for weekly feature which is also being used to construct generated features. For MOOC, this study discussed the utilization of generated features towards supporting automated feature engineering. The potential of generated features is shown where all the best features selected by principle component analysis (PCA) are the generated features. As the automate data pre-processing method is designed to reduce workloads, we aim to test the applicability of the method for larger dataset in the future. Lastly, the concern is about the generated features which are uninterpretable and thus may limit academician to relate and understand the student's factors with prediction implemented. However, the effort of this work contributes more on investigating a method to support automated analysis process.

Acknowledgment

The authors would like to express their deepest gratitude to RMC, UTM and Ministry of Education, Malaysia for the support in R&D and Soft Computing Research Group for the inspiration in making this study a success. Also, CAROL, Stanford University for their approval in accessing the data. This paper is supported by GUP Tier 1, vot no. 17H75 and vot no.16H65, UTM-TDR vot no. 06G23.

References

[1] Gaudioso, E and Talavera, L 2006 Data mining to support tutoring in virtual learning communities: Experiences and challenges. *Data Mining in E-Learning (Advances in Management Information),* (U.K: WIT Press) **4** 207-25.

[2] Li, X., Xie, L., and Wang, H. 2016 Grade prediction in MOOCs. *IEEE Intl Conf. on CSE and IEEE Intl Conf. on EUC and 15th DCABES* 386-92.

[3] Li, L. Y., and Tsai, C. C 2017 Accessing online learning material: Quantitative behavior patterns and their effects on motivation and learning performance. *Comput Educ* 114 286-97.

[4] Alario-Hoyos, C., Muñoz-Merino, P. J., Pérez-Sanagustín, M., Delgado Kloos, C., and Parada G, H. A 2016 Who are the top contributors in a MOOC? Relating participants' performance and contributions. *J Comput Assist Lear* **32** 3 232-43.

[5] Kizilcec, R. F., Pérez-Sanagustín, M., and Maldonado, J. J 2016 Recommending self-regulated learning strategies does not improve performance in a MOOC. *Proc. of the 3rd ACM Conference on Learning@ Scale ACM* 101-4.

[6] Qiu, L., Liu, Y., and Liu, Y 2018 An integrated framework with feature selection for dropout prediction in Massive Open Online Courses. *IEEE* 71474-84.

[7] Katz, G., Shin, E. C. R., and Song, D 2016 Explorekit: Automatic feature generation and selection. *2016 IEEE 16th ICDM.* 979-84.

[8] Nargesian, F., Samulowitz, H., Khurana, U., Khalil, E. B., and Turaga, D. S 2017. Generated feature engineering for classification. *Proc. of IJCAI2017* 2529-35.

[9] Zheng, A., and Casari, A 2018 *Feature engineering for machine learning: principles and techniques for data scientists* (United States of America:O'Reilly Media, Inc).

[10] Li, H., Lynch, C. F., and Barnes, T 2018 Early prediction of course grades: models and feature selection. *arXiv preprint arXiv:1812.00843*.

[11] Le, C. V., Pardos, Z. A., Meyer, S. D., and Thorp, R. 2018. Communication at scale in a mooc using predictive engagement analytics. *Proc. of AIED2018, Springer, Cham* 239-52.

[12] Levner, I., Bulitko, V., Li, L., Lee, G., and Greiner, R 2003 Automated feature extraction for object recognition. *Proc. of IVCNZ2003* 309-13.

[13] Cao, D., Lan, A. S, Chen, W., Brinton, C. G, and Chiang, M 2018 Learner behavioral feature refinement and augmentation using GANs. *Proc. of AIED2018* 41-46.

[14] Kanter, J. M., and Veeramachaneni, K 2015 Deep feature synthesis: Towards automating data science endeavors. *2015 IEEE International Conference on DSAA*. IEEE 1-10.

[15] Wedge, R., Kanter, J. M., Veeramachaneni, K., Rubio, S. M., and Perez, S. I 2018. Solving the false positives problem in fraud prediction using automated feature engineering. *Proc. of ECML PKDD.* Springer, Cham 372-88.

[16] Crues, R., Bosch, N., Perry, M., Angrave, L., Shaik, N., and Bhat, S 2018 Refocusing the lens on engagement in MOOCs. *Proc. of the 5th annual ACM conf. on L@Scale*.

[17] Mohamad, N., Ahmad, N. B., and Sulaiman, S 2017. Data pre-processing: a case study in predicting student's retention in MOOC. *J of Fund and Appl Sci* **9** 4S 598-613.