Analyzing Student Viewing Patterns in Lecture Videos

Carsten Ullrich, Ruimin Shen, Weikai Xie
Department of Computer Science
Shanghai Jiao Tong University
Shanghai, China
Email: ullrich_c@sjtu.edu.cn, rmshen@sjtu.edu.cn, wkxie@sjtu.edu.cn

***Abstract*—A large amount of educational content is available as lecture videos, which record teachers as they proceed through a course. Students watch these videos in different ways. They rewind, skip forward, watch some scenes repeatedly. This work investigates what can be learned by analyzing such viewing patterns. We show how to use machine learning techniques to analyze such data, and present the outcomes of an analysis of data collected from the interactions of 2992 students in 253 courses. The viewing pattern were put into relation to seven different variables, such as the final score of the student and the rating teachers received from students Our analysis shows that some variables, such as the teacher rating, were indeed predictable from the viewing patterns.**

***Keywords*-lecture videos; educational datamining; learning analytics; machine learning;**

I. Introduction

Video has become an important part of education. Here, we investigate what can be learned from the viewers' basic interaction with lecture videos (video recordings of a teacher's presentation). By seeking forward or backward, viewers watch some parts of a video several times, other parts not at all, thus generating a unique viewing pattern. Earlier work in this area has shown, e.g., that the first five minutes of a video are most heavily accessed, or that a peak in access occurs around the exam dates [1]. A large amount of existing work (such as [2], [3]) focuses on technical issues with the goal of modeling workload of the media servers. However, to our knowledge, no prior work analyzes viewing patterns from an educational perspective.

II. Data collection

The data used for analysis was collected at the School of Continuing Education (SOCE) of Shanghai Jiao Tong University. SOCE students are adult learners who study for an associate or bachelor degree. The college implements blended learning, i.e., students can come to classrooms in person to attend the lectures or they watch the lectures online live or as a recording. We collected data over a two month period (May to July 2010) from 2992 students and 253 courses. For each access to a video, whenever the viewer moved to a different place in the video, the data recorded the identifier of the view, the identifier of the video, the video time-stamp before the seeking action and the target of the seeking action. In total, we recorded 282.502 seeking actions. We decided to focus on seeking since it is the most basic navigation functionality supported in every media player. The data collection period covered a period in which no exams occurred (the first month) and an exam period (the second month) during which the exams for each course were held (on different dates).

For the analysis, we converted the viewing patterns into one vector format that captures the total amounts a segment of a video has been viewed by a student (total pattern, tp), and into another format that captures the sequence in which the student viewed the video segments (sequence pattern, sp). Analyzing these two formats allows to understand if more complex data (sp) yields better results than more simple data (tp).

The total pattern represent how often a student watched a particular segment of a given course. For each course, and for each student who watched a video lecture of that course, we create an input vector $(v_1, \ldots, v_n)$. Each v_i corresponds to a five minute segment of the complete course video, which is the concatenation of the videos of all lectures of the course ($n = 522$ is the length of the longest lecture video; in case of a shorter lecture the slots were filled with 0s). The value of each v_i is the number of times the student watched at least 30 seconds of that particular segment. For instance $(2, 1, 1, 1, 1, \ldots)$ represents that the student watched the first five minutes two times, and the subsequent segments once.

The sequence pattern represents the sequence in which a student watched the segments of a given course. For each course, and for each student who watched a video lecture of that course, we create an input vector $(w_1, \ldots, w_m)$. Each w_i corresponds to the index of the video segment which the student watched for the longest time in the $5i$ minute of his complete viewing session (which is the concatenation of all viewings of a student of the course).

The vectors represent viewing patterns of students in individual courses. Since we were interested in analyzing the date from a large number of perspectives, we decided to use all available data somehow connected to the courses. We explored rather obvious relationships, such as whether the viewing pattern would allow to predict the final score, but also not so obvious ones, such as whether from the viewing pattern we can guess the teacher who taught the course. Specifically, we used the following functions:

$studentScore$, the final score of the student in this course (on a scale from A to F); $studentPass$ (true/false), whether a student passed the course; $teacherRating$, the evaluation of the teacher of the course received from the students (for each teacher there is one rating on a scale from 1 to 3 (best rating), which is the average of all ratings received from students); $courseId$, the course identifier; $courseCategory$, the course category (e.g., language learning or computer science); $studentId$, the student identifier; and $teacherId$, the teacher identifier.

Finally, we removed those combinations of input vectors and output functions with an amount of vectors insufficient for training the classifier.

III. Results

We analyzed a total of 14 data sets (the two sets of input vectors, with the 7 output functions), using the Scikit-learn library [4]. We employed a k-nearest neighbors classifier with a cross validation of 100 iterations (in each iteration a randomly selected 70% of the input vectors served as the training set, and the remaining 30% for the validation). For each iteration, we calculated the precision, recall and f-Measure. The final score of each data set was the arithmetic mean of the scores the cross validations iterations.

Table I contains each data set with its the scores, ordered by decreasing value of the f-Measure. The name of each data set specifies the input vectors (tp and sp), and the output (the function as explained in the previous section).

The table also contains the probability of guessing the correct output by randomly selecting one of the possible output values (chance). Note that the chance value displayed in the table assumes that all output values occur equally often (except for $studentPass$). This is actually not the case. The problem is best explained with $studentPass$. There, instead of half of the students passing the course and half of the students failing to pass, in the real data 2119 students pass, while 1493 do not pass. This means that the "real" chance as reflected in the data that the guess that a randomly selected student is passes correct is .587, instead of .5. However, taking this into account makes the analysis very complicated if more than two output values occur. In the subsequent analysis, we will briefly discuss this issue for each data set where it is necessary.

The highest precision and f-Measure is achieved by $tp\_teacherId$. Its average precision of .7 is four times as good as chance (.17) (the output values are almost evenly distributed). $sp\_teacherId$ performs much worse with a precision of .35 (which is still significantly better than chance). Interestingly, for most output functions the classifier for data sets tp perform better than sp.

Next in the table come $studentPass$, with sp and tp almost similar. Their precision (tp:.61, sp: .50) and recall values (tp: .65, sp: .66) are high, but since there are only two categories, chance is above .5, too.

Data set	Chance	Precision	Recall	f-Measure
$tp\_teacherId$	0.17	0.7	0.65	0.65
$sp\_studentPass$	0.587	0.59	0.66	0.63
$tp\_courseId$	0.09	0.67	0.63	0.63
$tp\_studentPass$	0.587	0.61	0.65	0.63
$tp\_teacherRating$	0.33	0.61	0.6	0.6
$tp\_courseCategory$	0.2	0.59	0.59	0.59
$sp\_courseCategory$	0.2	0.45	0.48	0.46
$sp\_teacherRating$	0.33	0.45	0.44	0.44
$sp\_courseId$	0.09	0.35	0.36	0.34
$sp\_teacherId$	0.17	0.35	0.33	0.32
$tp\_studentScore$	0.2	0.28	0.29	0.28
$sp\_studentScore$	0.2	0.26	0.27	0.26
$sp\_studentId$	0.003	0	0	0
$tp\_studentId$	0.003	0	0	0

Table I
SCORES OF tp AND sp (ORDERED BY F-MEASURE

Like $teacherId$, the two results for $courseId$ are far apart, too. Again, tp performs rather well, with a precision of .67 and a recall of .63, compared to 0.09 for chance (the output values are similarly distributed). For sp, the precision of .35 and the recall of .36, is still much better than chance.

Both classifiers of the data sets for $teacherRating$ performed better than chance (.33). For tp the predictions are quite good, with a precision of .61, and a recall of .6. The performance of sp with .45 and .44, respectively, is less good, but still better than chance.

The values for $courseCategory$ are difficult to interpret due to the skewed distribution of the output values. While there are 5 categories, the largest category occurs 2400 times, and the smallest only 165 times. Yet, a detailed look at the classification results of one of the cross-validation samples of tp shows that the precision for the individual outputs, even the small ones, is higher than chance. For sp, precision and recall is slightly lower (.45, .48).

The $studentScore$ classifiers perform badly. For tp and sp, the performance is only slightly better than chance (.2).

In the case of $studentId$, for no student the required minimum amount of data was available. We generated classifiers anyway to see how they would behave. It turned out that the precision and recall figured close to 0.

IV. Discussion

Given that this was the first analysis of viewing patterns using ML methods, we did not know what to expect. The data itself captured only a very small part of the overall learning experience, which also encompassed visiting lectures, doing homework, interacting with the school LMS, etc. Intuitively we had expected that, if at all, the viewing pattern might predict student scores (i.e., students who watched regularly would get a high final score).

When looking at Table I, first of all, it is noteworthy that some prediction is possible at all, given the limited nature of the captured data. Also, it is impressive that some predictions are quite good, especially when compared to chance.

The highest predication rate is for predicting the individual teacher. On a first glance, this might be interpreted as if the teacher, e.g., her individual style, influences how students watch a lecture, an assumption that sounds persuasive. Yet, the data also reveals that single courses are predicted with high accuracy, which might be an indication that the content influences the viewing pattern. A look at the data reveals that all but three teachers teach only a single course. Thus, it is actually to be expected that results for course and teacher prediction are close together. The fact that they are different at all is due to the fact that for more than half of all courses we were unable to determine the lecturer because of missing data. The analysis also shows that the teacher rating given by students can be predicted from the viewing pattern. This indicates that students watch courses by teachers they like or dislike differently. Also, the data indicates that courses from different faculties are watched differently. The data contained courses from computer science, foreign languages, economics, engineering, and art. Despite this categories not being uniformly represented, their prediction was better than chance. The students' performance could not be predicted from the data. Albeit the classifier for determining whether a student passes a course or not figures high in the table, it is barely above chance. The precise score cannot be accurately predicted at all.

The data shows that for most classifiers tp performs better than sp (In the case of the outlier $sp\_studentPass$, the performance is barely above chance). This means that the simpler dataset, which represent just how often students viewed a particular segment yields better results than the overall sequence in which they navigated through a course.

The primary goal of this work was to explore what is possible which such a limited data set, and we have shown that some predictions were possible, and thus that the matter is worth investigating. The question now becomes what useful information can be predicted from viewing patterns? In general, being able to predict which teacher is teaching a specific course is not particularly interesting, since this information is typically available anyway. From our data, the main useful information is the teacher rating, as this information might help a teacher to adjust her teaching way to improve the expected rating. To be helpful though, this information has to be available before the end of a semester. We are now running a data analysis with half of the data and preliminary results show that even in this case, the teacher rating can be predicted. In any case, in a realistic application such data would be one part of an overall data analysis, which includes student behavior in the LMS, performance, etc. We have shown that viewing patterns can be one valuable piece of educational data mining.

V. Conclusion and Further Work

This work investigated what can be learned by analyzing how students watch video recordings of lectures. While earlier work focused on analyzing the patterns per se, that is, how students navigate, our analysis focused on data from the educational setting, such a student performance, the course category, the rating of the teacher who taught the course, etc. This is the first analysis of this kind.

The analysis shows that predictions are possible, and in consequence that it is worthwhile to collect and analyze such data. Considering how popular video has become, it should be further investigated to what extent the results found in our work are generalizable.

The data presented here had several shortcomings. First, The data set covers a two month period, near the end of the semester. It should be investigated whether the results differ if viewing data of a complete semester is analyzed. Also, the data is taken from one single semester, and thus does not allow to answer whether similar prediction performances will arise if data from a different semester is analyzed. Finally, this data was collected in Chinese adult distance education. Future work has to investigate whether similar predictions are possible in different settings.

In a first step, however, we will perform an additional analysis of the data, in which classifiers are run on subsets of the data, e.g., those from specific courses or course categories. This will improve understanding of how different variables influence the results.

We did not investigate how the results are influenced by parameters of the nearest neighbors method we are using. In our analysis, we used the default settings as provided by the library. Often, tuning parameters can significantly improve classification performance. Similarly, an open question is how results change if a different ML method would be used.

Acknowledgment

We would like to thank Heng Luo for his advice on how to use machine learning techniques to our data. This work was supported in part by the Science and Technology Commission of Shanghai Municipality under Grant No.11511503002.

References

[1] J. A. Brotherton and G. D. Abowd, "Lessons learned from eclass: Assessing automated capture and access in the classroom," *ACM Trans. Comput.-Hum. Interact.*, vol. 11, no. 2, pp. 121–155, 2004.

[2] J. Padhye and J. Kurose, "Continuous-media courseware server: a study of client interactions," *Internet Computing, IEEE*, vol. 3, no. 2, pp. 65–73, 1999.

[3] J. M. Almeida, J. Krueger, D. L. Eager, and M. K. Vernon, "Analysis of educational media server workloads," in *Proceedings of NOSSDAV '01*. New York, NY, USA: ACM, 2001, pp. 21–30.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.