

Video Visualization Profile Analysis in Online Courses

Gonzalo Martínez-Muñoz^{1b}, Miguel Ángel Álvarez-Rodríguez, and Estrella Pulido-Cañabate^{1b}

Abstract—In this article, student video visualization profiles are analyzed with two objectives: 1) to identify difficult sections in videos and 2) to predict student performance based on their video visualization profiles. For identifying critical sections in videos two novel indicators are proposed. The first one is designed to measure the complexity of the concept being described. The second proposal, identifies video sections that are more visually complex. For the first indicator, the average number of forward and backward passes are used. The higher the number of backward (forward) passes over a region, the more challenging (easy) the section is. For identifying sections with complex visuals, the number of pauses is recorded. Finally, the student performance prediction is carried out with the purpose of detecting the alignment between videos and their related questions. The results show that video visualization profiles are a good tool to identify video and question alignment.

Index Terms—Machine learning, massive open online course (MOOC), online learning, video analytics.

I. INTRODUCTION

IN RECENT years, online university-level education has experienced a fast increase in the course offer and number of students. Most universities offer online courses through online educational platforms, such as iversity, edX, Miriadax, etc. Educational platforms offer a myriad number of courses of multiple different fields. The growth of these platforms has further accelerated due to the covid19 pandemic, for instance, edX has almost doubled the amount of courses and students in the period from 2018 to 2020¹ going from 18 million students to 35M and from 2275 courses to 3090. Initially, the courses were offered for free in the format known as massive open online courses (MOOCs). However, the trend of open courses has shifted to a more controlled environment in which part of the course contents and exercises is made available only to students who pay to obtain a certificate. Under these conditions, institutions not only need to assure that students acquire the contents corresponding to the issued certificates, but also to improve the student educational experience and to provide high-quality educational resources. In this context,

Manuscript received 10 July 2023; revised 13 February 2024; accepted 9 April 2024. Date of publication 30 May 2024; date of current version 8 August 2024. This work was supported by the Agencia Estatal de Investigación under Grant PID2019-106827GB-I00. (Corresponding author: Gonzalo Martínez-Muñoz.)

The authors are with the Department of Ingeniería Informática, Universidad Autónoma de Madrid, 28049 Madrid, Spain (e-mail: gonzalo.martinez@uam.es).

Digital Object Identifier 10.1109/TE.2024.3396296

¹<https://www.classcentral.com/report/edx-2020-review/> (accessed 09/12/2022)

machine learning has been a handy tool to provide added value to monitor online education [1].

Automatic monitoring can be performed since these platforms log all the interactions of students with the course. This information is very valuable to monitor students' learning processes, including dropout prediction [2], grade prediction [3], [4], [5], [6], [7], or identification of student learning approaches [6], [8], [9]. In grade prediction, the goal is to predict student performance before the end of the course. The grade prediction can be based on activity-related events [3], on quizzes' results [4], event timestamps [5], event transitions [6], or summary statistics from video watching sessions [7].

Online courses are structured around video lectures as the main educational resource. In this context, video analytics can be used to assess the quality of courses and to analyze the learners' digital footprints in order to understand how students interact with them [10]. Most studies that apply machine learning techniques to analyze student visualizations are based on summary statistics [7], [10], [11], [12]. Student visualization footprints are generally described with attributes such as: number of pauses, number of feed forward, viewing total time, fraction of watched video, etc. Other studies use the complete student visualization sequence in order to identify peaks in videos [13] and as tools to help instructors analyze videos [14], [15], but not as predictive attributes.

In contrast with previous works that are based on analyzing the number of events generated by the students when interacting with videos and/or other elements of the course, in this article, the student video visualization profiles are used. Understanding as *video visualization profile*, the number of times each video second is watched by a student. These profiles are applied, first, to identify critical sections in videos, and, second, to predict the student performance in evaluation quizzes. We propose two indicators to identify critical sections in videos: one more related to the complexity of the concept being described and one related to the visual complexity of the images being shown in videos. In addition, the video watching profiles of students are also used to predict their performance in related questionnaires with the objective of detecting whether the videos and their related questions are aligned.

The remainder of this article is organized as follows. Section II describes other works related to the present study; Section III describes data and the proposed methodologies; Section IV shows the study results; and in Section V the conclusions of this research are summarized.

II. RELATED WORK

Mirriahi and Vigentini [10] defined video analytics as the collection, measurement, and analysis of learners' digital footprints when accessing videos for the purposes of understanding how learners use them and engage with them. They discuss different data mining approaches that can be applied to video related data and propose some questions that remain open in relation to the effectiveness of videos for learning. Two of the future directions given in this article in the context of video analytics more related to the present study are: 1) identify interventions that can be applied to inform instructors of possible changes to the video contents and 2) help in the instructional design to improve learning and student experience.

Kim et al. [16] proposed an enhance video viewer based on previous interaction data from watchers. The enhanced viewer provides a 2-D timeline that shows the current student the interaction events of previous students for each video time point and highlights those regions with the highest number of events. These regions can help students to identify the difficult sections in videos. Events are recorded collectively without taking into consideration the type of event.

Giannakos et al. [17] proposed a video visualization tool that was used to extract video events from videos. They tested the system with eleven freshmen in a course that included video lecturers to assist students during a 7-week period. From these events, they extracted a time-series plot with the viewers' watching activity for each video as repeated views. An important finding is that they observed a correlation between the activity peak for each video and the difficulty of the questions related to each of the video segments. In addition, the course included questionnaires related to the ease of use of the video tool showing that the tool was intuitive and useful.

In [18], they proposed an interesting procedure for detecting important segments in videos. The experiments were performed using a specifically developed video viewer that allowed for click event logging. The viewer has a pause/play button, a button for forwarding 30 s the video and a button for replaying again the last 30 s. The system was tested on Youtube videos that are "visually unstructured." They show that the timestamps of replay button events are very correlated with the important segments of the video, which occur right after these important sections. Observing the peaks of replays right after the important sections makes sense, as the viewer generally seeks back after an important section to watch it again. In this work, we extend this idea using more events to elaborate time series that can distinguish between different types of contents in videos.

In [12], video visualization patterns are analyzed. Each video watching session is described by summarizing features like the number of pauses, the average speed, the replayed length, the duration of pauses, the proportion of skipped content, etc. They found that the watching visualization patterns are correlated with the perceived difficulty of the video: more difficult videos were watched with more frequent pauses and

replays. They also noticed that more than half of the pauses occurred in code snippets.

Several visualizing tools have been proposed [14], [15]. In [14], they present a visualization tool that allow users to see: differences in the clickstream of different videos, evolution of the different click actions (pause, play, seek, etc.) along with the videos and statistical information about learners. In [15] they propose a graphical tool to help instructors extract valuable conclusions from video clickstream data, which includes actions, such as play, pause, seek, etc. The tool visualizes the timeline of clickstream data as a smoothed histogram and highlights the peaks, which are detected automatically. In addition, the tool shows the statistics for each detected peak and region identification of those learners who generated the peak. It also shows existing correlations between different learner groups.

Shridharan et al. [11] proposed a methodology for predicting MOOC learner behavior when watching videos. They define nine features to summarize the learner watching behavior that include: watched video percentage, total time on video divided by video length, amount of the video played by learner, with repetitions, divided by its total playback time, time spent in pause, number of pauses, number of backwards, number of forwards, etc. In order to predict future video watching behavior they fit a linear regressor model to each of the above features using the remaining features as input variables. They also applied collaborative filtering. The relation among variables can be used by instructors to rethink the videos. For instance, if a learner is predicted to have a high number of feed forwards, then abridged versions of the videos could be shown. Another interesting observation is that they showed that individual students tend to have similar watching patterns irrespective of the contents being watched. The study was carried out on a dataset containing one million clickstream events generated by 3976 learners in a Coursera MOOC with 92 videos.

In [7], they use video watching summarizing features, such as fraction watched, number of pauses, etc. to predict the student performance in the weekly quiz. The trained model manage to obtain fairly good results: 82%–93% as pass/fail accuracy. In this article, they considered a deep LSTM model in which a week of data was considered as a time step for the system.

Kim et al. [13] presented the results of analyzing user interaction data from 862 videos in four MOOCs on the edX platform. The goal of this analysis is to compare video dropout and peaks in viewership among tutorials, lectures and rewatching sessions. For the dropout they compare the number of unique viewing sessions for the first second of the video with respect to any other second. They found that shorter videos have lower-in-video dropout rates. In addition, they found that rewatching sessions had higher dropout rates than first time viewing sessions as students tend to review only specific sections of videos when rewatching them. In a second experiment, more related to the current analysis, two sequences are analyzed to locate activity peaks in the video timeline: 1) rewatching sessions and 2) play events. For rewatching

sessions the total number of times each second of the videos is watched in rewatching sessions by all students is aggregated. The play event sequence is computed by counting the number of play events for each video second. They show that peaks can be explained by five different activity patterns generally related to visual transitions in videos, including return to missing content, following a tutorial step or replaying a description. Finally, they provide some guidelines for the design of better-video learning experiences.

III. DETECTION OF IMPORTANT AND DIFFICULT SECTIONS IN VIDEOS

In order to identify important and difficult sections in videos, in this article we propose the use of different time based statistics. These statistics include for each second of the video: the number of unique visualizations, different event counts normalized by the number of unique students that watched the given video second, number of views per student, etc. Normalizing video sequences by unique viewers of each second has the advantage that the height of the peaks could be compared between different videos, in contrast to the use of the total number of counts as it is generally done [13]. The number of views per student of each portion of a video is a good indicator of the importance of the contents as an important section of a video will be visualized more often, for instance when students are preparing the exam. However, the use of more types of events can help identifying not only difficult sections but also the type of complexity the sections portray.

In order to ascertain the difficulty of the different sections of videos, we propose two novel indicators based on the number and type of events each visualization generates. The higher the complexity of a given section of a video, the higher the number of *pause* and *seek backward* events each visualization of that section will generate. On the other hand, easier (or uninteresting) sections will generate more *seek forward* events. Furthermore, the complexity of a video segment can be split into conceptual complexity and static content complexity (or visual complexity). These ideas have been indirectly shown by other authors as we point out hereafter [13]. We understand the conceptual complexity as a measure of the difficulty of the descriptions either because the video tackles abstract concepts or simply because the ideas are not well described. This can be caused by nonvisual explanations [12], [13]. By static content complexity (or visual complexity), we understand static cumbersome figures, texts or code that require more time from the viewer to analyze than the time provided by the normal flow of the video. In [12], this effect was described for code snippets. Also in [13], they identified visualization peaks caused by students that “return to visual contents that disappear shortly after.”

More specifically, we propose to build three sequences of events for each second of the videos as indicators of the importance, the content complexity and the visual complexity. The general importance of each section of a video can be estimated by using the average number of visualization as in previous studies [13], although here we propose to use the normalized sequence per unique student in each second.

In this way, the peak height is not affected by the number of students visualizing the videos. Note, that the number of unique students visualizing each second of a video is not a constant value as it changes from second to second.

For estimating the content complexity of each section, we propose to count the number of times each second is passed with backward seeks minus the number of times it is passed with forward seeks, per unique student. For this, the start and end timestamps of the backward and forward seeks are recorded. Then, if the starting time is greater than the ending time (backward seek), +1 is added to all seconds between start and end. If the end time is greater than the start time (forward seek) then +1 is subtracted to all seconds between start and end. This is a good indicator if a section of the video contains a complex description since students usually do not stop in the middle of the description but prefer to hear it completely and, once it is finished, they seek backwards to hear it again. In contrast, easy sections are passed with seek forward more often and do not generate backward events.

For the static content complexity, we propose to use the number of pause and stop events occurring at each second of the video per student. This sequence is different to the sequence of play events (even if generally a pause event is followed by a play event). This is so because our proposal distinguishes between play events occurring after seeks and after pauses. The pause is more suited to identify when the video is showing contents with a high-visual complexity in which the normal flow of the video is too fast.

IV. EXPERIMENTS

A. Dataset Description

The data analyzed in this article comes from the MOOC called “Jugando con Android—Aprende a programar tu primera App” (in English “Playing with Android—learn how to build your first App”). This MOOC was offered by the Universidad Autónoma de Madrid on the edX platform.² The courses offered through edX have different formats: self-paced, open, certified, etc. In the *self-paced* format all activities in the course are open and students can navigate throughout the course from the day they enroll in it. In this format, usually there is few or no assistance from teachers. In the programmed format, the course opens on a specific date and videos and activities are published on a weekly basis. In this format, the students are somewhat more guided through the contents of the course than in the self paced format, as they can only access the contents of the current and previous weeks. In addition, in the programmed format, there are usually teaching assistants who help students with contents and activities. The data analyzed in this work comes from a programmed course. The course is organized in seven weeks: the first six weeks introduce the contents of the course and the last week is for the final exam. For each of the first six weeks a series of videos, short questions and a programming activity are proposed to students. The short questions are intended to check whether the video contents have been understood.

²<https://www.edx.org/school/uamx>

TABLE I
TOTAL NUMBER OF THE DIFFERENT VIDEO EVENTS IN THE COURSE

Video event	# of events
Load video	160129
Play video	660024
Pause	241514
Seek	175137
Speed change	3584
Stop video	48682

The weekly programming activity proposes to students a more complex problem. This problem is ought to be solved by using the programming techniques described during the week in the videos and documents. After solving the programming assignment, students take a brief test to verify if he/she has implemented correctly the activity solution. In addition, there is a self graded programming project that runs throughout the course. In this study, we will not consider the performance of the students on this project. The course has a final test to evaluate students. In summary, the course has a total of 30 videos, 63 short questions (≈ 2 per video), 58 questions to test the programming activities and 61 questions for the final exam.

The students trace throughout the course is recorded in a log file that stores all the different events. From these events, we have kept all those related to videos, exercises, activity's tests and final exam questions. All events include time-stamp and user id. The events related to exercises and activity's tests include additional information, such as exercise id, student's answer and if the answer was correct or not. The events related to videos include specific information of the actions carried out by the student on the videos. The actions that can be performed on the videos are: play, seek, pause, stop, and change video speed. These actions include information of the second of the video in which the action was performed. In the analyzed course there are over 1 200 000 video events distributed as shown in Table I. This high number of events is due to the fact that the course had over 8000 enrolled students.

Students with less than 150 events of any type were removed to avoid noisy patterns. This threshold allows us to keep 25% of the original students (2042 students out of over 8000 students). However, since the removed students performed less than 150 events, the total number of removed events is only 35%, that is, we are keeping 65% of the original information, which is mostly related to students actually taking the course.

B. Video Analysis

In this section we propose a novel and comprehensive video analysis with two objectives in mind. First, to detect which sections of the videos are found to be more problematic and, second, to link the student video visualization profile with his/her performance in the course exercises.

The way students visualize videos can differ among students. Some students may visualize videos from start to finish in one go. However, in general, students tend to stop the video, to go backward to watch again a given portion, or to go forward to accelerate an uninteresting section. In order

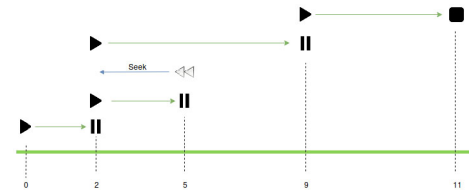


Fig. 1. Example of a video visualization sequence.

to simplify the analysis of the visualization of the different videos, we have converted the back and forth events into *visualization segments* that will represent the visualization of a portion of a video by a student. Each visualization segment is a portion of a given video watched by a given student without stopping. The portion is defined with a start and end points in the video (in seconds). For instance, in Fig. 1 it is shown an illustrative example of a video visualization with nine events, including play, pause, seek, and stop. After processing those events, we end up with four visualization segments: 1) 0–2; 2) 2–5; 3) 2–9; and 4) 9–11 s. Thus, in this example section 2–5 s is visualized twice and the rest of the video once. After this preprocessing, over 350 000 *visualization segments* are obtained. These sequences are sorted by user and video, which allows us to analyze efficiently the information of how students visualize videos.

This information is shown in Fig. 2 for all videos of the course. Videos are organized by week to avoid cluttering in the plots. Note that not all weeks had the same number of videos (it varies from 8 videos on the second week to 3 on the third week). The plots show, for each of the 30 videos, the total number of visualizations (vertical axis) for every time fraction of the video (x -axis). The x -axis is plotted as the time fraction with respect to the total duration of each video. In this way all videos are shown in the same horizontal scale independently of their duration. The curves have been softened using a moving average of 5 s in order to remove small distortions.

From these plots, several general patterns in the visualization profile of most videos can be observed. First, the number of visualizations clearly diminishes as the course advances. This is a common trend in this type of courses as student tend to dropout as the course advances [19]. A second general observation is that the number of visualizations for most videos is lower at the beginning and at the end of each video. This can be explained because the videos in this course include a brief introduction and summary of the video presented by the teacher. Hence, students tend to skip those parts at least when they revisualize the videos. A final important general trend is that the visualizations do not have a tendency to diminish as the video advances as shown in other studies [13]. This indicates that the length of the videos is adequate to their contents. In addition, from these plots, it can be observed that there are videos with rather high peaks. These peaks can be due to several reasons and not necessarily to the difficulty of a section. For instance, the narrow peak observed in video 1—that explains the installation of the development environment—around time fraction \approx



Fig. 2. Total number of visualizations of every time fraction for each video organized by week.

0.25 corresponds to a moment in which a long URL is given to download the Android SDK. Hence, it makes sense that students go backward and pause the video in order to be able to copy the URL. This generates more passes on that small section. On the other hand, the higher number of views observed for video 17—working with event listeners programmatically—on its second half correspond to a more complex section in which different program code snippets are shown. The four peaks in that section (at approximately 0.5, 0.6, 0.75, and 0.85) correspond to the description of the different parts of the implementation, that appear sequentially on the screen as they are described. These peaks at the different transition points were first described by [13].

However, the total visualization count provides a partial understanding of the difficulty and of how content is being visualized. In order to gain more insight into how students interact with contents, we have analyzed the different events fired by students while watching the videos. Specifically, we have analyzed, in a per unique student and per second basis: the average number of visualizations, the number of generated pause and stop events and the counts of forward/backward passes. These sequences are described in more detail in the previous section. In Fig. 3, this information is shown for four representative videos with respect to the real video time. The number of visualizations per student are computed for each video second as the number of times each second has been

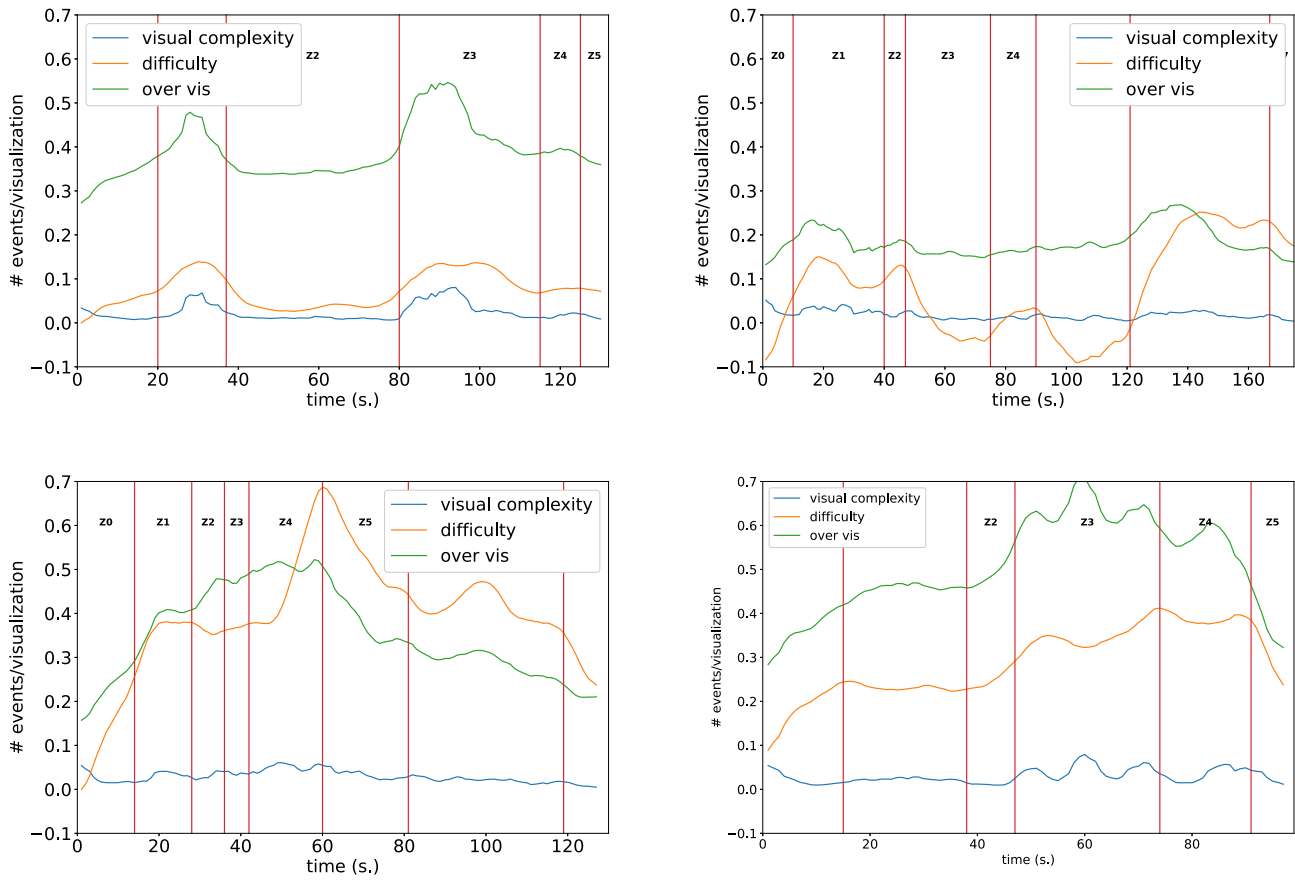


Fig. 3. Video sequences for video 2 (top-left), 9 (top-right), 10 (bottom-left) and 17 (bottom-right).

visualized divided the number of students that have visualized that second. This is shown in the plots as the number of visualizations per student minus 1, as values below 1 cannot be achieved by definition. In detail each plot shows the following.

- 1) The number of visualizations per student over one. This is an indicator of the difficulty of each video section. This is shown with a green solid line as *over visualization* (*over vis* in the legend).
- 2) The number of pauses and stops per second and per student, which is a proxy for the visual complexity or static complexity of each part of the video, is shown as *visual complexity* in the plots (blue solid line).
- 3) The number of backward minus the number of forward passes occurring in each second of the videos per student. This is a proxy of the difficulty of the different parts of the videos. It is shown with an orange solid line and with label *difficulty*.

All plots are shown in the same vertical scale. In addition, in these plots we have identified the different sections of each video, which have been named as Z0, Z1, ..., etc. and have been separated with vertical lines. The description of each section is given in Table II.

From these plots we can identify problematic sections of the videos. In video #2 (top left plot on Fig. 3), two problematic sections can be identified (Z1 and Z3). These sections show

higher proportion of seek backward events and pauses. This is because instructions are given to open the SDK manager (in Z1) and to install several packages (in Z3). If a student wants to follow the instructions to update his/her IDE, he/she would need to stop and go back in order to keep up with the video pace. This could be a bit annoying and indicates that a slower pace on those sections may help. Video #9 shows a different behavior. The pause events are almost 0 for the whole duration of the video and it has a low value of over-visualizations. In addition, sections Z3 and Z5 show a higher number of forward events than backward events meaning that those sections are uninteresting. In fact, those sections are a bit repetitive as the same concept is repeated 4 times for the different values (i.e., top, bottom, right and left) of margin and padding (see Table II). In video #10 (bottom left plot), again the number of pauses is almost 0 indicating that the visual complexity is low. However, the difficulty of the video is rather high, specially for the central part (segments Z4 and Z5) that shows an example of use. The concepts shown in this video (gravity and layout_gravity) is a feature on Android that is not present in most languages. Hence, students are not familiar with it. Finally, video #17 (bottom right plot) shows difficult concepts related to event listeners (not as difficult as video #10). The event listeners explained in this video are one of the basic elements for implementing interaction in Android apps, which explains the high number of visualizations. In

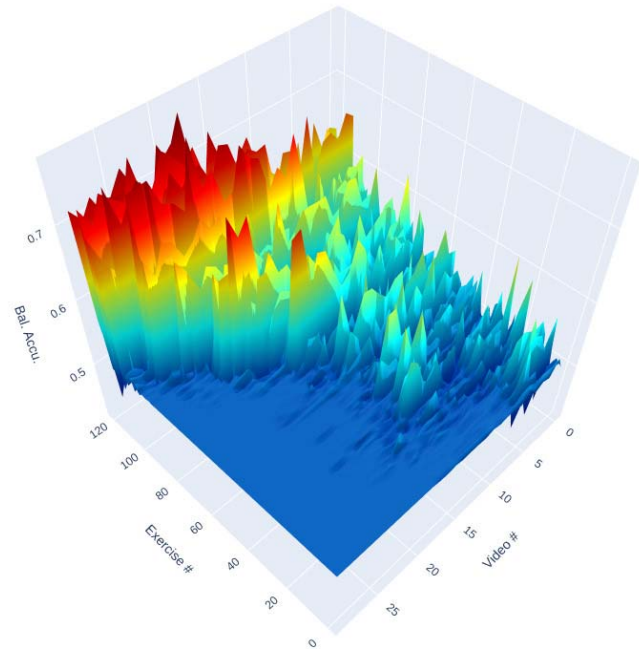
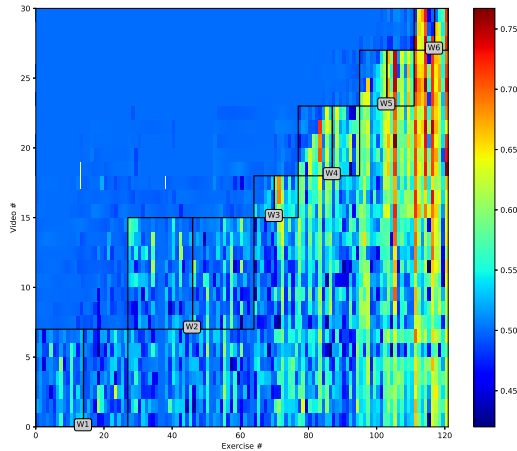


Fig. 4. Video problems relation shown in 2-D (left) and 3-D (right).

TABLE II
VIDEO SEGMENT DESCRIPTIONS

Video # 2: SDK manager	
Segment (seconds)	Description
Z0 (0-20s)	Introduction
Z1 (20-34s)	Instructions to open SDK manager
Z2 (35-80s)	Description of SDK manager
Z3 (81-115s)	Instructions to update packages with SDK manager
Z4 (116-125s)	Installation result
Z5 (126-132s)	Wrap-up
Video # 9: Margins and padding	
Segment (seconds)	Description
Z0 (0-10s)	Introduction
Z1 (11-39s)	Description of margin and padding
Z2 (40-46s)	Description of padding
Z3 (47-75s)	Description of padding top, bottom, left, right
Z4 (76-90s)	Description of margin
Z5 (91-121s)	Description of margin top, bottom, left, right
Z6 (122-167s)	Example
Z7 (168-180s)	Wrap-up
Video # 10: Alignment with gravity	
Segment (seconds)	Description
Z0 (0-14s)	Introduction
Z1 (15-28s)	Description of gravity
Z2 (28-36s)	Enumeration of values of gravity
Z3 (36-41s)	Description of layout_gravity
Z4 (42-60s)	Code example of gravity
Z5 (61-80s)	Code example of layout_gravity
Z6 (81-118s)	More complex example of both concepts
Z7 (119-130s)	Wrap-up
Video # 17: Event listeners	
Segment (seconds)	Description
Z0 (0-14s)	Introduction
Z1 (15-37s)	Description of event listeners
Z2 (38-47s)	Introduction to the implementation of event listeners
Z3 (48-73s)	Description of implementation of listeners with code
Z4 (74-91s)	Alternative implementation using Activity
Z5 (92-99s)	Wrap-up

addition, in segment Z3 an example with code is given and, again, the speed of the description is higher than the time students need to grasp the code. In consequence a higher

number of pause events are shown indicating a high-visual complexity.

C. Video-Problem Predictions

The objective of videos is to transmit concepts. These concepts are subsequently tested to check whether students have understood the ideas portrayed. In order for this process to work, the videos should transmit the content efficiently and the exercises should also be well aligned with the videos. In this section, and in order to get more insight into the efficiency of the videos, we have analyzed their relation to the outcome of the different evaluation items of the course.

For this purpose, machine learning models are used to learn the relation between video visualization patterns and the outcome to the exercises. These models will be trained to predict the correct or incorrect answer of a student to an exercise given the visualization pattern of the student to videos prior to her/his response. For each exercise, only students that responded to that exercise were considered. For the visualization pattern of a video by a student, the number of views of each second of the video could be used. However, granularity of single-second views may lead to over-fitting as they are noisy features. In order to confer the features more meaning and stability, video segments of 10 s in length are considered and the average number of views during each segment are used as attributes to predict the exercises outcome. These segments are computed every 2 s of the video, that is, the first attribute is the mean number of visualization done by the student of segment 0–10 s, the second from 2–12 s, the third from 4–14 s, and so on. This overlap between segments is done in order to mitigate the creation of artificial separations every 10 s.

The objectives of this process are: 1) to be able to predict the outcome of an exercise based on how the video was watched by the student and 2) to use the accuracy in the prediction as a measure of the correlation between visualization patterns and exercises. For this purpose a random forest ensemble was trained using 100 decision trees and the square root of the number of attributes to be selected at random at each split. The sklearn package was used [20]. In addition, the class weight hyper-parameter was set to “balanced” in order to equalize the weights of both classes as, in general, the generated datasets are imbalanced (there are approximately 9 correct answers to 1 incorrect). This equalization forces the model to put the same importance in both classes independently of the number of instances of each one. The Gini criterion is used to train the trees. Random forest was used as it is one of the best classifiers on average for tabular datasets and one of the models that needs less hyper-parameter tuning [21], [22]. In addition, we carried out some exploratory experiments with other models, such as decision trees, that produced poorer results. In this sense, the reported generalization score is balanced accuracy that measures the average recall obtained on each class. The validation process was repeated for stability using 10-fold cross-validation. This means that 90% of the data is used for training and 10% for validation. The process is repeated 10 times leaving for validation a different 10% of the data each time. Finally, for this part, we analyze the relation between the 30 videos of the course with respect to the 122 exercises of type short video questions and weekly activities. In total, 3660 classification problems are analyzed: 30 videos \times 122 exercises.

The results are shown in Fig. 4 using a 2-D and a 3-D plot. Both plots show the average balance accuracy of the random forests in the prediction of the outcome of each problem (axis labeled with “Exercise #”) given the visualization pattern of each video (axis labeled with “Video #”). The value for the balanced accuracy is shown using the “jet” color scheme with dark blue for the lowest values and dark red for the highest-mean balanced accuracies. For the 3-D representation (right plot), the balanced accuracy is also shown in the height of the surface. In addition, in the left plot, exercises and videos belonging to the same week are grouped with two rectangles marked with “W” and the week number. For each week, the left rectangle identifies the exercises directly related to the videos (and that are placed right after each video in the course) and the right rectangle are the exercises of the weekly activities (related to all videos of the week).

Several interesting aspects can be observed in these plots. Regions that relate exercises with videos that come later in the course obtain average balanced accuracies around 0.5, that is, the models are not able to distinguish between correct and incorrect responses. This is so, since we are considering the visualization pattern before the exercise was answered and, in general, no student watches a video belonging to a week later in the course before doing the exercises of previous weeks. This region corresponds to the flat area (see right plot) for low-exercise numbers and high-video numbers. More interestingly, the prediction generalization score improves for

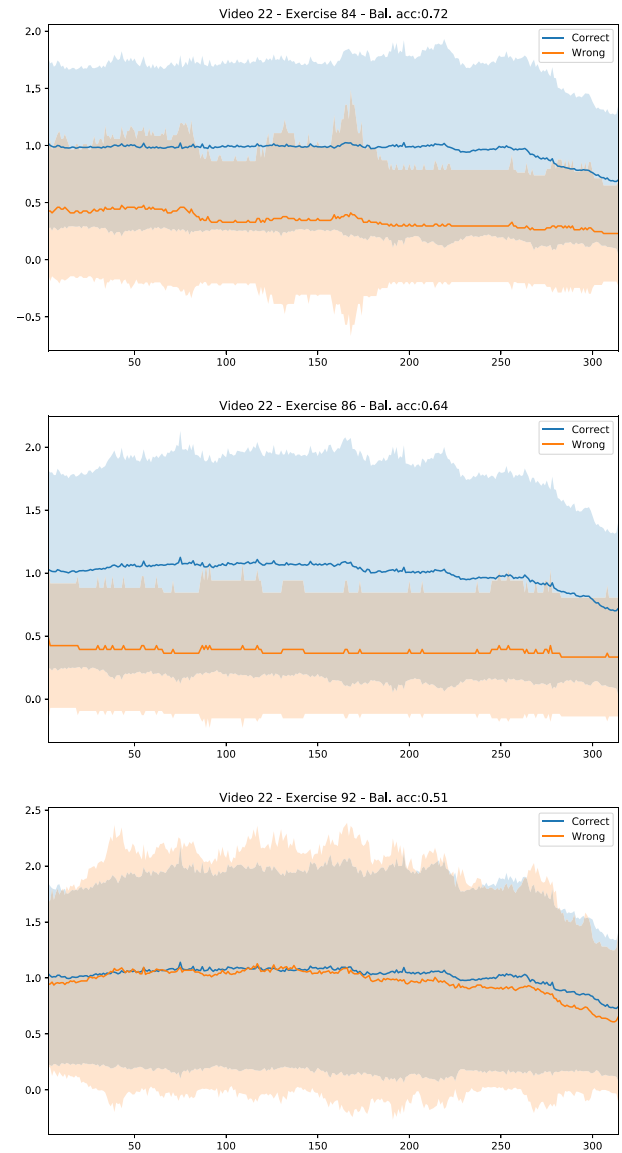


Fig. 5. Video problems relation for video 22.

the areas that correspond to exercises and videos belonging to the same week. In the first two weeks, results are rather low being scarcely above 0.5 on average. One reason for this is that during the first two weeks there are many trivial exercises that were responded correctly by most students: specifically, 20 exercises that were responded correctly by more than 95% of students in the first two weeks. Such imbalance classification tasks are very complicated to solve and results measured in balance accuracy are noisy. In addition to this, there are also noisy responses coming from students not completely interested in the course. As the course advances, much of these students dropout and only those students really committed to the course remain. In fact, from the third week on, the prediction scores improve. This indicates that there is a relationship between the student video visualization pattern and the outcome of the exercises related to that video. In general, the best-prediction performance for any given exercise occurs for one of the related videos.

In a further experiment to show how the visualization of videos relate to exercises, we average visualization pattern of a video for the students who respond correctly and incorrectly to an exercise in addition to the standard deviations. That information is shown in Fig. 5 for three exercises related to video 22. The title of each plot in the figure provides the video and exercise numbers and the mean balanced accuracy value. As it can be seen from these plots, for videos where the balanced accuracy is higher (two top plots) the differences in average number of visualizations of each segment of the video is higher for those who responded correctly to the exercise. For the lower plot, in which our model does not manage to extract any relation (i.e., balanced accuracy is 0.51), we can observe that the average number of visualizations is almost the same for students who answered correctly and incorrectly. Although in all cases the standard deviations are high, which limits greatly the possibility to obtain better results.

V. CONCLUSION

In this article, we propose a novel and comprehensive video analysis with two objectives in mind. On the one hand, to detect video sections that students find more difficult, and, on the other, to relate student video visualization profiles with their course performance. The analyzed videos in this work come from an edX course organized in seven weeks whose contents is organized in videos, short questions and programming activities that check whether the video contents have been understood. Several conclusions are drawn from this analysis. Two novel measures for the identification of difficult sections on videos are proposed. The first measure based in pause events is able to locate complex visual sections on the videos. The second proposed measure, based in backward and forward events, is able to detect where the difficult concepts are portrayed in the videos.

Video analysis is also performed to predict exercise outcomes based on video visualization patterns. This analysis shows that the prediction generalization score improves for exercises and videos belonging to the same week. An additional conclusion is that there is a relationship between the student visualization pattern for a video and the outcome of the exercises related to that video.

REFERENCES

- [1] R. S. Baker and P. S. Inventado, "Educational data mining and learning analytics," in *Learning Analytics*, J. A. Larusson and B. White, Eds. New York, NY, USA: Springer, 2014, pp. 61–75, doi: 10.1007/978-1-4614-3305-7_4.
- [2] W. Feng, J. Tang, and T. X. Liu, "Understanding dropouts in MOOCs," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 517–524.
- [3] M. M. Ashenafi, G. Riccardi, and M. Ronchetti, "Predicting students' final exam scores from their course activities," in *Proc. IEEE Front. Educ. Conf. (FIE)*, 2015, pp. 1–9.
- [4] S. Jiang, A. Williams, K. Schenke, M. Warschauer, and D. O'Dowd, "Predicting MOOC performance with week 1 behavior," in *Proc. Educ. Data Min.*, 2014, pp. 1–3.
- [5] B. Xu and D. Yang, "Motivation classification and grade prediction for MOOCs learners," *Comput. Intell. Neurosci.*, vol. 2016, Jan. 2016, Art. no. 2174613.

- [6] A. Pérez-Lemonche, G. Martínez-Muñoz, and E. Pulido-Cañabate, "Analysing event transitions to discover student roles and predict grades in MOOCs," in *Artificial Neural Networks and Machine Learning (Lecture Notes in Computer Science 10614)*. Cham, Switzerland: 2017, pp. 224–232.
- [7] A. A. Mubarak, H. Cao, and S. A. Ahmed, "Predictive learning analytics using deep learning model in MOOCs' courses videos," *Educ. Inf. Technol.*, vol. 26, pp. 371–392, Jan. 2021.
- [8] R. Ferguson and D. Clow, "Examining engagement: Analysing learner subpopulations in massive open online courses (MOOCs)," in *Proc. 5th Int. Conf. Learn. Anal. Knowl.*, 2015, pp. 51–58.
- [9] S.-F. Tseng, Y.-W. Tsao, L.-C. Yu, C.-L. Chan, and K. R. Lai, "Who will pass? Analyzing learner behaviors in MOOCs," *Res. Pract. Technol. Enhanc. Learn.*, vol. 11, no. 1, p. 8, 2016.
- [10] N. Mirriahi and L. Vigentini, "Analytics of learner video use," in *The Handbook of Learning Analytics*, 1st ed., C. Lang, G. Siemens, A. F. Wise, and D. Gašević, Eds. Beaumont, AB, Canada: Soc. Learn. Anal. Res., 2017, pp. 251–267.
- [11] M. Shridharan, A. Willingham, J. Spencer, T.-Y. Yang, and C. Brinton, "Predictive learning analytics for video-watching behavior in MOOCs," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2018, pp. 1–6.
- [12] N. Li, L. Kidziński, P. Jermann, and P. Dillenbourg, "MOOC video interaction patterns: What do they tell us?" in *Design for Teaching and Learning in a Networked World (Lecture Notes in Computer Science)*. Cham, Switzerland: Springer, 2015, pp. 197–210.
- [13] J. Kim, P. J. Guo, D. T. Seaton, P. Mitros, K. Z. Gajos, and R. C. Miller, "Understanding in-video dropouts and interaction peaks in online lecture videos," in *Proc. 1st ACM Conf. Learn. Scale Conf.*, 2014, pp. 31–40. [Online]. Available: <http://doi.acm.org/10.1145/2556325.2566237>
- [14] C. Shi, S. Fu, Q. Chen, and H. Qu, "VisMOOC: Visualizing video clickstream data from massive open online courses," in *Proc. IEEE Pacific Visualization Symp. (PacificVis)*, 2015, pp. 159–166.
- [15] Q. Chen, Y. Chen, D. Liu, C. Shi, Y. Wu, and H. Qu, "PeakVizor: Visual analytics of peaks in video clickstreams from massive open online courses," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 10, pp. 2315–2330, Oct. 2016.
- [16] J. Kim, P. J. Guo, C. J. Cai, S.-W. D. Li, K. Z. Gajos, and R. C. Miller, "Data-driven interaction techniques for improving navigation of educational videos," in *Proc. 27th Annu. ACM Symp. User Interface Softw. Technol.*, 2014, pp. 563–572.
- [17] M. N. Giannakos, K. Chorianopoulos, and N. Chrisochoides, "Making sense of video analytics: Lessons learned from clickstream interactions, attitudes, and learning outcome in a video-assisted course," *Int. Rev. Res. Open Distrib. Learn.*, vol. 16, no. 1, pp. 260–283, 2015.
- [18] K. Chorianopoulos, "Collective intelligence within Web video," *Human-Centric Comput. Inf. Sci.*, vol. 3, p. 10, Jun. 2013.
- [19] R. Cobos and L. Olmos, "A learning analytics tool for predictive modeling of dropout and certificate acquisition on MOOCs for professional learning," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manag. (IEM)*, 2018, pp. 1533–1537.
- [20] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [21] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, Oct. 2014.
- [22] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artif. Intell. Rev.*, vol. 54, pp. 1937–1967, Mar. 2021.

Gonzalo Martínez-Muñoz received the university degree in physics and the Ph.D. degree in computer science from the Universidad Autónoma de Madrid (UAM), Madrid, Spain, in 1995 and 2006, respectively.

From 1996 to 2002, he worked in industry. Until 2008, he was an Interim Assistant Professor with the Computer Science Department, UAM. In 2008/2009, he was a Fulbright Postdoctoral Researcher with Oregon State University, Corvallis, OR, USA, in the Group of Prof. Thomas G. Dietterich, world reference in the field of Machine Learning. He is currently an Assistant Professor with the Computer Science Department, UAM as well as the Coordinator of the master's program on Bioinformatics and Computational Biology. In addition, he is the IP of several research projects related to AI. He has published his research in some of the journals with highest impact in machine learning. His research interests include machine learning, decision trees, and ensemble learning, and applications of machine learning to science, sports, and education.

Miguel Ángel Álvarez-Rodríguez received the bachelor degree (with Hons.) in computer science from the Universidad Autónoma de Madrid, Madrid, Spain, in 2018.

He worked with UAM/IBM Chair of Large Systems and the industry as a Data Engineering for Financial Services with Ebury, Amsterdam, The Netherlands. He is currently a Senior Data Engineering with Twinco Capital.

Estrella Pulido-Cañabate received the Ph.D. degree in computer science from the Faculty of Computer Science, Polytechnic University of Madrid, Madrid, Spain, in 1989, and the master's degree in artificial intelligence and the Ph.D. degree in computer science from the University of Bristol, Bristol, U.K., in 1992 and 1996, respectively.

She worked in several Research and Development Departments from 1987 to 1991.

She was with the Higher Polytechnic School, Autonomous University of Madrid, Madrid, since October 1996 (as an Assistant Professor of Languages and Computer Systems, since July 2000) dedicating herself to teaching, research, and university management. She is the Co-Director of the UAM/IBM Chair of Large Systems, Cognitive Solutions, and Big Data Analytics, in 2013. Her research focuses on two main fields, the use of ICT in Education and Knowledge Management and Data Analysis.