Can Typical Behaviors Identified in MOOCs be Discovered in Other Courses?

Truong-Sinh An
Beuth Hochschule für Technik Berlin
Luxemburger Str. 10
13353 Berlin, Germany
truong-sinh.an
@beuth-hochschule.de

Christopher Krauss
Fraunhofer FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
christopher.krauss
@fokus.fraunhofer.de

Agathe Merceron
Beuth Hochschule für Technik Berlin
Luxemburger Str. 10
13353 Berlin, Germany
agathe.merceron
@beuth-hochschule.de

ABSTRACT
The emergence of Massive Open Online Courses (MOOCs) has enabled new research to analyze typical behaviors of learners. In this paper, we investigate whether this research is generalizable to other courses that are backed by a learning management system (LMS) as MOOCs are. Building on methods developed by others, we characterize individual learning behaviors in different ways taking into account specificities of the LMS we use. We then apply clustering techniques to uncover typical behaviors in two university courses. One course, JavaFX, teaching about the software programming framework, has been offered as a supplementary online course to students enrolled in an online degree. Enrolling in this course was voluntary and students did not earn any credit towards their degree; in this sense, the JavaFX course bears similarities to a MOOC though it is neither *massive* nor *open to everybody*. The other course is a classical face-to-face course on Advanced Web Technologies (AWT) backed by our LMS; students earn a degree when they pass the final exam. It turns out that the different characterizations of individual learning behaviors are consistent for the JavaFX course and uncover typical behaviors reminiscent of those found by others in MOOCs, while they aren't as applicable to the AWT course. However, typical behaviors found in the AWT course give insights on styles that lead to better marks.

Keywords
MOOC, Typical behaviors, X-means clustering

1. INTRODUCTION
The emergence of MOOCs with the general observation of their low completion rates has triggered new research to analyze typical behaviors of learners in MOOCs and brought forth evidence for various engagement/disengagement patterns such as *completing*, *auditing*, *disengaging* and *sampling, as* proposed by Kizilcec et al. [1]. In their paper, Kidzínski et al. [2] write that categorization schemes as found in [1] and others "remain robust in terms of generalizability within the MOOC's context, but they are hard to generalize outside of it". In this paper, we tackle that claim. We investigate whether this research can offer interesting insights to other courses that are backed by a learning management system (LMS), even though analyzed courses are not necessarily massive nor open, and even not completely online.

We investigate two courses presented with the Learning Companion App (LCA) [3]. The LCA is a LMS designed in the first place for vocational training. Compared to other LMSs common in higher education like Moodle, the Coursera-platform or edX, LCA has two salient features to encourage self-reflection and support efficient learning. The first feature concerns the learning objectives that need to be associated with each learning object (LO) in the course. All the learning objectives of one chapter are displayed for rating at the beginning and at the end of any chapter. A learner can assess how much s/he knows each learning objective. These self-assessments encourage learners to reflect on their previous knowledge, and on how much they know after learning the chapter. The second feature is a recommendation engine that suggests learners what to learn next [4]. Learners are free to consult these recommendations. Comprehensive user interactions are stored as xAPI statements [5]. The LCA is independent of any topic and any institution and, therefore, can be used in other contexts and for other courses.

The two courses considered in this study, JavaFX and Advanced Web Technologies (AWT) have taken place in the context of higher education. The JavaFX course has been offered as an optional online course to students enrolled in an online degree in computer science. These students learned to program graphical interfaces with the older framework Swing instead of the newer framework JavaFX. By taking part in this course, students did not earn any mark for their studies, they only increase their knowledge of the topic. The AWT course targeted master computer science students. It was a classical face-to-face course taught with the support of the LCA in winter semester 2016/17. When enrolled in this course, students usually had the aim of passing the final exam and earn the corresponding credits for their master degree.

In this study, we follow and adapt the approach of [1, 6] and explore several different ways of qualifying individual learning behaviors as similar. It turns out that for the JavaFX course, these different ways are consistent and uncover two to three typical learning behaviors reminding those exhibited by Kizilcec et al. [1]. For the AWT course, only one way of qualifying behaviors turns out to be sensible. The uncovered typical learning behaviors from both courses match those exhibited in [1, 6] and give insight on styles that lead to better marks.

This paper is organized as follows. Related works are discussed in Section 2. Specificities of courses in our LMS, the Learning Companion App, are presented in Section 3. Subsequently, different ways of characterizing individual learning behaviors are explained and typical learning behaviors found in both courses are presented and discussed. Conclusion and future works are given in Section 5.

2. RELATED WORK

Kizilcec et al. [1] investigated learners' engagement in courses from Coursera which consist of weekly videos and assessments, and proposed four typical engagement / disengagement patterns that they call

- *Completing*: "learners who completed the majority of the assessments offered in the class",
- *Auditing*: "learners who did assessments infrequently if at all and engaged instead by watching video lectures",
- *Disengaging*: "learners who did assessments at the beginning of the course but then have a marked decrease in engagement (their engagement patterns look like Completing at the beginning of the course but then the student either disappears from the course entirely or sparsely watches video lectures)" and
- *Sampling:* "learners who watched video lectures for only one or two assessment periods".

These categories have been identified in three courses; however, their proportions differ in each course. To discover these categories, they have first characterized a student by a tuple giving her/his status each week: "*on track [T]* (did the assessment on time), *behind [B]* (turned in the assessment late), *auditing [A]* (didn't do the assessment but engaged by watching a video or doing a quiz), or *out [O]* (didn't participate in the course in that week)" [1].

In an attempt to replicate the work of [1], Ferguson and Clow [6] suggest that the methodology used to uncover typical learning behaviors in a MOOC's context does not necessarily generalize to another MOOC adopting different elements of pedagogy and learning design. Since the courses analyzed in [6] follow a social constructivist pedagogy, Ferguson and Clow adapt the methodology of [1]. They consider also participation in discussions and end up with 10 values to characterize the weekly status of a student, instead of the four values T, B, A and O introduced in [1]. They have identified the following typical learning behaviors: *Samplers* ("Learners in this cluster visited, but only briefly", similar to sampling above), *Strong Starters* ("these learners completed the first assessment of the course, but then dropped out"), *Returners* ("these learners completed the assessment in the first week, returned to do so again in the second week, and then dropped out"), *Mid-way Dropouts* ("these learners completed three or four assessments, but dropped out about half-way through the course"), *Nearly There* ("these learners consistently completed assessments, but then dropped out just before the end of the course"), *Late Completers* ("this cluster includes learners who completed the final assessment, and submitted most of the other, but were either late or omitted some") and *Keen Completers* ("this cluster consists of learners who completed the course diligently, engaging actively throughout" similar to completing above). The two approaches in [1, 6] share the same principle of selecting a priori features that are sensible to describe a student's individual engagement, and then use K-means clustering to discover typical learning behaviors.

Gelman et al. [7] adopt a different, more bottom-up approach to discover typical behaviors: they use a set of 21 features that they can extract week by week from the log data and adapt non-negative matrix factorization to obtain weekly behaviors that are supported by a combination of those features. This approach is attractive because it does not need a careful selection of features to characterize the behavior of a student; instead, the algorithm selects and combines features from the set it receives as input. A difficulty lies in the interpretation and the practical use of the discovered behaviors. While an *auditing* behavior "learners who did assessments infrequently if at all and engaged instead by watching video lectures" [1] is easy to derive, it is less clear what a weekly *deep* behavior "the associated students must have spent a long time on a single resource" [7] means for educators.

In this paper, we adopt the first approach and adapt it to our context, taking inspiration from the work in [6].

3. COURSES IN THE LCA

The Learning Companion App (LCA) is a whole infrastructure that can be thought of as LMS equipped with a repository for learning objects, a recommendation engine and a learning analytics module. It is at the same time an App in responsive design which is the entry point for students to access courses, learning objects (LOs) and lecture schedule as well as to get recommendations for the next best contents to be learnt; furthermore, it triggers the tracking of all relevant user interactions [8].

In LCA, each learning object at the lowest level is paired with its metadata that includes at least one learning objective, a typical learning time and its prerequisites. A learning object can be a piece of text, a video, an exercise (similar to an exercise of an assessment in a MOOC), an animation, even a downloadable document and so on. Learning objects are bundled into learning units and a course is essentially a sequence of learning units. The learning objectives of a learning unit are the union of the learning objectives of its learning objects. A learning unit is rendered in the LCA as an "accordion" GUI element with a specific sequential structure. The top item of the accordion view that can be opened is the list of the learning objectives of that unit. Learners can rate each learning objective and so indicate how much they know already on that topic, from 1 "know nothing" to 5 "expert". We call this list *self-assessments*. This item is followed by the sequence of the LOs of that unit. The user can interact with the learning objects by clicking on the title in the accordion view whereupon the requested content is presented. The user is only shown one learning object at any time so that s/he can concentrate fully on this content. Following the sequence of LOs, the next item in the accordion view is again the list of learning objectives. By rating them, a student can reflect on how much s/he knows after learning the unit. The next item in the accordion view allows students to provide feedback on the typical learning time for that unit (from 1, "way too little time" to 5, "way too much") and give comments. The last item in the accordion view opens a discussion thread on that unit. Apart from its sequence of learning units, a course contains a schedule which specifies dates for the start and end of the course, as well as when each learning unit should be learned.

All users' interactions are stored using the xAPI specification [5] in the open-source learning record store called Learning Locker[1]. The accordion view allows inferring how long any item of the view is opened. Typical mined data include number of clicks on all items of the accordion view (self-assessments, LOs, feedback, discussion threads), time an item is open, answers and performance in exercises, ratings of pre- and post-study self-assessments, feedback, messages of discussion threads. Note that a student can access any LO directly by clicking on the recommendations. For this study, this does not change the kind of interactions that are stored.

The two courses discussed in this paper make all the learning material available from the start of the course to encourage self-pacing and self-organization of students. Furthermore, the time schedule of the courses is indicative only, in the sense that there is no penalty if someone does not follow the schedule. Finally, in both of them, students did not post in the discussion threads; they only wrote (few) comments in the feedback area. However, the two courses differ significantly in their didactical organization and contents.

The JavaFX online course, available for a period of 11 weeks, offers an introduction into the FX-Framework for the development of platform independent Java applications and targets bachelor computer science students. This course is suggested as an optional online course to students enrolled in an online computer science bachelor course. By taking part in this course, students do not earn any mark for their studies, only knowledge for themselves.

It comprises three learning units. Each learning unit has about five learning objectives and contains about fifteen to thirty LOs (units are not of equal length). About half of the LOs are texts to explain concepts and example programs, and half are exercises (single/multiple choice, cloze tests and so on). The last LO before the self-assessment of the learning objectives is a comprehensive programming task; students can send their program per email and obtain a manually commented evaluation. Based on the educational discussion on MOOCs, Daniel [9] pointed out that "students seek not merely access, but access to success". However, success can be different for each student. Driven by this consideration, a specific LO has been added to this course allowing each student to rate her/his motivation on a scale from 0 (do nothing) to 100 (engage thoroughly with everything offered). 51 students enrolled in this course; however, there were 23 no-shows (defined in [10] as "people register but never login to the course while it is active"). Only the remaining 28 students are considered for the analysis in this paper. The 28 users generated 3624 xAPI statements in total during the course.

Advanced Web Technologies (AWT) targets master computer science students. Technical experts teach in 12 weekly presence lectures diverse topics that are of interest for future web developers – from web technology basics, such as HTML, over media delivery and content protection, to personalization through recommender systems, and the Internet of Things. The lectures are mostly held with slides created in PowerPoint showing definitions, specifications, and source code, animations for concepts and videos for practical examples. The about 1000 presented slides are converted to digital learning objects, one slide being a single LO, and grouped into 105 learning units for the representation in the LCA – with videos, animations and additional multiple-choice questions at the end of the learning units. Moreover, as some students still want to learn with a printed version of the slides, the last LO of the accordion view is a downloadable PDF file containing all the slides of the unit.

142 students enrolled for AWT in winter semester 2016/17; however, there were 43 no-shows. Only the remaining 99 students are considered for the analysis in this paper. Especially in the first weeks before the official registration deadline, students frequently change their mind regarding participating in specific courses – which might explain the high loss ratio of the participants. At the end of the course, students can earn credits by completing an one-hour exam consisting of 50 multiple choice questions and 5 bonus questions. Exactly 75 students completed the final exam (even two who did not used the LCA) and the average mark was 1.90 (only one student failed the exam; note that the best mark is 1.0 and the worst possible mark is 5.0). The 99 users generated 92825 xAPI statements in total during the course.

In contrast to the courses offered by [1], [6] and the JavaFX course, the primary goal for students of AWT is to pass the final exam. AWT does not offer any intermediate assessment. Students access online material, first and foremost, for the wrap-up of face-to-face lectures and for exam preparation.

4. METHODOLOGY AND RESULTS

In our context, there are multiple sensible ways to compare students in their learning behaviors. Because this time schedule is purely indicative for students and all the materials are available from the start of the course, we compared students on how they have interacted with the course independently of time. In this paper, we investigate four such ways.

Clicks only: In this way, we consider only click counts per learning object. A student is represented by a vector that represents how many times s/he has clicked each element of the whole course. In this way, two students are similar if they access almost the same learning objectives, learning objects, feedback, and motivation (for JavaFX only as AWT does not have this feature) a similar number of times.

Elapsed time: In this way, elapsed time spend on that learning object replaces click count. A student is represented by a vector that has the size of all learning objects of the course. The learning objectives, feedback, and motivation are not considered because the time spend is not tracked individually for these features. Two students are similar if they spend a similar overall time on the same learning objects (texts, videos, exercises, etc.). The overall time is the sum of the elapsed times in each visit.

Assessment scores: In this way, we consider performance on all assessments, including programming tasks of the JavaFX course. A student is represented by a vector that has the size of all assessments; values are ratings given in all self-assessments, marks earned in all exercises, rating given in feedback and motivation (AWT does not have the motivation feature). The final exam for AWT is not considered. Two students are similar if they achieved similar scores on all assessments.

Elapsed time and assessment scores: In this way, we consider a combination of the latter two: elapsed time on what students look at (texts, videos and so on) and scores on what students answer (self-assessments, exercises and so on). Two students are similar if they spend a similar overall time on similar learning objects

[1] Learning Locker. See: https://learninglocker.net/

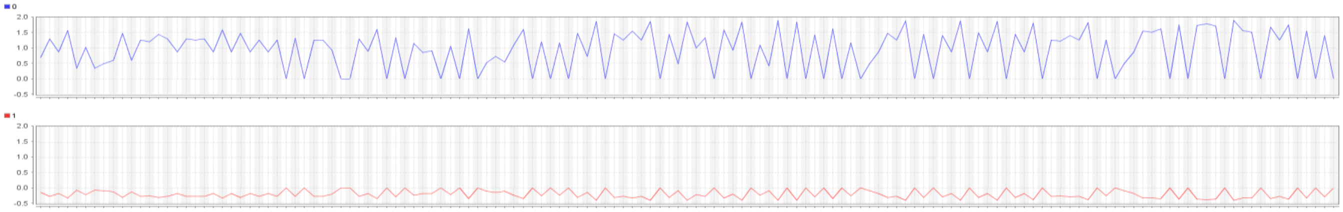

Figure 1: Plot of the centroids of the 2 clusters returned by X-means in the JavaFX course. The x-axis represents all the elements of the course (learning objectives, learning objects etc.); the y-axis gives the average normalized number of clicks per element.

such as texts, videos, slides and so on (that are not exercises) and achieve similar scores on all assessments.

We used RapidMiner[2] and applied the X-means clustering algorithm with Euclidean distance. X-means finds an optimal number of clusters and is known to find fewer clusters than K-means [11]. Due to the size of the vector representing each student (in the way *Clicks only* a student is represented by a vector with 143 values in the JavaFX course) and the small data sets, clustering is challenging. Furthermore, in RapidMiner, X-means is implemented in such a way that it will always find a minimum of two clusters, even if the data is uniform. To validate that the data does cluster naturally, we applied also K-means and checked for the drop in the curve plotting K against the sum of squared errors (which corresponds to the *average within distance* of RapidMiner). Values of clicked counts and elapsed time have been normalized. Assessment values like marks in exercises or self-assessments are already stored as scaled values.

4.1 JavaFX

X-means returns exactly the same two clusters for three approaches: *Clicks only*, *Elapsed time* and *Elapsed time and assessment scores* (the results of the fourth approach are described later on). Figure 1 shows a visualization of these two clusters for the *Clicks only* way; it lists all elements of the course on the X-axis and shows the corresponding normalized number of clicks of the clusters' centers on each element. The first cluster (cluster 1, the blue line in the upper diagram of Figure 1) consists of 5 students who engage with many elements such as self-assessments, learning objects and also interact with the automatically generated features like feedback. When sorting the students according to the number of distinct elements they have accessed in the course, these 5 students come on top. On average, students in this cluster have accessed 72 distinct course elements. If the elements are restricted to the exercises only, as they best match assessments in MOOCs, these 5 students remain on top: except for one, who performed 15 exercises, they have performed 25 to 30 exercises out of 34. The other 23 students in the course, represented in the second cluster (cluster 2, red line and bottom diagram of Figure 1) accessed the learning objects less often and did very few self-assessments. On average, students in this group have accessed 10 distinct elements of the course and solved exercises infrequently, if at all four times or less. Transferred to the categories in [1], we find that these two patterns of engagement are reminiscent of *completing* and *auditing* but without any reference to time. In [1] it is clear that *completing* students have solved assessments week by week because assessments are available in the course week by week only. In our course, *completing* students could have solved exercises regularly, or all during a few weeks only, depending on their own time-management.

The K-means algorithm finds an optimal set of 4 clusters; see the upper elbow-curve of Figure 2 with the drop when k is 4. One cluster matches exactly cluster 2 found with X-means, while the cluster with 5 students is split into 3 clusters. This finding shows that data naturally clusters; however, the two clusters returned by X-means are more interpretable.

X-means returns three clusters when using *Assessment scores*. Cluster 1 with the pattern *completing* is also found here. Cluster 2 above is now split into two clusters: one with 18 students and cluster 3 with 5 students. What distinguishes these 5 students from the remaining 18 students is that they answered self-assessments and engaged with exercises mostly from the first unit of the course, hardly from the follower units. They correspond to *disengaging* in [1] although beginning of the course does not refer to time but to the sequence of the units that are displayed in the LMS. K-means algorithm finds an optimal set of 5 clusters; as before, the *completing* cluster is split into 3 clusters.

At first, it may be surprising that the three characterizations: *Clicks only*, *Elapsed time* and *Elapsed time and assessment scores* give exactly the same clusters: *completing* and *auditing*. With some consideration, this result is understandable: what distinguishes the most two learners is when one has accessed an element and the other not. A *completing* student has accessed

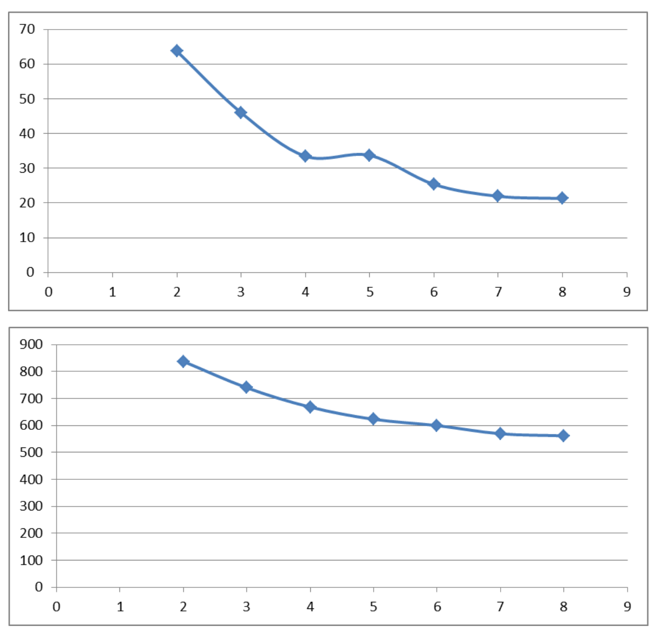

Figure 2: Plot of K against average within distance scenario clicks only for JavaFX (above) and AWT (below).

[2] Rapid Miner. See: https://rapidminer.com/

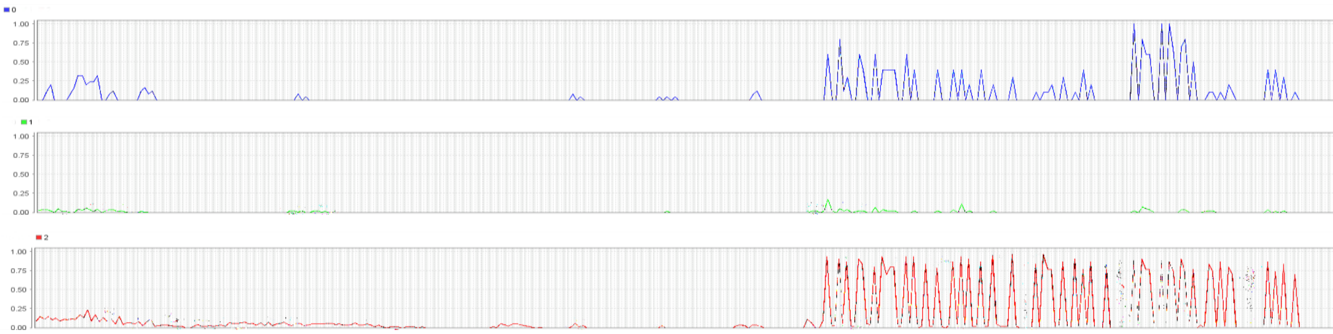

Figure 3: Plot of the centroids of the 3 clusters returned by X-means in the AWT course. The x-axis lists all the assessments of the course (self-assessments left part, exercises and feedback on time right side); the y-axis gives the scaled score of the center per element.

much more elements of the course than an *auditing* student; these two behaviors are discovered by X-means. The characterization *assessment scores* reduces the number of features used to perform clustering (interactions with LOs such as text or videos are omitted) and allows for distinguishing a sub-category in the *auditing* group: *disengaging*; those learners are completing activities primarily in the first unit of the course and then stop. They have hardly engaged with the course in the following units, what makes them similar to *auditing* students in the three other ways: they have engaged infrequently with exercises and have looked at few learning objects.

4.2 AWT

The first three approaches (*Clicks only*, *Elapsed time* and *Elapsed time and assessment scores*) lead to no meaningful results for the AWT course. On the one hand, K-means does not show a natural clustering of the data for any of these ways: plotting K against the *average within distance* does not show any drop, as the curve for the AWT course in Figure 2 bottom shows. On the other hand, these three ways are not really adequate to describe the engagement of an individual student due to the digital content of this course: at the end of each unit, there is a .pdf file containing all the slides of this unit. A student might download only the .pdf file of each unit and look at it as much as s/he wants, another student might access all the slides online multiple times. From the interactions that are stored and evaluated, these two students look very different, yet their learning behaviors are similar. At the beginning of the course, 66 Students have requested PDFs, and this number of students decreased to the end of the course to 19. One third of all students have requested all PDFs.

In contrast, for the *Assessment scores* approach, X-means generated three definite clusters. Figure 3 shows a visualization of these three clusters; it lists all assessments of the course on the X-axis and shows the corresponding score of the clusters' centers on each element. Two parts are clearly distinguishable: a rather flat left part and a right part where the blue (top) and the red (bottom) lines show spikes. The rather flat left part corresponds to the self-assessments; generally, not many students rated themselves. The right part corresponds to the exercises and student feedback Cluster 1 contains 9 students inclusive the one who did not pass the final exam (the upper diagram with the blue line of Figure 3). Students in this cluster provided self-assessments in the first three units, and worked out exercises but did not achieve good scores. They remind of *Strong Starters* and *Returners* proposed in [6] when this vocabulary is adapted to the sequential order of the units instead of the first weeks of the course. To some extent, they exhibit also some kind of *completing* pattern in terms of exercises, because they completed almost half of them: on average 22 from a total of 48. Their average mark in the final exam is 2.03 which is slightly worse than the general average of 1.90. The biggest cluster contains 64 students (cluster 2, the diagram in the middle with the green line of Figure 3) and is similar to the pattern *auditing* because they did exercises infrequently if at all: on average 1 out of 48. However, they did access .pdf files. All learners who did not participate in the final exam fall into this cluster. The average mark of the students in this cluster who participated in the final exam is 2.23 (no-shows are neglected), which is below the general average. The last cluster contains 26 students and shows a *completing* pattern (cluster 3, the bottom chart with the red line of Figure 3). If one sorts the students according to the number of distinct exercises they have solved in the course, 25 of these 26 students are the top 25. They have worked on nearly all the exercises, on average 42 out of 49, and completed almost all of them correctly. The final exam mark in this completing cluster reaches 1.50 on average, a better mark than the overall average of 1.90. The last two clusters are interesting: a *completing* student does well in the final exam, while an *auditing* student does worse in the final exam or even does not attend it. Although, as opposed to [1], these patterns do not tell anything on when students accessed the assessments in the time schedule.

K-means algorithm finds an optimal set of 4 clusters. It finds exactly the same big cluster of 64 students and finds almost the same first cluster as X-means does. However, it splits the last cluster to isolate three students. Students in both groups still solved in average 42 exercises but they differ in how they engaged with self-assessments. The small group of 3 students rated 74 self-assessments in average and the other students only rated 3 self-assessments in average in the first units of the course.

5. DISCUSSIONS AND FUTURE WORK

Considering the particularities of our courses, we have defined four meaningful ways of characterizing an individual learning behavior. We have used X-means clustering to extract typical learning behaviors from two distinct university courses, an optional online JavaFX course and a compulsory face-to-face course about Advanced Web Technologies. Because of the small data sets, particularly for the JavaFX course, clustering is challenging. We found that students do not act at random. In the JavaFX course, we could derive evidence behaviors that remind of patterns found in [1]: *completing*, *auditing*, and *disengaging*. Only the *Assessment scores* way delivers reliable clusters for AWT. From the three clusters uncovered by X-means, two are particularly interesting. All students that were ultimately not participating in the final exam were located in the *auditing* cluster. Other students in that cluster, who participate in the final exam, tend to do less well than average. Students of the *completing* cluster tend to pass the exam with very good marks. Note that *completing*, *auditing*, and *disengaging* in this paper are similar to [1] in terms of which kind of learning material has been accessed frequently or not; as opposed to [1], our approach does not provide information on when in the time schedule the material has been accessed.

The present results suggest that typical behaviors found in MOOCs can be transferred to other courses - with care. This situation bears similarities with predicting students at risk of deserting a course. Numerous articles show that models with good predictive power can be built to predict drop-off and also the performance of students in a course. These articles show also that there is no set of features and no classifier that works best in all contexts: no one-size fits all. On the contrary, the set of features and classifiers needs to be adjusted to the data and setting at hand to achieve a good predicting power. The work of [2] also supports this view for MOOCs. Our results suggest that the situation is the same for typical behaviors. We adjusted methods of others to our context and were able to extract interesting and interpretable typical behaviors from relatively small data sets. This work considers rather simple features like clicks and elapsed time. Future work should focus on a more sophisticated feature extraction.

In our setting, there is a time schedule, even if it is indicative only. It could make sense to devise ways of characterizing an individual behavior taking this time schedule into account. The method of [1] needs careful adaptation because a learner might be *on track* or *behind* and might also be *early*. Works on these lines have already begun. Preliminary work shows that four of the five students of the *completing* cluster of the JavaFX course began only after three weeks to engage with the course, while the majority of the *completing* cluster of the AWT course engaged with the course regularly each week. Another future work is to reflect on implications for the recommendation engine and the learning analytics module. Should the recommendation engine be adjusted to different typical behaviors for example? We plan to integrate these findings in the overall behavioral feedback shown to students.

6. ACKNOWLEDGMENTS

The authors would like to thank the whole Smart Learning team for their great work and many constructive ideas. We would also like to thank the instructors of the AWT course: Stephan Steglich, Louay Bassbouss, Stefan Pham, and André Paul. Special thanks go to Christian Fuhrhop, who proofread this paper and made a lot of practical suggestions.

The Smart Learning project is sponsored by the German Federal Ministry of Education and Research (Bundesministerium fuer Bildung und Forschung – BMBF) under the project funding number 01PD14002A.

REFERENCES

[1] Kizilcec, R.F., Piech, C., and Schneider, E. 2013. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In Third international conference on learning analytics and knowledge LAK'13, April 8-12, Leuven, Belgium. ACM, p. 170-179. doi>10.1145/2460296.2460330.

[2] Kidzínski, L., Sharma, K., Boroujeni, M. S., Dillenbourg, P. 2016. On generalizability of MOOC models. In Proceedings of the 9th International Conference on Educational Data Mining, EDM'2016, Raleigh, USA, June 29-July 2, p. 406-411.

[3] Krauss, C.; Merceron, A.; An, T.-S.; Zwicklbauer, M.; Arbanowski, S.. The Smart Learning Approach - A mobile Learning Companion Application. In: Proceedings of The Eighth International Conference on Mobile, Hybrid, and On-line Learning (eLmL 2016). IARIA, April 24 - 28, 2016 - Venice, Italy.

[4] Krauss, C. 2016. Smart Learning: Time-Dependent Context-Aware Learning Object Recommendations. Proceedings of the 29th International Florida AI Research Society Conference (FLAIRS-29), AAAI, Key Largo, p. 501-504.

[5] Kevan, J. M. and Ryan P. R., "Experience api: Flexible, decentralized and activity-centric data collection," Technology, Knowledge and Learning, vol. 21, no. 1, 2016, pp. 143–149.

[6] Ferguson, R. and Clow, D. 2015. Consistent commitment: Patterns of engagement across time in Massive Open Online Courses (MOOCs). Journal of Learning Analytics, 2(3), 55–80. http://dx.doi.org/10.18608/jla.2015.23.5.

[7] Gelman, B., Revelle, M., Domeniconi, C., Johri, A., and Veeramachaneni, K. 2016. Acting the Same Differently: A Cross-Course Comparison of User Behavior in MOOCs. In Proceedings of the 9th International Conference on Educational Data Mining, EDM'2016, Raleigh, USA, June 29-July 2, p. 376-381.

[8] An, T.-S., Dubois, F., Manthey, E., Merceron, A. 2016. Digitale Infrastruktur und Learning Analytics in Co-Design. In Proceedings of the Workshop Learning Analytics, co-located with the 13th e-Learning Conference of the German Society for Computer Science, Potsdam, Germany, September 11, 2016. http://ceur-ws.org/Vol-1669/.

[9] Daniel, J. 2012. Making sense of MOOCs: Musing in a maze of myth, paradox and possibility. Journal of Interactive Media in Education, 2012(3). http://dx.doi.org/10.5334/2012-18.

[10] Hill, P.. Emerging student patterns in MOOCs: A (revised) graphical view, 2013. http://mfeldstein.com/emerging-student-patterns-in-moocs-a-revised-graphical-view/ accessed March 3, 2017.

[11] Pelleg, D., and Moore, A. 2000. X-Means: Extending K-means with Efficient Estimation of the number of Clusters. In Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29, p. 727-734.