Semi-Markov model for simulating MOOC students

Louis Faucon, Łukasz Kidziński, Pierre Dillenbourg
Computer Human Interaction in Learning and Instruction
École Polytechnique Fédérale de Lausanne
{louis.faucon,lukasz.kidzninski,pierre.dillenbourg}@epfl.ch

ABSTRACT

Large-scale experiments are often expensive and time consuming. Although Massive Online Open Courses (MOOCs) provide a solid and consistent framework for learning analytics, MOOC practitioners are still reluctant to risk resources in experiments. In this study, we suggest a methodology for simulating MOOC students, which allow estimation of distributions, before implementing a large-scale experiment.

To this end, we employ generative models to draw independent samples of artificial students in Monte Carlo simulations. We use Semi-Markov Chains for modeling student's activities and Expectation-Maximization algorithm for fitting the model. From the fitted model, we generate simulated students whose processes of weekly activities are similar to these of the real students.

Keywords

MOOCs; simulation of students; generative models; Expectation-Maximization; Semi-Markov chains; Bayesian statistics

1. INTRODUCTION

Vast amounts of data which we gather and analyse in modern learning environments allow us to build models of unprecedented scale and accuracy. This phenomenon, in parallel with developments in computer science, gave rise to new possibilities of inference from educational environments. In particular, the growing field of Simulated Learners [8, 11, 14] provides us with tools for inference from educational simulations.

Inference from any simulations is bounded by the predefined level of abstraction of the analysis. In the context of Massive Online Open Courses (MOOCs), on one hand as an educational institution we have access to only a handful of MOOCs, on another hand, we have data as granular as student's clickstream in a video player. We are therefore obliged to model granularity robustly, depending on the availability of the data. We argue that understanding the properties of the statistical methodology at hand is crucial for successful inference.

We propose a probabilistic model, based on extended version of Markov Chains, called semi-Markov Chains. In the model, we can balance the complexity of the structure and the number of parameters to estimate by cross-validating its parameters. We present an algorithm for fitting the model as well as illustrative examples of the fit on a set of MOOCs.

The contributions of this paper are threefold. First, **we investigate to what extent Semi-Markov chains can be used to describe behavioural patterns of students (RQ1)**. Second, since our model implicitly divides users into clusters, **we analyse if these clusters are interpretable (RQ2)**. Third, **we analyse how these models can be used to infer distributions of events (RQ3)**.

2. RELATED WORK

Modeling students is a key concept in learning analytics and educational research in general. Researchers build models predicting motivation and cognition, based on student's goals [19] or they predict goals by motivational traits [7]. Large datasets allow researchers to find predictive power of seemingly slightly related signals like the length of pauses in a video [12] or potentially noisy signals like head movement in the classroom [16].

2.1 Generative models in MOOCs

All the aforementioned models are focused on prediction and belong to the class of so-called discriminative models. In this study, we suggest a generative model, which allow us not only to predict, but also to generate observations from the estimated distribution. These models capture the probability structure of input variables and the flow of the processes. Several generative models in MOOCs have been applied, e.g. to forums [3].

Among many generative models that can be encountered in educational research, Markov models were employed for visualization [5], for modeling engagement [17] and for modeling students retention [1].

2.2 Simulated learner

The area of simulating students' behaviour lays on the intersection of cognitive science and artificial intelligence. Examples of applications of simulation of students can be found even outside computer science, where the teacher simulates student's response in order to self-improve instructional skills [18]. An acknowledged example of the usage of simulating humans [9] for education deals with simulations of patients behaviour for training medicine students.

Emergence of Internet and new data storage techniques allow researchers to collect and analyse massive amounts of information about the users. Researchers employ simulations for clustering students [13]. For a review of earlier techniques we refer to [2]. We motivate our methodology by the advancements of user modeling in web context [4], as we find this environment conceptually close to the environment of a MOOC.

3. GENERAL FRAMEWORK

3.1 Dataset

From our internal MOOC database, aggregating data from Coursera and edX, we extracted events for 61 EPFL courses. The raw data contained approximately 23 million events for 500,000 students, arranged in tuples: `<StudentID, CourseID, EventType, Timestamp>`. The *EventType* describes the type of an activity and takes one of four possible values presented in Table 1. We choose these events as the most discriminative actions from the key areas: learning, validation and community engagement. Note that our modelling technique can be easily extended to cover other types of events.

Abbreviation	Description	Proportion
`VideoPlay`	watching a video	51%
`Submission`	submitting an assignment	33%
`ForumView`	visiting the forum	15%
`ForumPost`	posting on the forum	1%

Table 1: Distribution of events in the dataset.

For the analysis we developed our own Python implementation of the algorithm fitting the model[1]. In Section 5 we explain the algorithm in detail. Since 23 million events can still fit in memory of a single computer, we did not require a specific computing architecture to perform the analysis. However, given the considerable size of the dataset, the algorithm takes several minutes to run.

3.2 Definitions

We start with a general framework, in which student's activity in any MOOC can be very precisely described. Next, we elevate abstraction of the model by adding assumptions simplifying the analysis. Our goal is to introduce a model whose complexity can be adapted to the structure of a course and the amount of available data.

We consider a model in which students behaviour is described in a sequential manner by the type of activity they perform and the time they wait between two sessions. Furthermore, as most of the students perform at most 1 MOOC session per day, we choose a daily granularity of actions.

A sequence of student's daily activity is described as a list of 'active events' (`VideoPlay`, `Submission`, `ForumView` and `ForumPost`) followed by a 'end of the day event' (`EndOfDay`) or only a `EndOfDay` in the case the student did not perform any activity the given day. The formal definition of the model is following:

The set of all students $\mathscr{S}$: We use the symbol $s \in \mathscr{S}$ to designate an individual student.

The set A of all types of activities: For this study we chose a set of four types of events: { `VideoPlay`, `Submission`, `ForumView`, `ForumPost` }. We add to this set one special type of event, `EndOfDay`. This event corresponds to the end of interactions with MOOCs on a given day. We use the symbol $a \in \mathbf{A}$ to designate any type of activity. One can extend the set of activities to other events if needed for certain application.

Note that we do not specify the regular 'end of a course' event, since we only model the behaviour within the limited time-frame of a course and we treat the last day of the course as the last day of the process. Therefore, each student who went through the whole course without dropping out has just a `EndOfDay` event on the last day of the course. Number of `EndOfDay` events is therefore equal to the number of days of the course.

The random sequential variable $\mathbf{X}_1^{(s)}, \mathbf{X}_2^{(s)}, \ldots, \mathbf{X}_n^{(s)}$ represents the sequence of activities of one student s. Each $\mathbf{X}_i^{(s)} \in \mathbf{A}$ and the sequence stops after an `EndOfDay` when the student reaches the end of the course. We denote the length of the sequence for a student s as $n^{(s)}$. The observation of one student activity along one MOOC is thus a **realization** of the random sequence $\mathbf{X}$.

The probability distribution P: In general, for each student $s \in \mathscr{S}$ we can model the i-th event $\mathbf{X}_i^{(s)}$ with a probability distribution

$$\mathbf{P}^{(s)}(\mathbf{X}_i^{(s)} = a \mid \mathbf{X}_{i-1}^{(s)}, , \mathbf{X}_{i-2}^{(s)}, \ldots, \mathbf{X}_1^{(s)}, \mathbf{C}_s),$$

where $a \in \mathbf{A}$, $\mathbf{X}_1^{(s)}, \ldots, \mathbf{X}_{i-1}^{(s)}$ are the previous events of that student and $\mathbf{C}_s$ are personal characteristics of the student.

This distribution represents the student's behaviour profile and allows to generate typical sequences of activities. Our main objective is to model this distribution as accurately as possible, given the limited information. The accurate distribution would allow us to draw samples of students.

3.3 Assumptions

As discussed in the previous section, assessing $\mathbf{P}^{(s)}$ is unfeasible due to dependence on too many events in the past and due to the lack of information on personal student features. In order to fit a probabilistic model we need to relax these dependencies. We introduce following assumptions:

- **A1** Students' behaviours fit into a small number of natural categories of behaviour.
- **A2** The type of activity depends only on his previous activity and not on old past activities.

Assumption **A1** maps the space of all possible students' characteristics into a limited number of categories, which are much easier to attribute. Many studies on MOOCs explicitly classify students into a small number of categories [10], students are divided between 'Viewers' who only watch videos, 'Forum Actives' who share with their peers in the MOOC discussion forum and 'Completers' who succeed in the assignments. As we present in the next section, our method is based on unsupervised clustering, where groups emerge in the way optimal in terms of maximum likelihood of the model.

Assumption **A2** we impose that only the last activity has an impact on the current activity. This assumption is more constraining, but since the complexity of history grows exponentially with the number of steps and, in order to be able to estimate parameters, we have to

[1] Our implementation is available under https://github.com/lfaucon/edm2016-mooc-simulator

reduce the search space. This simplification is usually called the 'Markov assumption'.

Apart from technical assumptions required for Markov Models, we impose other assumptions for convenience. First, we do not consider length of events, so the `VideoPlay` event is only the moment when a student starts watching a video. Second, if the series of events happens during midnight, still an event `EndOfDay` is added to the sequence.

4. PROBABILISTIC MODELING

4.1 Soft clustering

In Section 3 we proposed a simplified framework, in which we assume that there are only a few different possible classes of students **(A1)**. We enumerate clusters $1,2,...,K$. For each student $s \in \mathscr{S}$ we introduce a probability distribution $\mu_k^{(s)}$ which describes probability that the student belongs to the behaviour classes k, for $k \in \{1,2,...,K\}$.

This technique is often referred to as *soft clustering*, *weighted clustering* or *fuzzy clustering* [15]. Instead of discret cluster assignment, as for example in K-means, we obtain for each student a probability distribution among the clusters. These probabilities can be intuitively seen as our certainty that the student belongs to a given cluster.

4.2 Semi-Markov Chain

Assumption **(A2)**, i.e. dependence only on the last state, allows us to model the process Markov Chains. Formally, in the definition of distribution of the next event we can drop dependence of the events which occurred before the current one, i.e. we identify

$$\mathbf{P}^{(s)}(\mathbf{X}_i^{(s)} \mid \mathbf{X}_1^{(s)},...,\mathbf{X}_{i-1}^{(s)}) = P^{(s)}(\mathbf{X}_i^{(s)} \mid \mathbf{X}_{i-1}^{(s)})$$

A preliminary analysis revealed an important weakness of using classic Markov Models in our context. A traditional Markov model considers that a student is equally likely to stop watching videos when they have watched one, as when they have already watched ten videos. In practice, students watch videos sequentially and Markov Model does not capture appropriately the number of events in the sequence.

To remedy this issue we employed Semi-Markov Models (also called Markov Renewal Processes). The key feature of this model is that it allows to replace the self-loops (transitions from one event type to itself) in the Markov Chain, by a probability distribution of the number of repetition of a given state.

In Semi-Markov Models, we still need to choose a parametric distribution, but we have more freedom than in traditional Markov Chain. Markov Chain implicitly assumes that probability of staying in the same state is the largest for 1 step and decreases with number of steps. However, we would expect that 1 is not the most probable number of repetition at least for a particular group of students. This phenomenon can be captured by, for example, Poisson distribution, which proved to be more accurate in our preliminary analysis. Thus, for an event $a \in A$ and a class k we model the number of repeated events R_a^k by

$$\mathscr{P}(R_a^k = r) = \frac{e^{-\lambda_a^k}\left(\lambda_a^k\right)^r}{r!}$$

where r is the number of repetitions and λ_a^k is the average number of repetition and needs to be estimated from the data for each k and a.

To illustrate that the Poisson distribution improves the model, let us consider an example. Suppose we expect that some group of students connects to a MOOC twice a week, with approximately three days interval between connections. In that case, the average number of repetitions of the `EndOfDay` event is 3. Simple Markov Model, accurately models the average to be 3 but implicitly assumes that the majority of students gets only 1 repetition. Semi-Markov model with Poisson distribution also gives the average equal to 3 and the distribution is concentrated around 3.

5. FITTING THE MODEL

5.1 Algorithm

The Expectation-Maximisation (EM) algorithm has been introduced in 1977 in [6]. The goal of this iterative technique is to compute the parameters that maximize the likelihood of a given probabilistic model. The EM algorithm has been proven to converge at least to a local minimum. This minimum depends on the initialization point, thus multiple runs with different random initialisations are often used in practice in order to increase the chances of finding the global minimum.

In this study we use the EM algorithm for unsupervised learning. Neither the parameters of the latent classes nor the repartition of the students are known at the beginning and the algorithm has to estimate both quantities at once. In our settings, we define for each $k \in \{1,2,...,K\}$ and states a and b:

- $p_{b->a}^{(k)}$, the probability that a student with the behaviour profile k performs the activity a after the activity b:

$$p_{b->a}^{(k)} = \mathbf{P}(\mathbf{X}_i = a \mid \mathbf{X}_{i-1} = b)$$

- $\lambda_a^{(k)}$, the average number of repetitions of an event a from a student of profile k.

- $\mu_k^{(s)}$, the probability that a student s belongs to the profile k.

We can thus compute the likelihood of the observed sequence, as a function of cluster repartition and parameters of Markov Chains by

$$likelihood = \prod_{s \in \mathscr{S}} [\sum_{k=1}^{K} \mu_k^{(s)} \prod_{(a,b,r) \in \mathbf{T}_s} p_{b->a}^{(k)} \mathscr{P}_{\lambda_a^{(k)}}(r)], \qquad (1)$$

where $\mathbf{T}_s$ is the set of tuples $(a,b,r) \in \mathbf{A} \times \mathbf{A} \times \mathbf{N}$ corresponding to transitions from activity b to activity a with r repetitions of activity a. The goal of the algorithm is to find the parameters that maximize the likelihood.

In the first stage, the algorithm initialize randomly K profiles. Next, it iteratively improves the *likelihood*, by alternating two steps as described below. In each step it modifies the repartition or the Markov chain parameters.

Initialization: The initialization consists in choosing randomly either the $p_{b->a}^{(k)}$ and $\lambda_a^{(k)}$ or the $\mu_k^{(s)}$. In our algorithm, we start with the $\mu_k^{(s)}$. This can be done by generating a random number k^* from 1 to K for each student s and by setting

$$\mu_k^{(s)} = \begin{cases} 1 & \text{if } k = k^* \\ 0 & \text{otherwise.} \end{cases}$$

Iterations: The iteration phase has two steps. First, we compute the optimal values for $p_{b->a}^{(k)}$ and $\lambda_a^{(k)}$ given that $\mu_k^{(s)}$ are fixed (equations (2) and (3)).

$$p_{b->a}^{(k)} = \frac{\sum_{s\in\mathscr{S}} \sum_{(a,b,\_)\in\mathbf{T}_s} \mu_k^{(s)}}{\sum_{s\in\mathscr{S}} \sum_{(\_,b,\_)\in\mathbf{T}_s} \mu_k^{(s)}} \quad (2)$$

$$\lambda_a^{(k)} = \frac{\sum_{s\in\mathscr{S}} \sum_{(a,\_,r)\in\mathbf{T}_s} r\mu_k^{(s)}}{\sum_{s\in\mathscr{S}} \sum_{(a,\_,\_)\in\mathbf{T}_s} \mu_k^{(s)}} \quad (3)$$

Next, we compute the new values of $\mu_k^{(s)}$ according to the new $p_{b->a}^{(k)}$ and $\lambda_a^{(k)}$ (equations (4)).

$$\mu_k^{(s)} = \frac{\prod_{(a,b,r)\in\mathbf{T}_s} p_{b->a}^{(k)} \mathscr{P}_{\lambda_a^{(k)}}(r)}{\sum_{c=1}^{K} \prod_{(a,b,r)\in\mathbf{T}_s} p_{b->a}^{(c)} \mathscr{P}_{\lambda_a^{(c)}}(r)} \quad (4)$$

Intuitively, the in the first step we compute the parameters of the latent classes given the repartition of the students and in the second step we recompute the repartition from the new classes parameters.

5.2 Example: Interpretation clusters (K=3)

Before we present the results for the choice of the number of clusters, in this section, we illustrate the behaviour of the algorithm and the model when the number of clusters is small ($K = 3$). Although in this case we may lose important variability among groups of students, small number of clusters allows us to visualise the Semi-Markov models and interpret each of the clusters.

The visualizations of the Semi-Markov models on Figure 1 can reveal general characteristics of students' behaviours. For example, Profiles 1 and 3 are in general less active as they have more `EndOfDay` events. On the contrary, Profile 3 has a very high average number of repetition on `VideoPlay` and considerable probability to go back to `EndOfDay` events. This means that students of this cluster are not fully engaged in all MOOC activities.

A more insightful way to analyse and interpret the differences is to generate sequences of events and compare the outcomes. We can compute the expected number of videos watched or the expected number of post on the forum directly from simulated sequences. Table 2 shows the average number of several types of events for 100 simulated students (average from 10000 simulations) over four weeks generated with the three Markov models from Figure 1. For

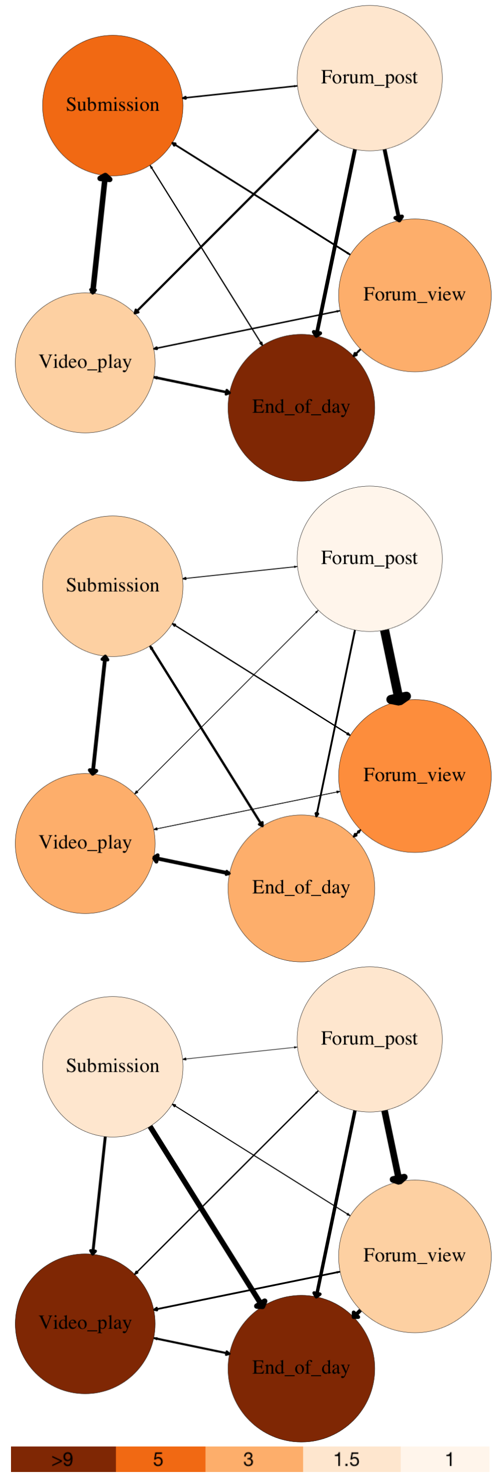

Figure 1: Three graphical representations of behaviour profiles extracted by the EM algorithm. From top to bottom: profiles 1, 2 and 3 (thickness: transition probability; color: average number of repetitions)

example, we can see that students of Profile 1 participate in the collaborative activities of the MOOC more rarely, but engage in the assignments more than in watching the videos. This might indicate that they already have a good understanding of the content of the course and do not need to spend more time on studying. To fully investigate this hypothesis, further analysis should be conducted.

Profiles	1	2	3
Watched Videos	1060	3133	2363
Submissions	1535	2423	442
Forum Visits	68	1711	255
Forum posts	3	96	15

Table 2: Average number of events for 100 students over the first four weeks of the MOOC

5.3 Choice of the parameter K

A common challenge of unsupervised learning and fitting a probabilistic model is finding the correct number of classes. In our case, the similarity of the algorithm with other clustering techniques such as the K-means leads to the "elbow heuristic", often used in practice. The idea is to choose the number of clusters large enough to explain a large part of the variability, but such that a greater number of clusters would not explain substantially more.

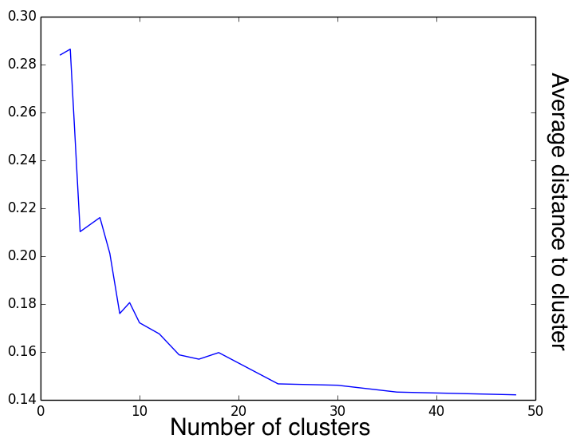

Figure 2: Average distance of students from their model for different number of classes

In order to confirm the result of this first measure of quality, we designed another measure described in the equation (5). The goal is to quantify how the students sequences diverge from their attributed cluster. In the equation, $|A|$ is the cardinality of the set of possible activities, $p_s(a)$ is the probability of finding the activity a if we take uniformly at random an activity of student s and $p_k(a)$ is the probability of finding the activity a if we take uniformly at random an activity from a sequence generated by the class k.

$$d^2(s,k) = \frac{1}{|A|} \sum_{a \in \mathbf{A}} (p_s(a) - p_k(a))^2 \quad (5)$$

This distance measure shows an elbow shape for the same values of K between 10 and 15 as it can be seen on Figure 2. We conclude that MOOC students from our dataset can be meaningfully clustered into $10-15$ different classes.

6. SIMULATIONS

With a model fitted with the EM algorithm at hand, the algorithm repartitioned students and chose parameters of a Semi-Markov Chain for each of the clusters. Since both the repartition and the Semi-Markov Chains are generative, we can draw samples from the fitted distribution, i.e. we can simulate the students. We run the simulations and show a possible way to measure the validity of the results.

To validate potential value of simulations, we first propose a simple accuracy measure. In equation (6), $P_{real}(|a| > n)$ represents the probability that a student performs more than n events of type a during the time of the MOOC. $|a|$ is the count of events of type a. $P_{sim}(|a| > n)$ represents the same probability but for a simulated student. In the measure we chose the value $N = 50$ because it covers most of the variability in the students activity sequences and is not too large as still 19% of the students have an activity with more than 50 repetitions.

$$MSE = \frac{1}{(|A|-1)*N} \sum_{a \in \mathbf{A}} \sum_{n<N} (P_{real}(|a|>n) - P_{sim}(|a|>n))^2 \quad (6)$$

In order to prove the correctness of the modeling method, we divided our dataset into a training set and a test set for validating the results. The first step is to run the algorithm on the training set with several parameter K and then, use the computed parameters to simulate a new population of students and finally compare this population with the students from the testing set. In Figure 3 we can see that the fit does not improve much after $K = 15$, because too high number of clusters makes the algorithm learn mostly the noise from the random actions of the students instead of their real intrinsic behavioural patterns.

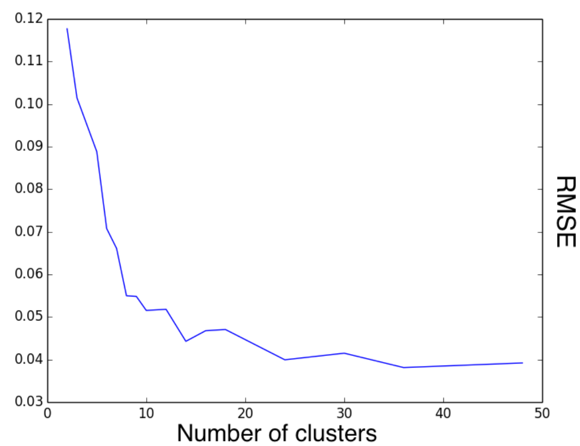

Figure 3: Measure of accuracy of a simulation for different number of classes

The small error proves that the distribution obtained from simulations is close to the original distribution. This implies that the model properly trained on small sample of students or on just few first events, can be extrapolated by simulation to further events or larger samples.

In an experimental setup, simulations with varying initial conditions of the model (e.x. probabilities of transitions) can give us distributions of events at the later state. Knowing probability distributions of the results of two conditions allows to estimate sample sizes needed for finding statistical evidence of the investigated effect.

7. DISCUSSION

In Section 5 we showed that Semi-Markov chains can be successfully applied to describe behavioural patterns of students **(RQ1)**. In Section 5.2, a simple study with reduced number of clusters prove their potential interpretability **(RQ2)**. In Section 6, we discuss how these models can be used to infer distributions of events **(RQ3)**.

Our method has two main limitations. They can be further relaxed with additional data or with incorporation of domain knowledge.

The Homogeneity of the Markov process: The Markov assumption was introduced for reducing the number of parameters of our model. It is a strong simplification, which entails some drawbacks. This assumption implicitly requires that student behave with exactly the same transition matrix during the whole course. The motivation to keep learning should increase when getting closer to the end of the course and thus the dropout rate decreases, which cannot be capture by our method. A good way to overcome this weakness is to use inhomogeneous Markov models with transitions probabilities that are functions of time.

Differences between courses: The quality of the videos, the level of difficulty of the assignments or the discussion topics in the forums are all factors that can greatly influence the behaviour of a student. None of these were included in our model. We hypothesize that adding external annotations that would impact the transition probabilities of our Markov models could help solve this problem. As for now, our model can be used to compare courses. For example, if we run the algorithm on two MOOCs and realise that the Video Watchers of one course have a lower engagement, that shows a lower quality of video content while differences for the Forum Follower may reveal differences on the quality of the Forum discussions.

8. REFERENCES

[1] Girish Balakrishnan and Derrick Coetzee. Predicting student retention in massive open online courses using hidden markov models. *Electrical Engineering and Computer Sciences University of California at Berkeley*, 2013.

[2] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(suppl 3):7280–7287, 2002.

[3] Christopher G Brinton, Mung Chiang, Sonal Jain, HK Lam, Zhenming Liu, and Felix Ming Fai Wong. Learning about social learning in moocs: From statistical analysis to generative model. *Learning Technologies, IEEE Transactions on*, 7(4):346–359, 2014.

[4] Ed H Chi, Peter Pirolli, Kim Chen, and James Pitkow. Using information scent to model user information needs and actions and the web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 490–497. ACM, 2001.

[5] Carleton Coffrin, Linda Corrin, Paula de Barba, and Gregor Kennedy. Visualizing patterns of student engagement and performance in moocs. In *Proceedings of the fourth international conference on learning analytics and knowledge*, pages 83–92. ACM, 2014.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977.

[7] Andrew J Elliot and Todd M Thrash. Approach-avoidance motivation in personality: approach and avoidance temperaments and goals. *Journal of personality and social psychology*, 82(5):804, 2002.

[8] José P González-Brenes and Yun Huang. Using data from real and simulated learners to evaluate adaptive tutoring systems. In *Proceedings of the Workshops at the 18th International Conference on Artificial Intelligence in Education AIED*, 2015.

[9] James A Gordon, William M Wilkerson, David Williamson Shaffer, and Elizabeth G Armstrong. "practicing" medicine without risk: students' and educators' responses to high-fidelity patient simulation. *Academic Medicine*, 76(5):469–472, 2001.

[10] René F Kizilcec, Chris Piech, and Emily Schneider. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses. In *Proceedings of the third international conference on learning analytics and knowledge*, pages 170–179. ACM, 2013.

[11] Kenneth R Koedinger, Noboru Matsuda, Christopher J MacLellan, and Elizabeth A McLaughlin. Methods for evaluating simulated learners: Examples from simstudent. *17th International Conference on Artificial Intelligence in Education AIED*, 5:45–54, 2015.

[12] Nan Li, Łukasz Kidziński, Patrick Jermann, and Pierre Dillenbourg. How do in-video interactions reflect perceived video difficulty? In *Proceedings of the European MOOCs Stakeholder Summit 2015*, number EPFL-CONF-207968, pages 112–121. PAU Education, 2015.

[13] Ran Liu and Kenneth R Koedinger. Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities. In *Educational Data Mining 2015*. EDM, 2015.

[14] Gord McCalla and John Champaign. Aied 2013 simulated learners workshop. In *Artificial Intelligence in Education*, pages 954–955. Springer, 2013.

[15] Richard Nock and Frank Nielsen. On weighting clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(8):1223–1235, 2006.

[16] Mirko Raca, Łukasz Kidziński, and Pierre Dillenbourg. Translating head motion into attention-towards processing of student's body-language. In *Proceedings of the 8th International Conference on Educational Data Mining*, number EPFL-CONF-207803, 2015.

[17] Arti Ramesh, Dan Goldwasser, Bert Huang, Hal Daumé III, and Lise Getoor. Modeling learner engagement in moocs using probabilistic soft logic. In *NIPS Workshop on Data Driven Education*, 2013.

[18] Ipke Wachsmuth and Jens-Holger Lorenz. Sharpening one's diagnostic skill by simulating students' error behaviors. *Focus on learning problems in mathematics*, 9(2), 1987.

[19] Christopher A Wolters. Advancing achievement goal theory: Using goal structures and goal orientations to predict students' motivation, cognition, and achievement. *Journal of educational psychology*, 96(2):236, 2004.