Modeling MOOC Student Behavior With Two-Layer Hidden Markov Models

Chase Geigle
Department of Computer Science
University of Illinois (UIUC)
Urbana, Illinois, USA
geigle1@illinois.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois (UIUC)
Urbana, Illinois, USA
czhai@illinois.edu

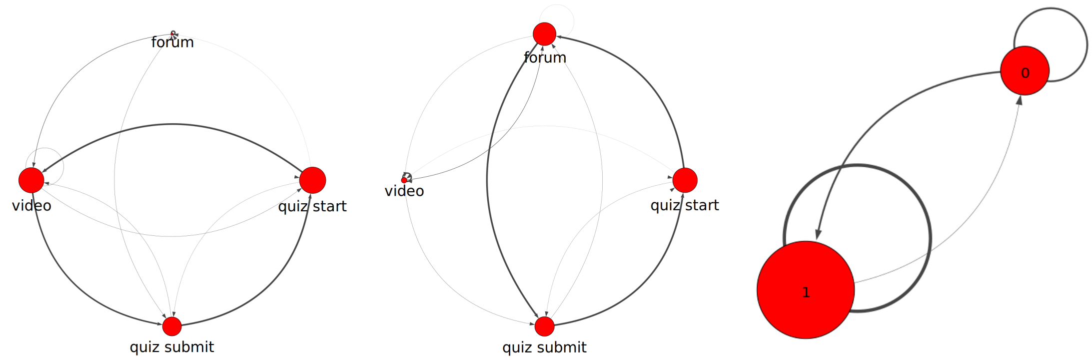
Massive open online courses (MOOCs) provide educators with an abundance of data describing how students interact with the platform, but this data is highly underutilized today. This is in part due to the lack of sophisticated tools to provide interpretable and actionable summaries of huge amounts of MOOC activity present in log data. To address this problem, we propose a student behavior representation method alongside a method for automatically discovering those student behavior patterns by leveraging the click log data that can be obtained from the MOOC platform itself. Specifically, we propose the use of a two-layer hidden Markov model (2L-HMM) to extract our desired behavior representation, and show that patterns extracted by such a 2L-HMM are interpretable and meaningful. We demonstrate that the proposed 2L-HMM can also be used to extract latent features from student behavioral data that correlate with educational outcomes.

1. Introduction

The proliferation of massive open online courses (MOOCs) has resulted in a profound impact on education. As more and more students participate in these novel educational environments, it is of utmost importance that we be able to understand the behavioral patterns students exhibit in these environments. While we can easily observe the changes in behavior of students in real classrooms, MOOC environments present some significant challenges in this regard: the structure of the course itself is more hands-off in nature than that of the traditional classroom (in most cases), and thus attracts more students that are full-time workers with irregular learning schedules.

At the same time, this influx of learners turning to MOOC platforms to educate themselves directly leads to the collection of larger datasets of behavioral data through the platform's logging. This presents a unique opportunity: the data present in these logs has the power to aid us in understanding the behavior of students who take our MOOCs. However, due to the vast scale of these behavioral logs, student behavior patterns are mostly undetectable for MOOC instructors and as a result, the rich data available through MOOC logs is highly underutilized today.

What stands in the way? Instructors require intelligent systems to create concise and digestible summaries of the massive amount of interaction data collected. If we can understand how users are interacting with our MOOCs, we are much more likely to be able to make changes to these courses that positively impact learners. We view this paper as attempting to bridge this gap.



(a) Behavior pattern 0

(b) Behavior pattern 1

(c) Transitions between the two behavior patterns to the left.

Figure 1: An idealized example of what our behavior representation could capture.

How should we represent behavioral patterns, and what does it mean to understand changes in student behavior with respect to these patterns? These are still very open questions and are active areas of research (Kizilcec et al., 2013; Faucon et al., 2016; Davis et al., 2016; Shih et al., 2010). In this paper, we advocate for a representation of student behavior patterns as well as *behavior transitions* that we believe is simultaneously interpretable but also amenable to unsupervised, automated discovery via statistical means. Specifically, we choose to visualize behavior patterns as labeled directed graphs where a node represents a "behavior state" (such as watching a lecture video or visiting a forum), a directed edge indicates a transition from one behavior state to another, node sizes are proportional to steady-state probabilities, and edge widths are proportional to the probability of leaving a node following that edge. We can use this same representation for visualizing both the student behavior patterns as well as the transitioning behavior between them. In Figure 1 we show a hypothetical example of the kind of output our proposed representation could convey. Here we see two different behavioral patterns (1a and 1b) as well as the *transition behavior* between these two behavior patterns (1c). Such a visualized state-transition representation is very informative for describing student behavior. Indeed, we could infer many things from even such a simple example: The first might be that, when students are taking quizzes, they tend to either use the forum or the videos for support, but not both. They also tend to take quizzes in a sort of "cycle" pattern, indicating perhaps that this course allows quiz re-takes. Finally, in Figure 1c we could conclude that users tend to change their quiz-taking behavior over time from one that is more video-focused (pattern 0) to one that is more forum-focused (pattern 1).

Our goal in this paper is to design a model that can automatically capture student behavior in this way via unsupervised learning methods applied to large collections of click logs associated with MOOCs. We view our model as a component of a system that enables collaboration between the machine and a human instructor to extract knowledge from large collections of MOOC data. Automatically extracting interpretable patterns from the clickstream data associated with MOOCs is a necessary step for instructors to identify the hidden knowledge in massive interaction datasets. Without the availability of a suitable model for identifying behavioral patterns, instructors are not empowered to use this available data to improve their courses without expending extraordinary amounts of manual effort (even with which the raw data can still be very hard to interpret).

Our proposed model (as well as the proposed behavior representation) is motivated by the following observations:

1. Student behavior is complicated and cannot necessarily be captured sufficiently by rule-based methods such as those explored by Kizilcec et al. (2013) and Davis et al. (2016). We instead propose to treat student behavior patterns as being characterized (represented) via a sequence of *latent states*. This allows us to automatically capture patterns that we might not have been able to articulate clearly a priori via a series of rules, and allows us to model the inherent uncertainty in assigning a student's behavior to a pattern or group.

2. Student behavior can vary over time. Previous models that treat students as exhibiting only one behavioral pattern over time (Faucon et al., 2016) miss out on the opportunity to understand student behavior dynamics in a course. We propose a latent space model with *latent state transitions* to flexibly model the dynamics.

3. Analysis of student behavior can and should be performed at varying levels of granularity. This requires us to aggregate data over time with *different levels of resolution*; existing models tend to come with an assumption about the resolution of time they consider (Faucon et al., 2016; Kizilcec et al., 2013; Shih et al., 2010). We propose a model that is agnostic to the time resolution considered, allowing it to be applied at different levels of resolution more naturally.

Thus, what we propose is a *latent variable approach* for mining student behavior patterns that is *probabilistic* for inference and does *not* force assumptions about time resolutions, making it *flexible to model state changes over different time resolutions* more easily. More specifically, we propose a novel two-layer hidden Markov model (2L-HMM) to discover latent student behavior patterns via unsupervised learning on large collections of student behavior observation sequences. Evaluation results on a MOOC data set on Coursera demonstrate that the 2L-HMM can effectively discover a variety of interesting interpretable student behavior patterns at different levels of resolution, many of which are beyond what existing approaches can discover. We show that the patterns uncovered by the 2L-HMM capture meaningful behavior by quantitatively showing that features extracted from a trained 2L-HMM correlate with learning outcomes. Since our proposed methods are unsupervised, they can potentially be applied to any MOOC data without requiring manual annotation effort at the level of sequences, thus empowering instructors to use the latent patterns discovered by the 2L-HMM to further extract knowledge about the behaviors his/her students exhibit in the MOOC.

2. Related Work

Our model is based heavily on the prior art of Hidden Markov Models (HMMs) (Rabiner, 1990) for sequence labeling tasks. As a member of the more general family of probabilistic graphical models (Koller and Friedman, 2009), HMMs are widely applicable and have been used for tasks such as speech recognition (Huang et al., 1990), part-of-speech tagging (Jurafsky and Martin, 2009), and econometrics (Hamilton, 1990). A major challenge in applying HMMs and other graphical models successfully to solve a problem is to design an appropriate architecture of the model, which always varies according to specific applications.

For example, in part-of-speech tagging (Jurafsky and Martin, 2009), the output distributions are categorical (distributions over words from a fixed vocabulary) and the latent states represent the part-of-speech category for a word. In speech recognition (Huang et al., 1990), the output distributions might be mixtures of Gaussians to predict real-valued vectors extracted from short windows of a speech signal. In the domain of econometrics, Hamilton (1990) explores HMMs in the context of "regime-switching." In this framing, the goal is to understand how econometric data changes by modeling discrete changes in "regime" as having an impact on the resulting real-valued vector data observed. The "regimes" are thus represented with a model that can produce real-valued vector data, such as a multivariate Gaussian or an auto-regressive model. The analogy with HMMs is that a "regime" is a latent state, and the characterization of the regime itself is the output distribution for that latent state. Our model can be seen as such a "regime-switching" model where the output of the "regimes" that students are switching between are discrete-valued *sequences* (as opposed to real numbers, vectors of real numbers, or categorical symbols) and the model used to represent a specific "regime" is an (observable) Markov chain over the observed student actions. We view the switching between "regimes" as the first "layer" of our model, and the transitioning behavior *within* a "regime" between the actions a student takes as the second "layer" of our model.

A multi-layered approach to HMM modeling of sequence data has been performed before in other domains. Zhang et al. (2004), for example, explored a two-layer HMM framework for modeling actions in meetings, but their definition of "two-layer" differs from ours. In their formulation, the "lower-layer" level is used to label audio-video action sequences into basic events, and the "upper-layer" is used to label the output of the lower-layer to discover higher-level office behavior abstractions. Oliver et al. (2004) propose a similar layered HMM approach for modeling office activity at multiple different levels of time granularity. In their approach, each layer L is represented as a "bank" of K different HMMs that model sequences of some length T_L. At the bottom layer ($L = 1$), the bank of K HMMs corresponding to that layer is run on some initial observation data, considering windows of observations of length T_1. Then an output is generated by using the inferential results of these K HMMs to make a prediction: which of the K HMMs was most likely to produce that sequence of observations? This output is then fed to the next layer of HMMs, which considers sequences of length T_2 and outputs prediction results as to which of the K HMMs at layer two were most likely to produce the sequence of outputs produced by the previous layer, and so on.

Our formulation differs from both Zhang et al. (2004) and Oliver et al. (2004) in that we do not feed the labeled sequence of the lower level into the input of the higher level. Instead, our lower level is treated using a non-hidden Markov model, and the higher level is modeling transitions between the K different non-hidden Markov models we consider. The problem to be solved is similar in that we wish to predict a "label" for a sequence of actions a student takes as well as understand the transition behavior between those labels. However, one of the consequences of modeling the lower layer using a non-hidden Markov model instead of an HMM directly is that the meaning of the K different latent states can be more clearly captured by using our proposed behavior representation. If we were to use an HMM as our first layer, the behavior patterns (like from Figure 1) would instead have nodes that represent another set of latent states instead of being concrete actions themselves. Each of these latent states would then be associated with some other output distribution over the possible concrete actions to be considered. In order to understand a single behavior pattern uncovered, one would first have to understand the different output distributions for the latent states in that pattern to understand the meaning of that latent state. This further complicates the understanding of higher level patterns because understanding the higher layer patterns requires understanding the lower layer patterns.

By taking a more restrictive view of the first layer, we can produce a representation that can be more readily interpreted due to the states in our first layer representation having an immediately clear meaning (the concrete action they represent).

The 2L-HMM model we propose is more closely related to the Hierarchical Hidden Markov Model (HHMM) detailed in Fine et al. (1998). Here, the "layers" are modeled by having the hidden Markov model have two kinds of transitions. Horizontal transitions move between states within a layer, where vertical transitions move between different layers. At the bottom layer lie the "production" states, which output symbols according to some probability distribution. Our specific model, in this case, can be modeled as an HHMM where the horizontal transitions between nodes at the highest layer (including self-loops) *must* be immediately followed by a vertical transition to the lower layer. The output probability distributions over symbols in the lower layer "production" level are forced to emit only one kind of symbol, and vertical transitions are only allowed into the original higher-layer state that transitioned down into the lower-layer.

Mixtures of hidden Markov models are also conceptually similar to our formulation. Song et al. (2009) explored using a mixture of hidden Markov models in the context of anomaly detection in the security domain. Ypma and Heskes (2002) use mixtures of HMMs to categorize web pages and cluster users by investigating web log data, which is quite similar to the clickstream log data we obtain from MOOCs. The major difference between our approach and a standard mixture of HMMs is that we also model the *transition behavior* between the Markov models that make up our model's lowest layer, where a standard mixture of HMMs would ignore the potential dependence of the previous sequence's latent state on the next sequence's latent state.

HMMs or similar ideas have been previously applied to model education data (Shih et al., 2010; Kizilcec et al., 2013; Davis et al., 2016), but the previous models are not well tuned toward the student behavior task and thus cannot adequately address all the aspects of the complexity of student learning behavior. A main technical contribution of this paper is to propose a more general HMM that can better adapt to the variations of student behavior via its variable resolution and nested HMM structure, and thus enable discovery of more sophisticated behavior patterns and provide a more detailed characterization of student behavior than the previous models.

For example, Kizilcec et al. (2013) assigned students to states following a rule-based approach based upon when the student submitted the assignment for a particular week in the course. They investigated how students transitioned between these states as the course progressed, and used the sequence of states a student exhibited as a representation for performing k-means clustering of students into related groups. This differs from our method substantially: we assign students to states using a probabilistic framework to account for uncertainty in this state assignment and jointly learn *representations* for these states, which are treated as being *latent* as opposed to pre-defined using some rule (or set of rules). Furthermore, our model provides more flexibility in how the time segments are defined, allowing for both finer (for example, day-by-day) or coarser (for example, month-by-month) granularity. Shih et al. (2010) investigated the use of HMM-based clustering techniques for automatic discovery of student learning strategies when solving a problem. This is similar to our approach in that the description of behavior profiles is a Markov model, but cannot further characterize each latent state with another informative HMM. Thus, their work can be regarded to modeling "micro" behavior, whereas our model can model both "micro" and "macro" behavior.

Davis et al. (2016) investigate frequent student behavior pattern chains with a set of actions that are defined similarly to ours. However, their method for finding the common behavioral patterns involves a manual clustering step to identify behavioral "motifs," which is then followed by an automatic (rule-based) assignment of all sequences to these motifs. Our method, by contrast, attempts to do this automatically: the latent state representations obtained by our model attempt to capture similar meanings to their behavior motifs in a completely automated fashion. They also automatically generate and investigate Markov models for different MOOCs, but do so by considering *all* student action sequences as belonging to a *single Markov model*. In our approach, we allow each student behavior sequence to belong to one of K different Markov models (and further model the transition probabilities between these latent state Markov models between each sequence a student generates). Thus, their Markov models presented are a special case of our model when $K = 1$.

Faucon et al. (2016) proposed a semi-Markov model for simulating MOOC students. They produce behavior profiles that characterize groups of students in the form of semi-Markov models like our proposed model does, but they assume that a student can belong to only one behavior profile across the entire course rather than allowing this profile to change over time. Because we do not have this restriction, our model is also able to learn the transition probabilities between the different behavior profiles we discover.

There are a few additional related studies worth mentioning. Bayesian Knowledge Tracing (Corbett and Anderson, 1994) in its basic form uses a hidden Markov model to model the probability that a learner knows a certain skill at a given time. Modifications to this algorithm include contextual estimation of the "slip" and "guess" probabilities of the model (Baker et al., 2008) and most recently a re-framing as a neural network problem (Piech et al., 2015).

3. A Two-Layer HMM for MOOC Log Analysis

3.1. Basic idea and rationale

Our general idea is to use a probabilistic generative model to model the student activities as recorded in a MOOC log, which means we will assume that all the observed student activities are samples drawn (i.e., "generated") from a parameterized probabilistic model. We can then estimate the parameter values of the probabilistic model by fitting the model to a specific MOOC log data set. The estimated parameter values could then be treated as the latent "knowledge" discovered from the data. Because such a generative model attempts to fit *all* the data, it enables us to discover interesting patterns that can explain the *overall* behavior of a student or the *common* behavior patterns shared by many students.

An HMM is a specific probabilistic generative model with a "built-in" state transition system that would control the data to be generated by the model, thus it is especially suitable for modeling sequence data (Rabiner, 1990; Huang et al., 1990). At any moment, the HMM would be in one of k states $U = \{u_1, \ldots, u_k\}$, and at the next moment, the HMM would move to another state stochastically according to a transition matrix that specifies the probability $p(u_i \mid u_j)$ of going to state u_i when the HMM is currently in state u_j. When the HMM is in state u, the HMM can generate an observable data point x according to an output probabilistic model $p(x \mid u)$. Thus, if we "run" an HMM for N time points denoted by $t = 1, \ldots, N$, the HMM could "generate" a sequence of observations $x_1 \ldots x_N$, where each x_i is an output symbol by going through a sequence of *hidden states* $w_1 \ldots w_N$ where each w_i is a random variable taking a value from the HMM's state set U. The association of such a latent sequence of state transitions with the observed symbols makes it possible to use the HMM to "decode" the latent behavior of students behind the surface behavior we directly observe in the log data, allowing for understanding student behavior more deeply than a model with no latent state variables.

In many ways, the generation process behind an HMM is meant to simulate the actual behavior of a student. We may say that students transition through different "task states" (or "behavior states") in the process of study. One such task state may be to learn about a topic by mostly watching lecture videos, another task state may be to work on quizzes, and yet another may be to participate in forum discussions. While in each of these different states, the student would tend to exhibit different surface "micro" behaviors. For example, in the lecture study state, the student would tend to have many video-watching related behaviors and occasionally forum activities, while in the quiz-taking state (to pass each module), the student would tend to show many quiz-related "micro" activities as well as asking questions or checking discussions on the forum. Note that due to the complexity of the student behavior, it is very difficult to accurately *prescribe* the specific surface "micro" behavior patterns for each state in advance, especially without prior knowledge about the students. For example, forum activities are likely interleaved with other activities in every task state and the interleaving pattern can be somewhat irregular with potentially many variations. The major motivations for using an HMM are that (1) it uses a probabilistic model (the output probability distribution $p(x \mid u)$ conditioned on each state) to directly capture the inevitable uncertainty in the association of surface "micro" activities with their corresponding latent task/behavior state, which is often our main target to discover and characterize, and (2) it does not make any assumption about which latent task/behavior state must be associated with which observed activities or how a student would move from one state to another, but instead allows our data to "tell" us what kind of associations are most likely, what kind of transitions are most probable, and which states tend to be more long-lasting for any set of students.

However, if we were to use an ordinary HMM to analyze our data, we would treat each observed "micro" activity (such as video watching or forum post reading) as an output symbol, and thus the output distribution $p(x \mid u)$ for each discovered latent state would be a simple distribution over all kinds of observable micro activities recorded in our log data (e.g., 50% lecture watching, 8% quiz taking, 7% quiz submission, 2% course wiki reading, etc.). While such a distribution is meaningful and can already help us interpret the corresponding latent state, it only gives us a rather superficial characterization of student behavior.

Ideally, we want $p(x \mid u)$ to characterize the directly observable "micro" behavior in more detail to further capture the relations and dependencies of these "micro" activities. To this end, we would treat an *entire sequence* of "micro" activities (e.g., one session of activities) as an observed "symbol" from a latent state, and further model the generation of such a sequence with another Markov model where we treat each micro activity as an *observable* state, and model the transitions between these activity states in very much the same way as the state transitions in an HMM.

Adding this second layer would allow us to characterize a latent task state in much more detail, as it would reveal not only what activities are most common to a task state, but also the transition patterns between these "micro" activities (e.g., it can reveal frequent back-and-forth transitions between quiz-taking and quiz-submission, which would suggest a concentrated period of taking quizzes). Combining this "surface" Markov model over the "micro" actions with the "deep" hidden Markov model over the latent task states gives us a general and powerful two-layer HMM (2L-HMM) that can simultaneously learn "deeply" the latent task/behavior states and their transitions as well as the corresponding "micro" activity transition patterns associated with each latent state to facilitate interpretation and analysis of the discovered latent state patterns.

To estimate the parameters of our model, we will use the EM algorithm (Dempster et al., 1977) which allows us to perform maximum likelihood estimation of the model in the presence of latent (unobserved) variables. This algorithm, intuitively, works as follows: first, the model parameters we wish to estimate are initialized to some random (but valid) starting point. Then, we "guess" what the latent variable values might be given the current model parameters. We can then use this guess to re-estimate the model parameters, which will improve their accuracy slightly. We can then use these newly estimated parameters to again "guess" what the latent variable values might be, and so on. We repeat this process until the parameter estimates no longer shift by much (or, equivalently, the log likelihood of the data does not improve by much). The computation for the "guessing" of latent variable values is somewhat involved in the case of HMMs (see Section 3.3. for the full details), but the general intuition behind the iterative hill-climbing remains valid.

Next, we present this model more formally and discuss how to estimate its parameters to uncover these latent patterns in an unsupervised manner.

3.2. Formal Definition of the 2L-HMM

Given a MOOC log, we can define a set $\mathbf{A}$ of actions that a student can take at any given time. For example, an action $a \in \mathbf{A}$ might be "viewing lecture," "taking quiz," or "viewing forum." For each student in the course $\ell \in \mathbf{L}$, we then extract a list of action sequences $\mathbf{O}_\ell$ that he or she produced as observed in the log, where each sequence $\mathbf{o} \in \mathbf{O}_\ell$ is itself a list of actions $(a_1, a_2, \ldots, a_T)$ with each $a_i \in \mathbf{A}$. Each sequence can be divided flexibly; in this paper, we chose to denote the end of a sequence as occurring when no further actions occur within a 10-hour window of time (and thus our sequences roughly correspond to one day's worth of activity). This decision reflects our desire to uncover latent state transition behavior at the granularity of day-to-day behavior. A different segmentation strategy could be used to uncover hour-by-hour behavior or week-by-week behavior, and this depends entirely on the desired time resolution one wishes to be exhibited in the transitions between the latent states. Different segmentation strategies will result in different underlying training data, and thus different meanings behind the patterns that the 2L-HMM will extract. The flexibility of using different segmentation strategies is intentional as it allows a user to adjust the segmentation as needed to obtain patterns at different granularity levels.

Each sequence $\mathbf{o} \in \mathbf{O}_\ell$ is associated with a latent state $u \in \{1, \ldots, K\}$ (where K is a fixed constant picked in advance). The actions within the sequence $\mathbf{o} = (a_1, a_2, \ldots, a_T)$ are then modeled as a first-order Markov process conditioned upon u where each action is drawn from a distribution conditioned upon the previous action (except for the first which is sampled from an initial starting distribution). We can write the parameters for the first-order Markov model associated with latent state u as $\lambda^{(u)} = (\pi^{(u)}, A^{(u)})$ where $\pi^{(u)}$ indicates the initial probability vector of length $|\mathbf{A}|$ and $A^{(u)}$ is an $|\mathbf{A}| \times |\mathbf{A}|$ matrix indicating the transition probabilities between each pair of actions from $\mathbf{A}$.

Thus, the probability of a sequence $\mathbf{o}$ of length T given a latent state u is

$$P(\mathbf{o} \mid \lambda^{(u)}) = P(a_1 \mid \pi^{(u)}) \prod_{i=2}^{T} P(a_i \mid a_{i-1}, A^{(u)}) \tag{1}$$

where $P(a \mid \pi^{(u)}) = \pi_a^{(u)}$ is the probability of starting with action a and $P(a_i \mid a_{i-1}, A^{(u)}) =$

$A^{(u)}_{a_{i-1},a_i}$ is the transition probability of moving from action a_{i-1} to a_i given that the model is currently in latent state u.

We can compute the likelihood of a list of action sequences $\mathbf{O}_\ell$ of length N for a student ℓ by marginalizing over all possible latent state sequences $(v_1, \ldots, v_N) \in U$ as

$$P(\mathbf{O}_\ell \mid \Lambda) = \sum_{(v_1,\ldots,v_N)\in U} \left(P(v_1 \mid \Lambda) P(\mathbf{o}_1 \mid \lambda^{(v_1)}) \times \prod_{i=2}^{N} P(v_i \mid v_{i-1}, \Lambda) P(\mathbf{o}_i \mid \lambda^{(v_i)}) \right) \quad (2)$$

where Λ is the set of all model parameters. In our model, we let $\Lambda = (\pi, A, B)$ where π and A are the parameters of a first-order Markov model over the latent states and $B = (\lambda^{(1)}, \ldots, \lambda^{(K)})$ where each $\lambda^{(i)}$ consists of the parameters for the first-order Markov model over action sequences for latent state i. Thus π (without superscripts) is an initial probability vector of length K and A (without superscripts) is a $K \times K$ transition probability matrix, analogous to the case with the individual first-order Markov model parameters $\lambda^{(i)}$ for each latent state. We have in total $K + K^2 + K(|\mathbf{A}| + |\mathbf{A}|^2)$ parameters to be estimated from our sequence data.

This can be seen as a modification of the traditional hidden Markov model with categorical outputs (Rabiner, 1990) where instead of discrete observations (one for each latent state transition) we have observations that take the form of entire sequences $\mathbf{o}_i = (a_1, a_2, \ldots a_T)$ whose probabilities are computed using another (non-hidden) Markov model conditioned upon the latent state u_i.

3.3. Parameter Estimation

To learn the parameters of our model, we may use maximum likelihood estimation. Unfortunately, a closed-form solution does not exist, so we must appeal to the EM algorithm (Dempster et al., 1977). In particular, we propose a minor modification of the Baum-Welch algorithm (Rabiner, 1990) which is an efficient EM algorithm for learning the parameters of hidden Markov models in an unsupervised setting where the Markovian assumption is exploited to significantly reduce the computational complexity of the EM algorithm by avoiding explicit enumeration of all the possible state transitions. In the following sections, we will provide a brief description of the original Baum-Welch algorithm for unsupervised parameter estimation for categorical valued hidden Markov models, and then describe our modification to allow for parameter estimation for our 2L-HMM modification for sequence valued observations.

3.3.1. Baum-Welch for Traditional HMMs

In the traditional HMM formulation with categorical outputs, we have $\Lambda = (\pi, A, B)$ where π is the initial probability distribution over the latent states (of length K), A is a $K \times K$ matrix indicating the latent state transition probabilities, and $B = (b_1, \ldots, b_k)$ is a length K vector where each entry b_i is a probability distribution over the possible discrete output symbols from $\mathbf{A}$.

The goal of the Baum-Welch algorithm (also called the forward-backward algorithm) is to learn the values for the parameters Λ from a collection of observed sequences of values from $\mathbf{A}$. Concretely, our training data $D = (\mathbf{o}^{(1)}, \ldots, \mathbf{o}^{(M)})$ is a collection of M sequences $\mathbf{o}^{(k)}$, each of which consists of T_k symbols (observations) from $\mathbf{A}$. The Baum-Welch algorithm defines two sets of variables $\alpha_t^{(\mathbf{o})}(i)$ called the forward variables and $\beta_t^{(\mathbf{o})}(i)$ called the backward variables (Rabiner, 1990) for each sequence $\mathbf{o} \in \mathbf{D}$.

$\alpha_t^{(\mathbf{o})}(i) = P(o_1, \ldots, o_t, v_t = i \mid \Lambda)$ is the probability of generating the sequence of observations $(o_1, o_2, \ldots, o_t)$ up to time t and arriving in state i at that time. They are typically defined using the following recursion:

- $\alpha_1^{(\mathbf{o})}(i) = \pi_i b_i(o_1)$, the probability of starting in state i (π_i) times the probability of generating the first observation o_1 from state i.
- $\alpha_{t+1}^{(\mathbf{o})}(i) = b_i(o_{t+1}) \sum_{j=1}^{K} \alpha_t^{(\mathbf{o})}(j) A_{ji}$, the probability of generating the observation o_{t+1} from state i times the probability that we arrive in state i from any other previous state after generating all of the other observations.

Analogously, $\beta_t^{(\mathbf{o})}(i) = P(o_{t+1}, \ldots, o_T \mid v_t = i, \Lambda)$ is the probability of generating the *rest of the sequence* given that we are in state i at time t. They are also defined using a recursion:

- $\beta_T^{(\mathbf{o})}(i) = 1$
- $\beta_t^{(\mathbf{o})}(i) = \sum_{j=1}^{K} \beta_{t+1}^{(\mathbf{o})}(j) A_{ij} b_j(o_{t+1})$, the probability of transitioning to any state j and generating the observation o_{t+1} times the probability of generating the rest of the sequence given that we transitioned to state j.

Given the αs and the βs, we can compute $\gamma_t^{(\mathbf{o})}(i)$, the posterior probability of being in a given state i at time t, and $\xi_t^{(\mathbf{o})}(i,j)$, the posterior probability of going through a transition from state i to state j at time t as

$$\gamma_t^{(\mathbf{o})}(i) = \frac{\alpha_t^{(\mathbf{o})}(i)\beta_t^{(\mathbf{o})}(i)}{\sum_{j=1}^{K} \alpha_t^{(\mathbf{o})}(j)\beta_t^{(\mathbf{o})}(j)} \tag{3}$$

and

$$\xi_t^{(\mathbf{o})}(i,j) = \frac{\alpha_t^{(\mathbf{o})}(i) A_{ij} b_j(o_{t+1}) \beta_{t+1}^{(\mathbf{o})}(j)}{\sum_{j=1}^{K} \alpha_t^{(\mathbf{o})}(j)\beta_t^{(\mathbf{o})}(j)} \tag{4}$$

respectively (Rabiner, 1990). Computing these variables for each sequence $\mathbf{o} \in D$ is the E-step of the EM algorithm.

Given $\gamma_t^{(\mathbf{o})}(i)$ and $\xi_t^{(\mathbf{o})}(i,j)$ for each sequence $\mathbf{o} \in \mathbf{D}$, we can update our model parameters Λ as

$$\pi_i = \frac{\sum_{\mathbf{o} \in \mathbf{D}} \gamma_1^{(\mathbf{o})}(i)}{|\mathbf{D}|}, \tag{5}$$

$$A_{ij} = \frac{\sum_{\mathbf{o} \in \mathbf{D}} \sum_{t=1}^{T} \xi_t^{(\mathbf{o})}(i,j)}{\sum_{\mathbf{o} \in \mathbf{D}} \sum_{j=1}^{N} \sum_{t=1}^{T} \xi_t^{(\mathbf{o})}(i,j)}, \text{ and} \tag{6}$$

$$b_i(a) = \frac{\sum_{\mathbf{o} \in \mathbf{D}} \sum_{t=1, o_t=a}^{T} \gamma_t^{(\mathbf{o})}(i)}{\sum_{\mathbf{o} \in \mathbf{D}} \sum_{t=1}^{T} \gamma_t^{(\mathbf{o})}(i)} \tag{7}$$

respectively (Rabiner, 1990). This is the M-step of the EM algorithm.

3.3.2. Baum-Welch for HMMs with Sequence Observations

The major deviation of our model from the traditional categorical-valued HMM is that our observations are themselves sequences of actions from $\mathbf{A}$ rather than individual tokens. We again denote our parameters as $\Lambda = (\pi, A, B)$ where π is the initial probability distribution over the latent states (of length K), A is a $K \times K$ matrix indicating the latent state transition probabilities, but $B = (\lambda^{(1)}, \ldots, \lambda^{(K)})$ is now a vector of length K where each element $\lambda^{(i)}$ consists of the parameters for a first-order Markov model over the action space $\mathbf{A}$. Recall that each $\lambda^{(i)} = (\pi^{(i)}, A^{(i)})$ where $\pi^{(i)}$ is an initial action distribution over the $|\mathbf{A}|$ available actions, and $A^{(i)}$ is the $|\mathbf{A}| \times |\mathbf{A}|$ transition matrix between those actions.

The goal of the Baum-Welch algorithm is still to learn the parameters Λ for our model. What differs from the categorical-valued HMM case is that our data now consists of a collection of *lists* of sequences, rather than just a collection of sequences. This means that our observation values $\mathbf{o}_t$ are now *themselves* sequences of values from $\mathbf{A}$, rather than just being a single token from $\mathbf{A}$. Formally, our training data $\mathbf{D} = (\mathbf{O}^{(1)}, \ldots, \mathbf{O}^{(|\mathbf{L}|)})$, where each $\mathbf{O}^{(\ell)} = (\mathbf{o}_1, \ldots, \mathbf{o}_{T_\ell})$ is itself a list of sequences. Each sequence $\mathbf{o}_k \in \mathbf{O}^{(\ell)}$ consists of a list of actions $(a_1, \ldots, a_{T_k})$, each of which is a member of $\mathbf{A}$.

In order to run the forward-backward algorithm for an element $\mathbf{O} \in \mathbf{D}$ to compute the α and β recursions like before, we must define $b_i(\mathbf{o}_t)$ in this setting where $\mathbf{o}_t = (a_1, \ldots, a_{T_t})$ is now a sequence instead of a single token. We define it as follows:

$$b_i(\mathbf{o}_t) = P(\mathbf{o}_t \mid \lambda^{(i)}) = P(a_1 \mid \pi^{(i)}) \prod_{k=2}^{T_t} P(a_k \mid a_{k-1}, A^{(i)}) = \pi_{a_1}^{(i)} \prod_{k=2}^{T_t} A_{a_{k-1},a_k}^{(i)}. \tag{8}$$

Fortunately, the recursions for $\alpha_t^{(\mathbf{O})}(i)$ and $\beta_t^{(\mathbf{O})}(i)$ remain the same, as do the definitions of $\gamma_t^{(\mathbf{O})}(i)$ and $\xi_t^{(\mathbf{O})}(i,j)$, in the E-step. We can simply substitute the new definition for the output distribution $b_i(\mathbf{o}_t)$ in those equations.

The substantial change is in the updating equations in the M-step, where we replace the update for $b_i(a)$ (which used to be a categorical distribution) by a pair of updates for the Markov chain for state i: one for $\pi_a^{(i)}$ one for $A_{ab}^{(i)}$. The two updates can be written as

$$\pi_a^{(i)} = \frac{\sum_{\mathbf{O} \in \mathbf{D}} \sum_{\mathbf{o}_t \in \mathbf{O}, \mathbf{o}_{t,1} = a} \gamma_t^{(\mathbf{O})}(i)}{\sum_{\mathbf{O} \in \mathbf{D}} \sum_{\mathbf{o}_t \in \mathbf{O}} \gamma_t^{(\mathbf{O})}(i)}, \text{ and} \tag{9}$$

$$A_{ab}^{(i)} = \frac{\sum_{\mathbf{O} \in \mathbf{D}} \sum_{\mathbf{o}_t \in \mathbf{O}} \sum_{m=2, \mathbf{o}_{t,m-1} = a \wedge \mathbf{o}_{t,m} = b}^{|\mathbf{o}_t|} \gamma_t^{(\mathbf{O})}(i)}{\sum_{\mathbf{O} \in \mathbf{D}} \sum_{\mathbf{o}_t \in \mathbf{O}} \sum_{m=2, \mathbf{o}_{t,m-1} = a}^{|\mathbf{o}_t|} \gamma_t^{(\mathbf{O})}(i)}. \tag{10}$$

These updates can be understood as follows. $\pi_a^{(i)}$ is the probability that a sequence being generated by state i begins with action a. $\gamma_t^{(\mathbf{O})}(i)$ gives the probability of generating the sequence $\mathbf{o}_t$ from state i, so we simply aggregate this for all sequences $\mathbf{o}_t$ where the first action is a. We then normalize this distribution to sum to 1 across all possible actions $a \in \mathbf{A}$ to obtain our new estimate for $\pi_a^{(i)}$. $A_{ab}^{(i)}$ is the probability that a sequence being generated by state i currently at action $a \in \mathbf{A}$ transitions to action $b \in \mathbf{A}$. Thus, we compute the expected number of times we observe a transition from a to b in a sequence generated by state i, and we normalize this distribution to sum to 1 across all possible actions $b \in \mathbf{A}$. Our modified EM algorithm for 2L-HMMs is provided as part of the MeTA toolkit (Massung et al., 2016).

Table 1: Statistics about the sequences extracted from the two MOOCs.

MOOC	Students	Sequences	Avg. $|s|$
textretrieval-001	18,941	85,240	7.31
sustain-001	85,240	231,881	15.4

4. EXPERIMENT RESULTS

As an analysis tool, the 2L-HMM model provides us with the following two patterns to characterize student behavior: (1) the latent state representations, and (2) the latent state transitions. Thus, to evaluate the proposed model, we conduct experiments to qualitatively analyze both types of patterns discovered from empirical MOOC log data.

Specifically, we look at the MOOC logs associated with two different Coursera MOOCs offered by UIUC: textretrieval-001 and sustain-001. The textretrieval-001 MOOC represents a highly technical computer science course, where the sustain-001 MOOC is more representative of a humanities course. We picked these two MOOCs because of their vastly different content domains. Table 1 summarizes the two datasets we extracted from the MOOCs.

4.1. LATENT STATE REPRESENTATIONS

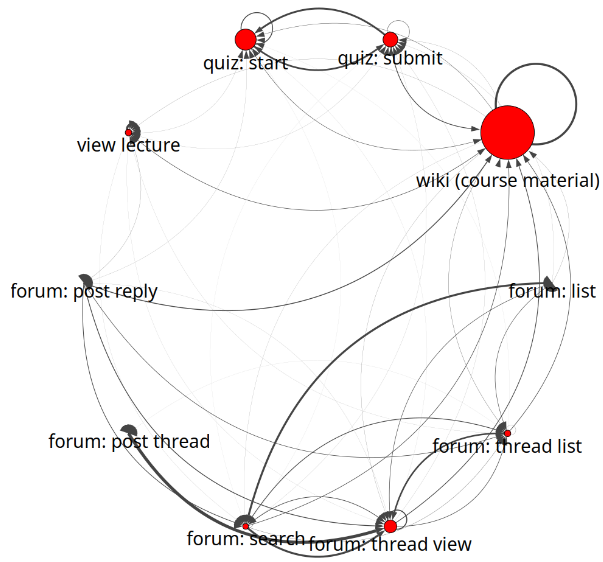
The 2L-HMM is meant to be a tool for exploratory analysis of student behavior. As such, the number of states should be empirically set based on the goal of analysis. A higher number of states will lead to a finer-grained modeling of student behavior. In our experiments, we explored using between 2–6 states. First, we fit a 6-state 2L-HMM to the textretrieval-001 sequence dataset and show some of the latent state representations we find. We used the following ten actions as our action set $\mathbf{A}$: (1) quiz start, (2) quiz submit, (3) wiki (course material), (4) forum list (view the list of all forums), (5) forum thread list (view the list of all threads in a specific forum), (6) forum thread view (view the list of posts within a specific thread), (7) forum search (a search query issued against the forum), (8) forum post thread (a new thread was posted), (9) forum post reply (a new post was created within an existing thread), and (10) view lecture (defined as either streaming or downloading a lecture video).

To visualize these Markov models that represent our latent states, we plot them as a directed graph where we set the size of a node to be proportional to its personalized Pagerank score (Page et al., 1999; Jeh and Widom, 2003) where the personalization vector is the initial state distribution for the Markov model. We let the thickness of a directed edge (u, v) reflect the probability of taking that edge given that a random walk is currently at note u (as indicated by the transition matrix)[1]. In the interest of reproducibility, the source code for analyzing the MOOC logs we use in this paper and for producing the figures themselves is publicly available as open-source software[2].

Figure 2 includes two such representations we learned. Under our interpretation, the first corresponds to a "quiz taking" state (it has higher "quiz: start" and "quiz: submit" state probabili-

[1]We do not plot the transition probabilities directly within the figure to ease readability; we instead will mention relevant transition probabilities in the text as we discuss the plots. The plots were created using python-igraph: `http://igraph.org/python/`.

[2]`https://github.com/skystrife/clickstream-hmm`



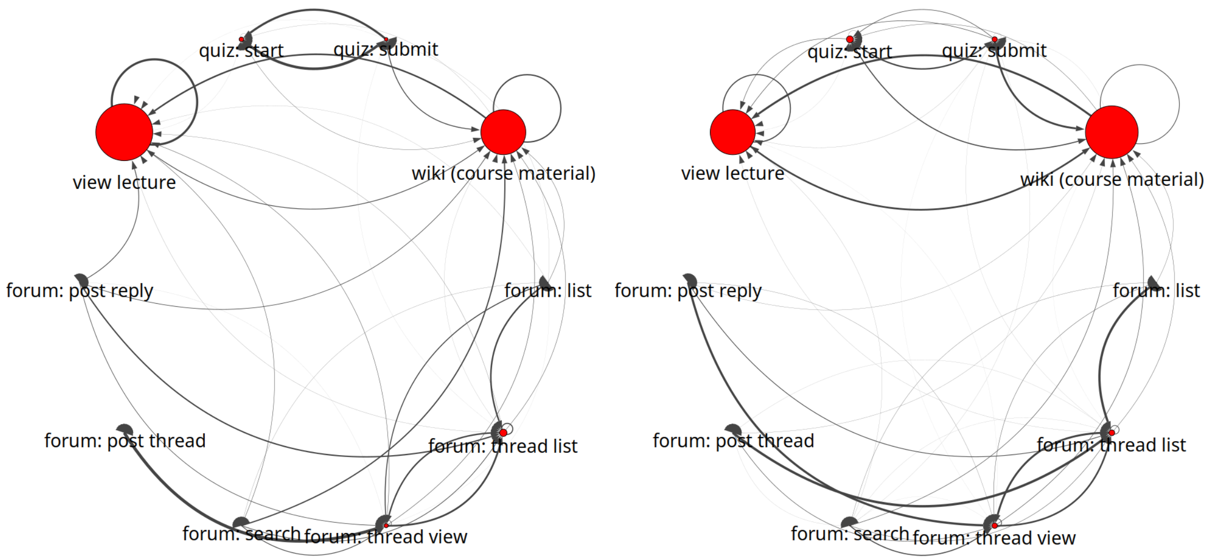
(a) An example "quiz taking" state. (b) An example "lecture viewing" state.

Figure 2: Two example states found by the 6-state 2L-HMM. (The naming of these states reflects our own interpretation.)

ties than the other five states) whereas the second corresponds to a "lecture viewing" state. Our unsupervised method can uncover states that do indeed correspond to student behavior modes that we would expect to find a priori.

We also argue that it is important that the latent state representation is a Markov model rather than just a discrete distribution over actions in $\mathbf{A}$ (as would be the case for a traditional single-layer HMM). We can observe why if we take a closer look at each of the two latent state representations in Figure 2 and look at their forum component (bottom right). We can see that the relative probability of the forum activities is roughly the same between these states, but the *transitions* are quite different. In Figure 2a we have a relatively low probability of walking from the "forum thread view" action back to the "forum thread list" action (see the bottom rightmost arc; transition probability $p \approx 0.17$), but in Figure 2b we actually observe a much stronger link in this direction (transition probability $p \approx 0.63$). This difference highlights an important distinction between these two latent states: in the first you are more likely to visit the forum *looking for a post*, where in the second you are more likely to visit the forum *to browse existing posts*. Thus, capturing the action transition matrix within a latent state is important for capturing detailed insights involving bigrams of actions.

We can also use our model for cross-course behavior analysis. In Figure 3 we show two latent state representations learned by a 6-state 2L-HMM, one from textretrieval-001 and one from sustain-001. These two states were chosen as they are the most similar between the two courses. However, if we look at the transitions we can see some important differences. First, in the state from textretrieval-001, we can see that the probability of returning to the course wiki after viewing a lecture (transition probability $p \approx 0.24$) is considerably lower than that probability in the state from sustain-001 (transition probability $p \approx 0.57$). We also notice that



(a) A state from textretrieval-001. (b) A state from sustain-001.

Figure 3: Two similar example states found by the 6-state 2L-HMM.

the self-loop for staying in a lecture activity in textretrieval-001 (transition probability $p \approx 0.70$) is significantly higher probability than it is in the state from sustain-001 (transition probability $p \approx 0.34$). This gives us some insight into the lecture viewing behavior of students in these two MOOCs which might reflect the course's structure (which, as demonstrated in Davis et al. (2016), can differ substantially across different MOOCs). In textretrieval-001, students are likely to view lecture videos in succession directly without first visiting the course wiki, whereas in sustain-001 students are much more likely to first return the course wiki before watching the next lecture video. This observation would be lost if we did not consider the transitions between the actions within the latent states.

4.2. Varying the Number of Latent States

Our 2L-HMM model has a parameter K that sets the number of latent states to be learned. We believe that this can allow our model to flexibly discover patterns of different granularity, and we can show this by varying K for a course and observing how the latent state representations evolve.

In Figure 4 we see[3] the evolution of the latent state representations found for the textretrieval-001 MOOC. With $K = 2$ we uncover a video watching pattern (state 1) and a course material browsing pattern (state 0). However, we can see that when $K = 3$ we can uncover a new pattern involving forum behavior in state 3 (notice the node sizes on the bottom right). As we increase K to four, we can see that state 1 splits into state 1 and state 3. These states appear quite similar at a glance, but there are still some key differences. First, we can see that state 1 now has a non-negligible forum component, whereas state 3 has hardly any weight on the forum.

We observe similar behavior in Figure 5 as we increase K when fitting our 2L-HMM on the

[3]The graphs are quite small, but the states are in the same positions as they were in previous figures.

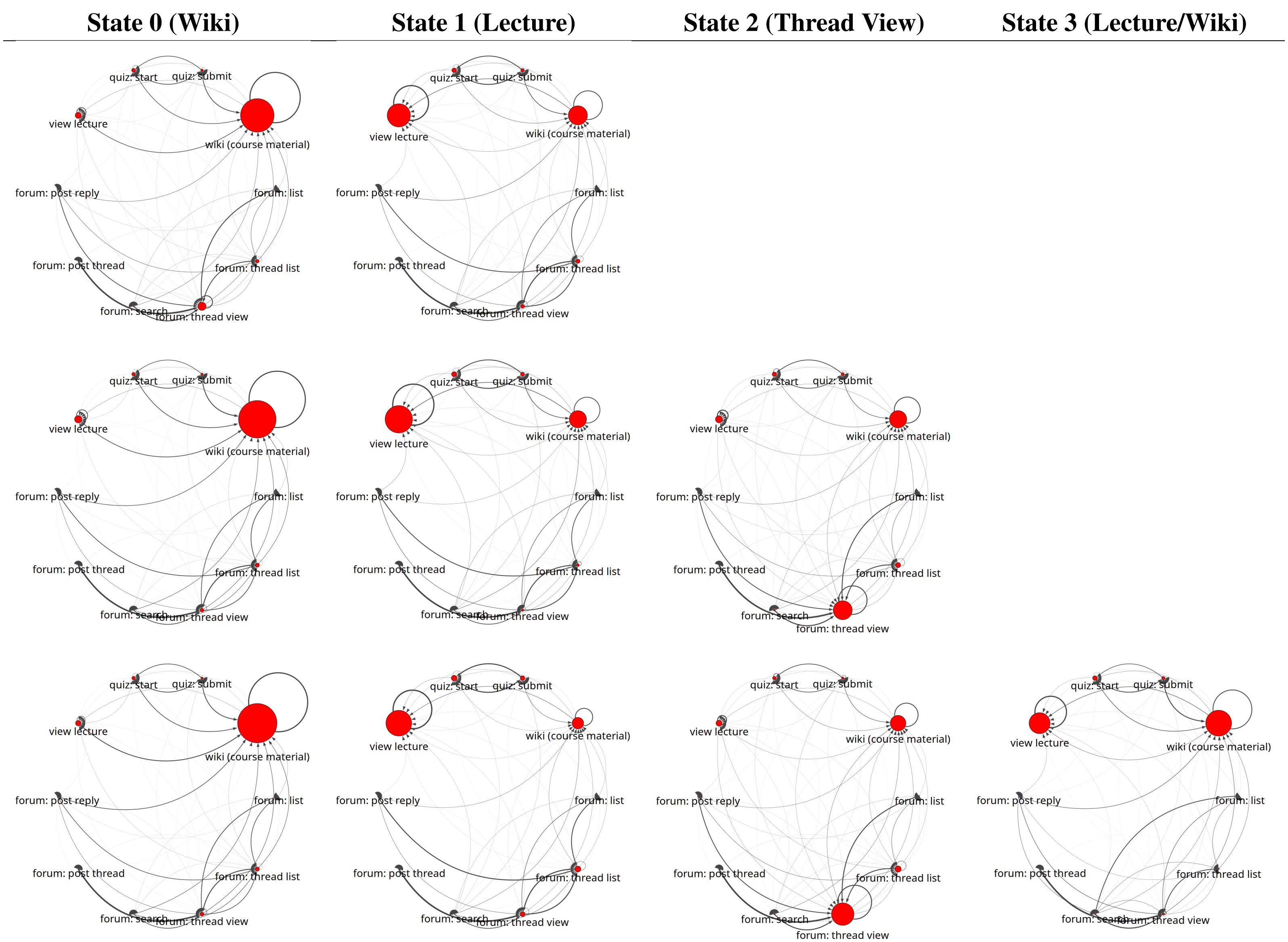


Figure 4: The evolution of states for increasing K for the textretrieval-001 MOOC. Each row corresponds to the next value of K, starting from $K = 2$. State “names” indicate the highest probability action(s) within the state.

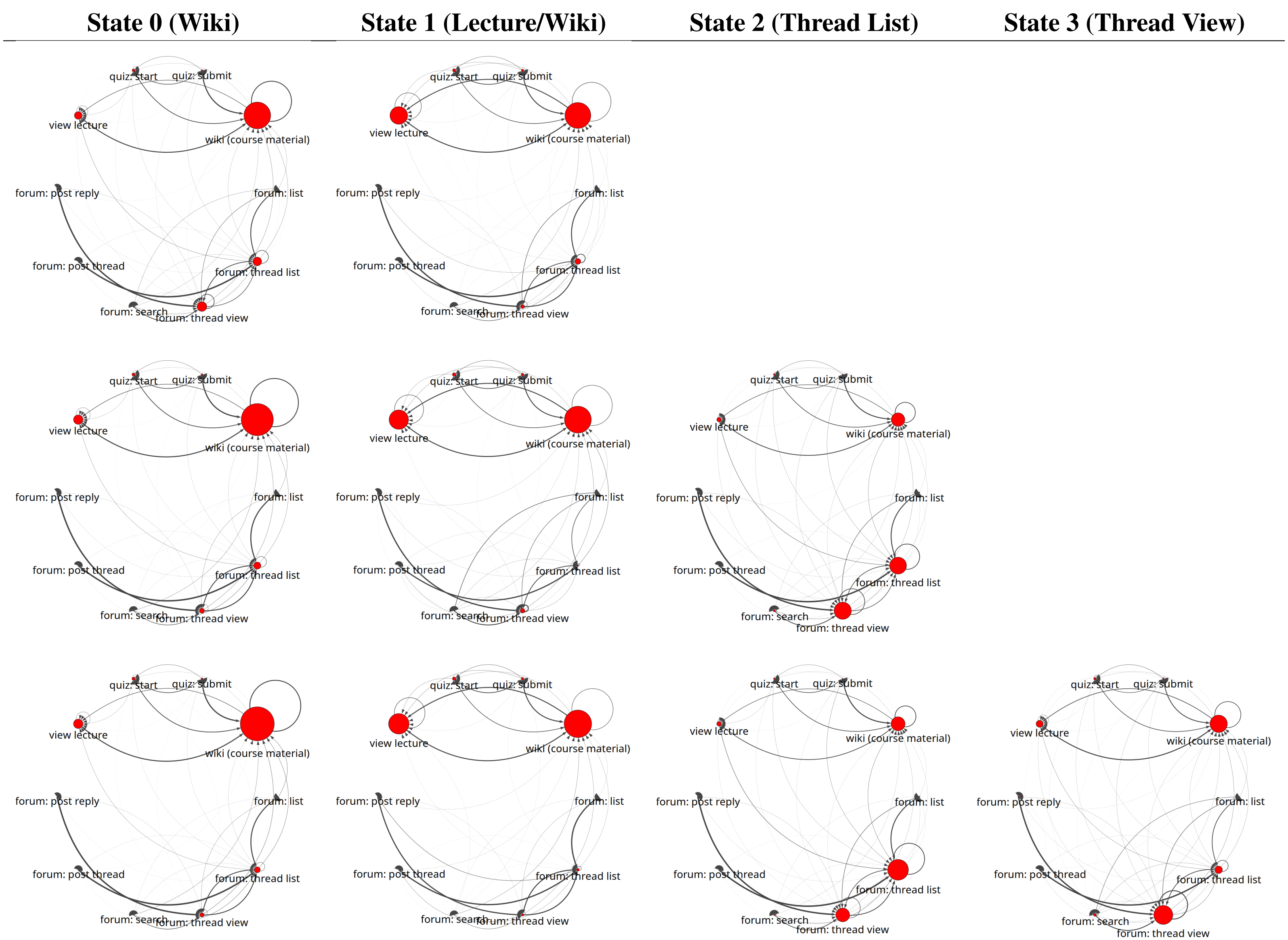


Figure 5: The evolution of states for increasing K for the sustain-001 MOOC. Each row corresponds to the next value of K, starting from $K = 2$. State “names” indicate the highest probability action(s) within a state.

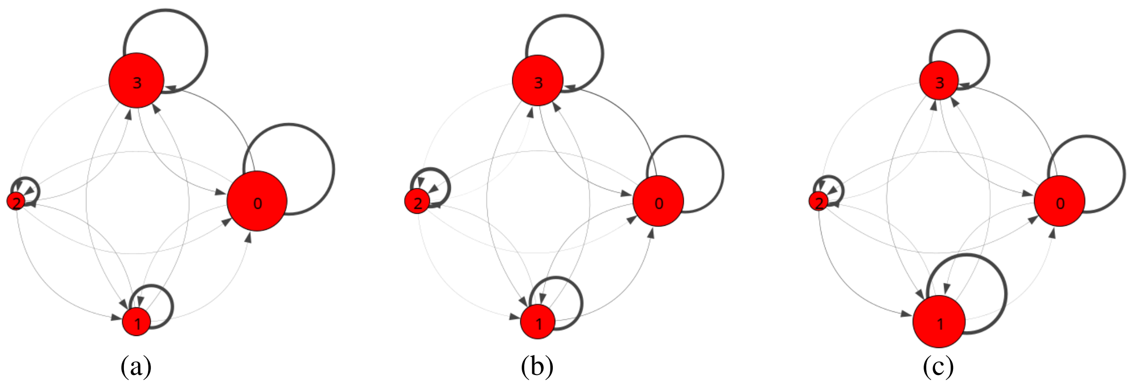


Figure 6: The latent state transition diagrams for a 4-state 2L-HMM fit to textretrieval-001 for all students (a) compared to only "perfect" students (b) and only "low" students (c).

sustain-001 MOOC. Again, in the transition between $K = 2$ and $K = 3$, we discover forum behavior patterns as a latent state. However, in the transition between $K = 3$ to $K = 4$, we instead see a refining of that discovered forum state, where state 3 captures asymmetric transition probabilities between "forum thread view" and "forum thread list". These states could be seen as redundant, in which case a setting of $K = 3$ may be more appropriate for the sustain-001 dataset.

4.3. Transitions Between Latent States

A unique property of our model is its ability to capture transitions between the *behavior patterns themselves* that are captured by the latent states. In Figure 6a we show the latent state transition diagram for a 4-state 2L-HMM fit on textretrieval-001. We can immediately observe two things: (1) each latent state has a very high "staying" probability, and (2) the prevalence of each latent state matches our intuition. In particular, we can see that the forum browsing state (state 2) has relatively lower probability than the other states as we might expect. It also makes sense that state 0 has a rather high probability as this state likely captures all sequences where a student logged in to the platform and then did nothing else (likely checking for updates). If we look at state 1 and state 3, their relative probabilities match our intuition as well. State 1 seems to capture a more engaged browsing session, where there is non-negligible probability associated with different activities such as quiz taking and forum browsing and, importantly, these activities have high probability symmetric edges (so students are taking quizzes one after the other, or viewing forum threads in succession). By contrast, state 3 seems to capture a more passive student, with negligible probability mass associated with forum activity (with low symmetry in the edges). The link between "quiz submit" and "quiz start" (indicating quiz repetition) is also significantly lower than state 1.

Thus, we might expect to see students that perform well in the course preferring states 1 and 2 over states 0 and 3. To verify this, we took the model we learned on the full training data and retrofit it to training data consisting only of sequences produced by students in textretrieval-001 that had perfect marks. To prevent the latent state meanings from drifting, we forced the model parameters associated with their Markov model representations to be fixed, in effect only learning initial and transition probabilities for the top layer of our 2L-HMM. We show the updated latent state transition diagram in Figure 6b. We can clearly see that the probability of state 2 has increased dramatically, consistent with previous observations of the positive correlation between forum activity and grades (Huang et al., 2014), while the probability of states 0 and 3 has

Table 2: Average rank for "perfect" and "low" student groups in the ranked lists associated with the four latent states found by a 2L-HMM. † indicates statistically significant different mean ranks at $p < 0.01$ according to an unpaired t-test.

Group	**State 0**	**State 1†**	**State 2†**	**State 3**	**State 2 → 2†**
Perfect	975.3	1001.5	999.0	1056.5	939.6
Low	1024.9	816.4	1230.5	1161.2	1187.4

decreased. State 1 had its probability increase, but only very slightly.

In Figure 6c we plot the latent state transition diagram for a second group of "low" students. These students were selected so that they attempted all required quizzes in the course, but such that their average quiz score was $\leq 70\%$. Here, we see that state 1 has a large increase in size, where we might have expected state 3 to grow instead. However, there is an alternative explanation for this phenomenon. Since state 1 seems to indicate a highly engaged student, it is a perfectly reasonable explanation for the "low" student group as they are going to be working hard to try to fill in the gaps in their knowledge. By contrast, the "perfect" student group likely has many members who can take the quiz more passively and get perfect marks, perhaps because they already know much of the material being presented, or are just naturally strong and do not require much background review to perform well. This also explains why state 1 did not increase in size for the "perfect" group like we were anticipating. Kizilcec et al. (2017) observe similar phenomena with the courses they studied where they find that certificate earning is negatively correlated with help seeking behavior. Our model lends itself well to discovering this potentially counter-intuitive insight directly from data.

To quantify this finding, we perform the following experiment. First, we select all students from textretrieval-001 who completed all of the quizzes $\mathbf{L}_q \subset \mathbf{L}$. This gives us 1,985 students along with their average quiz score. We then create a ranked list of the students in $\mathbf{L}_q$ by sorting them by their "preference" for a specific latent state

$$p_\ell(i) = \frac{\sum_{t=1}^{T} \gamma_t^{(\mathbf{o}_\ell)}(i)}{\sum_{j=1}^{K} \sum_{t=1}^{T} \gamma_t^{(\mathbf{o}_\ell)}(j)} \tag{11}$$

where $\mathbf{o}_\ell$ is the list of action sequences for student ℓ and γ is defined as before and computed using the Baum-Welch algorithm. We can now compare the average rank in this list for both the "perfect" student group and the "low" student group: a useful state for distinguishing the two groups should result in a ranked list with statistically significant differences in average rank between the two groups. Our results are summarized in Table 2. Indeed, we discover that states 1 and 2 are correlated with the "perfect" or "low" groups: state 1 ranks students in the "low" group significantly higher than those in the "perfect" group, and state 2 does the opposite and prefers students in the "perfect" group to those in the "low" group.

Returning to Figure 6, we can also see a difference in the transitions between the latent states. In particular, look at the transitions in Figure 6b and Figure 6c that leave state 2. In the "perfect" group, nearly all this probability mass is allocated for the self-loop ($p \approx 0.93$). In the "low" group, this self-loop is less strong ($p \approx 0.80$; most easily seen by noting that the edges leaving state 2 are darker than for the "perfect" group). We can perform a similar experiment to above by producing a ranked list of students $\ell \in \mathbf{L}_q$ by their staying probability for state 2 (that is, given that a student ℓ is already in state 2, how likely are they to remain there in the next action sequence?). This can be computed as

$$p_\ell(2,2) = \frac{\sum_{t=1}^{T-1} \xi_t^{(\mathbf{o}_\ell)}(2,2)}{\sum_{i=1}^{K} \sum_{t=1}^{T-1} \xi_t^{(\mathbf{o}_\ell)}(2,i)} \tag{12}$$

where ξ is defined as before and computed using the Baum-Welch algorithm. The last column of Table 2 indicates that this transition feature also correlates with the achievement group and results in significant differences in mean rank.

5. Limitations and Potential Drawbacks

There are a few limitations of our model that are important to highlight. First, there are specific technical limitations due to the statistical nature of the model and the particular methodology we propose for fitting our model parameters. Second, there are limitations in the kinds of patterns our model is able to discover in its current formulation and the ease with which instructors are able to extract knowledge from these patterns. We discuss both lines below.

5.1. Technical Limitations and Implementation Challenges

One potential limitation is that the model is complex and has many parameters in order to truly uncover the relevant patterns in the data. To properly estimate these parameters at training time, a large amount of data must be available to the training algorithm. The assumption that we have a large amount of sequence data available for training on should hold for most MOOC courses, but this assumption may be problematic if attempting to apply our model to smaller online (or on-campus) courses.

Our model fits its parameters using maximum likelihood estimation using the EM algorithm. The EM algorithm is a hill climbing algorithm that is optimizing in a highly non-convex parameter space. Thus, it can only guarantee that we reach a local maximum in practice (Dempster et al., 1977). This may mean that the parameters found for the model may not be the "best" parameters in a global sense, which may lead to suboptimal latent state representations and transition patterns. Empirically, however, we observed in our experiments that strong patterns tend to always show up if algorithm reaches a reasonably good local maximum, and the differences of the results tend to be related to weak "unstable" patterns which may not be reliable anyway. Since it is far more important and useful to reveal the strong salient patterns than weak unreliable ones for instructors, the problem might not necessarily affect the utility of the approach so significantly. A commonly applied approach to address the problem of multiple local maxima is to run the model multiple times with different starting points to allow the model to explore a larger portion of the parameter space. One can then compare the log-likelihood of the data between the models that were started from different initialization points and select the one that has the highest value. This still does not guarantee that we find a global maximum, but it does help alleviate the potential for finding a particularly bad local maximum. In practice, we have found that our model can be fit to the data quickly on commodity hardware, so running it multiple times to address this concern is not computationally unfeasible.

A further complication in blindly applying the EM algorithm to our model is the fact that the observation probabilities will be incredibly small. The probability that a specific sequence is generated by a specific Markov chain (one of our latent states) will decrease exponentially with its length. While there are general approaches to avoiding numerical underflow in hidden Markov models, applying the "scaling" method proposed by Rabiner (1990) will still result in numeric stability issues in our case due to the incredibly small sequence-generation probabilities. We address this in our open-source implementation by computing the trellises in log-space and using the log-sum-exp trick when we need to take summations, which exploits the identity

$$\log \sum_{i=1}^{n} e^{x_i} = a + \log \sum_{i=1}^{n} e^{x_i - a} \tag{13}$$

where a is typically set to $\max_i x_i$ to improve stability. We did not encounter further stability issues once we applied these two tricks.

As is the case with traditional hidden Markov models, it is often important to smooth the underlying model's distributions to ensure that there is a non-zero probability of generating the observations. We employed a simple additive smoothing in our implementation with a small additive constant (10^{-6}) for all our transition matrices to avoid this problem.

5.2. Limitations of Discovered Patterns

The proposed behavior representation is most suitable for representing recurring behavior patterns, which presumably are most interesting to extract from the data, but may not cover all the interesting patterns in the data. It would be interesting to explore other complementary models such as time series models, which may help capture non-recurring patterns.

Our proposed model is flexible in the patterns it can discover in two main ways. The first is that the granularity of the patterns can be adjusted by changing the segmentation strategy one uses to divide the user action stream into discrete "sessions." The other is the number of latent states K that are used to describe the segmented action sequences. One drawback of these two lines of flexibility is that there is not necessarily a clear answer as to the "correct" approach for both in any given scenario. Varying the segmentation strategy changes the granularity of the patterns the latent states explain, which will change their meaning. Varying the number of latent states increases the flexibility of the model, but also may result in latent states that are not substantially different from the other latent states and/or latent states with very low probability. This flexibility forces a user of our model to make some assumptions about the patterns they wish to find (granularity, diversity), and the model itself does not necessarily provide clear guidance as to what the best approach is.

Furthermore, we have made an implicit assumption that the segmentation strategy is applied as a pre-processing step (and is most obviously a deterministic process). The proposed segmentation strategies in this paper do not specifically allow for the transitioning between the different latent behavior patterns to occur over different windows of time for different users: we have made a strong assumption that transitions between latent states only occur at action sequence boundaries. One can model how much time a user stays in each state in terms of the number of sessions they remain there before transitioning, but it may be better to directly model the amount of time we expect a user to stay in the given state. In other words, a more powerful model might be one in which the segmentation and the latent behavior pattern discovery are jointly modeled in a single probabilistic framework rather than being separate pieces as we investigate in this work.

While the model can discover *patterns* in the data in a fully automated way, there is still clearly a burden on the user of our model to interpret the patterns it has discovered to create actionable *knowledge* about the MOOC from which the data was extracted. Extracting the patterns is a necessary step towards the creation of knowledge, and we view our model as a component in a collaborative system which leverages the machine to perform statistical modeling to extract patterns which then enable a human actor to extract knowledge and take actions on the basis of the data. The pattern discovery is an important and absolutely necessary component, without which it would be very difficult, if not impossible, for humans to directly digest the student behavior buried in the large amount of data. Of course, pattern discovery is only a means to help humans obtain knowledge, not the end of the knowledge discovery process.

6. Conclusions and Future Work

As a tool to help instructors and education researchers better understand the behavior of MOOC students, we proposed a two-layer hidden Markov model to automatically extract student activity patterns in the form of behavior state-transition graphs from large amounts of MOOC log data. This model is different from existing methods in that it treats behavior patterns as a sequence of *latent states*, rather than assigning these states in a rule-based manner. It captures the variable behaviors of students over time and allows analysis at different levels of granularity.

We showed that such a model does, in fact, capture meaningful behavior patterns and produces descriptions of these behavior patterns that are easy to interpret. We argued that it is important to capture student behavior patterns with more sophisticated models than simple discrete distributions over actions to capture information present in bigrams of actions (or larger sequences). By varying the number of latent states inferred, we showed that the model is flexible and can capture patterns of differing levels of specificity in this way. Finally, we investigated whether we can detect differences in student behavior patterns as they correlate with course performance. Specifically, we demonstrated that high-performing students produce substantially different HMM transition diagrams that tend to show longer concentration span in quiz-taking and more active forum participation as compared with the average students. These results show the great potential of the proposed model for serving as a tool to help humans discover knowledge about student behaviors.

There are a number of interesting directions to further extend our work in the future. First, the proposed model is completely general and can thus be easily applied to analyze the log of many other courses to enable deep understanding of student behaviors as well as the correlations of such behaviors and other variables such as grades. To realize these benefits, it would be useful to develop a MOOC log analysis system based on the proposed model to facilitate education research and help instructors improve course design.

Second, our model can empower many new comparative analyses. For example, we could now look at how behavior patterns change between different offerings of the same MOOC to understand how changes in course structure or materials influence student behavior. Individual students can now also be compared against each other or against groups. For example, by decoding the latent state sequences for each student, we can measure how "surprising" their latent state transition sequence is relative to the average we would expect according to the model, or to the average "perfect" student, etc. We can now investigate how certain behavioral patterns correlate with properties of a student (e.g., demographics, prior aptitude, etc.). After decoding the students' latent state sequences, we could also correlate course-wide drifts in these latent states with events in a course. For example, we might be able to automatically discover difficult or confusing parts of a course by noticing spikes in the distribution of students over latent states over time.

Third, there is more recent work on better learning algorithms for mixtures of Markov models (Gupta et al., 2016). It would be worth exploring whether the advances proposed in this and similar work can be applied to our model to address some of the concerns surrounding our use of the EM algorithm for our parameter estimation.

Finally, the proposed model can be extended in several ways. For example, although our model does not explicitly model drop-out like Kizilcec et al. (2013), doing so is an obvious extension. Our model would be able to provide predictions of when a student is "at risk" for dropping out under such an extension. Also, currently, the model learns a transition matrix over the latent states that is *shared* across all students. It would be interesting to instead learn a different latent state transition matrix for each individual student, but keep the second-level Markov models shared. This would provide the model with more flexibility which may be desirable itself, but would also naturally result in a description of a student (via his or her HMM transitions) that could be incorporated into existing supervised learning techniques that try to predict student outcomes for understanding which of the latent behavior patterns discovered by 2L-HMM are most predictive of the performance of student learning. One could also relax this somewhat and posit that *groups* of students transition between the lower layer patterns identified by our method in distinct ways; this way, we can do a soft clustering of students into K_2 groups based on the similarity of their transitioning behavior between the higher-level behaviors we can identify.

7. Acknowledgments

The authors would like to thank Hao Zheng, Jason Cho, and Sean Massung for their helpful discussion while developing this work, and the ATLAS group at UIUC for their help with securing the datasets used. We also thank the anonymous peer reviewers whose critical reading and constructive feedback were crucial for improving the final manuscript. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant Number DGE-1144245 and by the National Science Foundation Research Program under Grant Number IIS-1629161.

References

Baker, R. S. J. d., Corbett, A. T., and Aleven, V. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing. In *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, B. Woolf, E. Aïmeur, R. Nkambou, and S. Lajoie, Eds. ITS 2008. 406–415.

Corbett, A. T. and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction 4,* 4, 253–278.

Davis, D., Chen, G., Hauff, C., and Houben, G.-J. 2016. Gauging MOOC learners' adherence to the designed learning path. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. EDM '16. International Educational Data Mining Society (IEDMS), 54–61.

DEMPSTER, A. P., LAIRD, N. M., AND RUDIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological) 39,* 1, 1–38.

FAUCON, L., KIDZINSKI, L., AND DILLENBOURG, P. 2016. Semi-Markov model for simulating MOOC students. In *Proceedings of the 9th International Conference on Educational Data Mining*, T. Barnes, M. Chi, and M. Feng, Eds. EDM 2016. International Educational Data Mining Society (IEDMS), 358–363.

FINE, S., SINGER, Y., AND TISHBY, N. 1998. The hierarchical hidden Markov model: Analysis and applications. *Mach. Learn. 32,* 1 (July), 41–62.

GUPTA, R., KUMAR, R., AND VASSILVITSKII, S. 2016. On mixtures of Markov chains. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 3441–3449.

HAMILTON, J. D. 1990. Analysis of time series subject to changes in regime. *Journal of Econometrics 45,* 1, 39 – 70.

HUANG, J., DASGUPTA, A., GHOSH, A., MANNING, J., AND SANDERS, M. 2014. Superposter behavior in MOOC forums. In *Proceedings of the First ACM Conference on Learning @ Scale*, A. Fox, M. A. Hearst, and M. T. H. Chi, Eds. 117–126.

HUANG, X., ARIKI, Y., AND JACK, M. 1990. *Hidden Markov Models for Speech Recognition*. Columbia University Press, New York, NY, USA.

JEH, G. AND WIDOM, J. 2003. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, G. Hencsey and B. White, Eds. 271–279.

JURAFSKY, D. AND MARTIN, J. H. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

KIZILCEC, R. F., PIECH, C., AND SCHNEIDER, E. 2013. Deconstructing disengagement: Analyzing learner subpopulations in massive open online courses. In *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, D. Suthers, K. Verbert, E. Duval, and X. Ochoa, Eds. LAK '13. 170–179.

KIZILCEC, R. F., PREZ-SANAGUSTN, M., AND MALDONADO, J. J. 2017. Self-regulated learning strategies predict learner behavior and goal attainment in massive open online courses. *Computers & Education 104*, 18 – 33.

KOLLER, D. AND FRIEDMAN, N. 2009. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press.

MASSUNG, S., GEIGLE, C., AND ZHAI, C. 2016. MeTA: A unified toolkit for text retrieval and analysis. In *Proceedings of ACL-2016 System Demonstrations*, S. Pradhan and M. Apidianaki, Eds. Berlin, Germany, 91–96.

OLIVER, N., GARG, A., AND HORVITZ, E. 2004. Layered representations for learning and inferring office activity from multiple sensory channels. *Comput. Vis. Image Underst. 96,* 2 (Nov.), 163–180.

PAGE, L., BRIN, S., MOTWANI, R., AND WINOGRAD, T. 1999. The PageRank citation ranking: bringing order to the web.

PIECH, C., BASSEN, J., HUANG, J., GANGULI, S., SAHAMI, M., GUIBAS, L. J., AND SOHL-DICKSTEIN, J. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds. 505–513.

RABINER, L. R. 1990. Readings in speech recognition. Chapter A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, 267–296.

SHIH, B., KOEDINGER, K. R., AND SCHEINES, R. 2010. Unsupervised discovery of student strategies. In *Proceedings of the 3rd International Conference on Educational Data Mining*, R. S. Baker, A. Merceron, and P. I. Pavlik, Jr., Eds. EDM 2010. International Educational Data Mining Society (IEDMS), 201–210.

SONG, Y., KEROMYTIS, A. D., AND STOLFO, S. J. 2009. Spectrogram: A mixture-of-Markov-chains model for anomaly detection in web traffic. In *16th Annual Network and Distributed System Security Symposium*, G. Vigna, Ed. NDSS. ISOC.

YPMA, A. AND HESKES, T. 2002. Automatic categorization of web pages and user clustering with mixtures of hidden markov models. In *4th International Workshop on Mining Web Data for Discovering Usage Patterns and Profiles*, O. R. Zaïane, J. Srivastava, M. Spiliopoulou, and B. Masand, Eds. WEBKDD 2002. Springer, 35–49.

ZHANG, D., GATICA-PEREZ, D., BENGIO, S., MCCOWAN, I., AND LATHOUD, G. 2004. Modeling individual and group actions in meetings: A two-layer HMM framework. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*. 117–117.