Infinite Mixtures of Markov Chains

Jan Reubold[1] and Ahcène Boubekki[2] and Thorsten Strufe[1] and Ulf Brefeld[2]

[1] TU Dresden, Dresden, Germany {jan.reubold,thorsten.strufe}@tu-dresden.de
[2] Leuphana University, Lüneburg, Germany {boubekki,brefeld}@leuphana.de

Abstract. Facilitating a satisfying user experience requires a detailed understanding of user behavior and intentions. The key is to leverage observations of activities, usually the clicks performed on Web pages. A common approach is to transform user sessions into Markov chains and analyze them using mixture models. However, model selection and interpretability of the results are often limiting factors. As a remedy, we present a Bayesian nonparametric approach to group user sessions and devise behavioral patterns. Empirical results on a social network and an electronic text book show that our approach reliably identifies underlying behavioral patterns and proves more robust than baseline competitors.

1 Introduction

Being able to translate a user's behavior into an educated guess of her intent is often the key to provide a satisfying user experience. Users express different behavior in different contexts to satisfy their needs, fulfill a task, etc. [1]. Characteristic behavioral traits may thus serve as indicators for future behavior and capturing these traits is important in many application domains:

Content providers on the Web often rely on repeated user visits. Their success depends highly on how well they are able to anticipate a user's needs by providing the right content, at the right time, and in the right place. Accurately modeling user behavior not only predicts a user's actions but informs design and content decisions. This includes predicting what links a user will click on, deciding where webpage components should be placed, and what content to provide.

A similar problem arises in emerging areas such as *educational research* that aim to provide tailored learning environments and tutoring systems to children and students. Often it is either undesirable or not possible to build personalized models, and even when available, such models suffer from the cold start problem, or are unable to deal with context-dependent variations in user behavior. Accurately modeling user behavior leads to accurate assessments of a user's competency and allows for selecting next items, appropriate feedback, etc.

Recently, user behavior plays an increasing role in *security* related areas. Behavioral models are studied as replacements for passwords and intelligent pieces of operating systems are being developed to actively block security related components, such as access to a company data base, when the user is checking news on Facebook. Similarly, security relevant features can be blocked by such a

system if the user behavior deviates from the expected behavior; e.g., to prevent hacking a stolen device.

Traditionally, Markov models are frequently studied methods in behavioral contexts [4,10,7,27] due to their good interpretability. The underlying idea is to exploit the sequential nature of user behavior and translate user sessions into Markov processes. Using Expectation-Maximization (EM)-based approaches [28], similar sessions can be grouped to draw conclusions about different types of users and their behaviors from the arising clusters. While there is nothing wrong with the general blueprint of these analyzes, they often suffer from being parametric approaches and using greedy optimization strategies that may lead to poor local optima. The problem arises because the optimal number of clusters is *a priori* unknown and needs to be identified with heuristics (e.g., [30,29]) or trial and error. Often, this leads to repeated parameter estimations on subsets of the data. In addition, EM-based algorithms potentially converge to local optima and, therefore, several repetitions of the same experiment with random initializations are required. In the presence of todays data set sizes, the multiplicative consequences of deploying heuristics with EM-based algorithms quickly become prohibitive.

We present a non-parametric Bayesian approach to fit a mixture model of Markov chains to sequential data, turning behavior into data. We draw conclusions from the resulting models that constitute novel insights and show how these insights impact future developments and design decisions.

2 Related Work

Modeling user behavior is often performed using probabilistic models in combination with some sort of clustering. The most commonly studied type of approaches are based on Markov models [7,10,4,11,12]. Early work investigates the use of probabilistic methods and subsequent publications use the formalism of Markov chains [7,10] to build stochastic models that capture behavioral patterns. [4] further explores the idea by proposing a mixture model of Markov chains to divide data into meaningful groups and focus on these groups in the analysis. Here, each manifestation of a common behavioral pattern is represented by a Markov chain. Putting this in the context of our application scenario, a user interacts with a system and, by doing so, transitions between (possibly latent) states of a Markov model. Each state represents a possible interaction between user and system. Due to the use of first-order Markov chains, the next state is only conditioned on the previous state. The approach by [4] yields interpretable results and is computationally efficient. However, model selection may lead to sub-optimal results as identifying the number of groups is not always straight forward and the non-convex problem may have many local optima. A similar approach using a mixture model of hidden Markov models [11] takes intertwined click traces into account while [12] propose selective Markov models for identifying user behavior patterns.

Generally, higher-order Markov models [13,14,15] capture user behavior in more detail. However, [13] suffers from inefficient computations and results that

are difficult to interpret. Other approaches require unreasonably large data sets as the model parameters grow exponentially with the number of states N and order o [14,15]. [19,16,17] make use of Bayesian nonparametric mechanisms to control the complexity of the respective models. E.g., combining a temporal point process with a Bayesian nonparametric prior, [19] study the relation between both areas. The resulting Dirichlet-Hawkes process allows to model user behavior in greater detail compared to first-order Markov models. However, point processes focus on predictive performance and often lack interpretability.

To satisfy all requirements, we propose an approach that combines both, Bayesian nonparametric methods and Markov models. We derive a model that adapts to the complexity of the data and, at the same time, retains interpretability.

3 Non-parametric Bayesian User Behavior Models

In this section, we briefly introduce mixtures of Markov chains models and discuss their properties. After pointing out the drawbacks of this approach, we present a Bayesian nonparametric interpretation that mitigates these issues.

3.1 Mixtures of Markov Chains

Markov chains are probabilistic models for generating sequences of discrete events. The probability of observing an element directly depends on the previous one[3]. Let us consider N sequences (or user sessions) $\mathbf{x}^{(i)} = (x_1^{(i)}, \ldots, x_{T^{(i)}}^{(i)})$ of length $T^{(i)}$ over an alphabet M such that every $x_t^{(i)} \in M$ with $i \in \{1, \ldots, N\}$. For ease of notation, every sequence is augmented by auxiliary start $x_0^{(i)} = S$ and terminal $x_{T^{(i)}+1} = E$ symbols, where $M \cap \{S, E\} = \emptyset$. The probability of observing adjacent elements is then given by the conditional $\theta_{u,v} = p(x_{t+1} = v | x_t = u)$ where $u \in M \cup \{S\}$ and $v \in M \cup \{E\}$. Note that the first event of a sequence is selected according to the prior surrogate $\theta_{S,v} = p(x_1 = v|S)$. Thus, the auxiliary start and terminal symbols allow for capturing prior and terminal distributions, respectively, where the latter eventually serves as a natural duration model of a cluster. The parameters θ are estimated by a maximum likelihood approach[4].

If there are several, say K, generating distributions instead of a single one, a mixture model of Markov Chains (MMC) is required for parameter estimation. Latent indicator variables z_i assign sequences to one of the K clusters and priors $\pi_k = p(z^{(i)} = k|\Theta)$ assess the importance of these clusters where $\Theta = (\pi_1, \ldots, \pi_K, \theta_1, \ldots, \theta_K)$. The quantity $p(z^{(i)} = k|\mathbf{x}^{(i)}, \Theta)$ estimates the probability that sequence i has been generated by the k-th component. To not clutter the notations unnecessarily, we omit superscript i whenever context allows. The likelihood of the model is given by

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^{K} p(z = k|\Theta) \prod_{t=1}^{T+1} p(x_t|x_{t-1}, z = k, \Theta) = \sum_{k=1}^{K} \pi_k \prod_{t=1}^{T+1} \theta_{x_{t-1},x_t}^k$$

[3] We focus on first-order dependencies but the approach is easily generalized to higher-order models; notation is quickly getting messy though.

Parameters Θ are estimated using Expectation Maximization (EM) and related techniques [28,4].

While EM-based approaches yield interpretable results in an efficient and straight forward way, they suffer from two major drawbacks. Firstly, the actual number of components is generally unknown and consequently K becomes a parameter that has to be adjusted in the model selection. Secondly, the greedy inference by EM-based approaches can converge to local optima. This not only renders a single solution unquantifiable but, also implies repetitions of the same experiment necessary (e.g., using different initializations). Combining the two arguments leads to complex experimentations and quickly becomes tedious.

By contrast, our contribution addresses both limitations of EM-based approaches. Being a Bayesian nonparametric interpretation of the mixture of Markov chains, the number of components is adjusted in a data-driven way during the optimization. The latter is performed by a Gibbs sampling approach that does not share the greedy nature of EM-based methods.

3.2 Infinite Mixtures of Markov Chains

Our contribution, infinite Mixtures of Markov Chains (iMMC), makes use of a computationally efficient approximation to the hierarchical Dirichlet processes (HDP) [2], known as the degree L weak limit approximation [5]. The limiter L denotes the maximum cardinality of the approximated distribution. The approach encourages the learning of models with a state space of less than L components while allowing for the creation of new ones. It can be shown that such an approximation converges to the original HDP as $L \to \infty$ and provides a common solution to efficient Bayesian nonparametrics [23].

Graphical Model Our model consists of a maximum number of L clusters, each comprised of a subset of events $M_l \subseteq M$ with $l \in \{1, ..., L\}$. As before, we differentiate between observations $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$ that assign sequences to clusters. The model is build of two well-known concepts in Bayesian nonparametrics, the Dirichlet distribution (Dir) and the finite-dimensional hierarchical Dirichlet process [2,5]. A hierarchical Dirichlet process (HDP) consists of a two-layer hierarchy of Dirichlet processes (DP).

While the Dirichlet distribution is used to substitute the Multinomial distribution of the MMC to allow for an adaptive prior distribution over the cardinality of the clusters, the observation layer is modeled by a degree L weak limit approximation [5] which captures the Markovian structure of a cluster. The idea of this design choice is that the distribution over the events of a cluster serves as a natural base measure to the emission distributions of the events. Here, the emission distributions denote the transition probabilities from a state to any other. By representing these emission distributions by DPs themselves, we build an HDP representing a cluster. Note that this way we define the Markov models by the emission distributions of its states.

The approximated HDPs consist of a Dirichlet G_l to model the state distribution within a cluster l and a set of subordinate Dirichlet distributions θ_{lm}, which

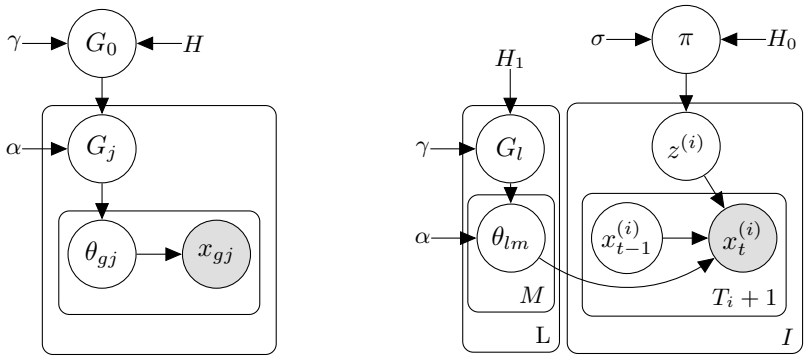

Fig. 1: (left) Graphical model of an HDP mixture model; (right) graphical model of the proposed iMMC; M is the set of events; I is the cardinality of the set of input sequences with T_i as the length of the corresponding sequence; $i \in I$ and $t \in \{0, \ldots, T_i + 1\}$; white- and gray nodes represent hidden states and observed states, respectively.

represent the transitions within a cluster, i.e., the transition distribution given the current cluster l and its current state $m \in M_l$. The prior distributions π, G_l and θ_{lm} are then computed by

$$\begin{aligned} \pi|\sigma &\sim \mathrm{Dir}(\alpha/L, \ldots, \alpha/L) \\ G_l|\gamma &\sim \mathrm{Dir}(\gamma/L, \ldots, \gamma/L) \\ \theta_{lm}|\alpha, G_l &\sim \mathrm{Dir}\left(\alpha G_{l1}, \ldots, \alpha G_{lL}\right). \end{aligned} \tag{1}$$

Note that the prior and terminal state distributions are encoded within θ due to the augmentation of start and terminal symbols. Figure 1 (right) shows the graphical model and the generative process of a single sequence based on the prior distributions is given by

$$z|\pi \sim \pi \qquad x_t|z, x_{t-1} \sim \theta_{z x_{t-1}} \qquad t \in \{1, \ldots, T_i + 1\}. \tag{2}$$

Inference To estimate parameters we make use of a two-step sampling algorithm which consists of the alternation of sequence assignments and parameter updates. In the assignment phase we obtain a realization of the latent parameters which is then used for the update of the prior distributions. These two steps are then repeatedly run to obtain the final model parameters. In the following we explain both steps in detail.

Assignment Step Given randomly initialized prior distributions (see Eq. 1), we compute the likelihood of a sequence x as

$$p(\mathbf{x}|\Theta) = \sum_{l=1}^{L} p(z = l|\Theta) \prod_{t=1}^{T+1} p(x_t|x_{t-1}, z = l, \Theta) = \sum_{l=1}^{L} \pi(l) \prod_{t=1}^{T+1} \theta_{l x_{t-1}}(x_t), \tag{3}$$

Algorithm 1 Blocked Gibbs sampler for iMMC

Given the hyperparameters $\boldsymbol{\sigma}, \boldsymbol{\gamma}, \boldsymbol{\alpha}$

(i) Initialize prior distributions according to Eq. 1

Until convergence **do**:
- (ii) Obtain a realization of $\boldsymbol{z}$ according to Eq. 5
- (iii) During assignment step update auxiliary variables as follows:
  - $\rightarrow$ For each assigned sequence, increment:
    - $\cdot$ $b_{l=z^{(i)}}$ $\equiv$ # observations assigned to cluster l
    - $\rightarrow$ For each observation in the sequence, increment:
      - $\cdot$ $d_{l=z^{(i)},x_t}$ $\equiv$ # observations of state x_t assigned to $l = z^{(i)}$
      - $\cdot$ $s_{l=z^{(i)},x_{t-1},x_t}$ $\equiv$ # transitions from x_{t-1} to x_t in $l = z^{(i)}$
- (iv) Re-sample prior distributions

(v) Build final model from multiple sample-sets of the parameters

where x_0 and x_{T+1} represent the artificial boundary nodes and π the prior distribution over the clusters. The marginal distribution is

$$p(x|z = l, \Theta) \propto \pi(l) \prod_{t=1}^{T+1} \theta_{lx_{t-1}}(x_t). \tag{4}$$

Therefore, the assignments can be sampled as

$$z^{(i)} \sim \mathrm{Mu}\left(\sum_{l \in L} p(x|z = l, \Theta)\delta_l\right), \tag{5}$$

where δ represents the Dirac delta.

Update Step After obtaining a new sample of assignments the prior distributions have to be updated. This is an essential step in the Gibbs sampler and, in our case, straight-forward given that all distributions consist of DPs. Therefore, statistics are gathered during the assignment step. We keep track of the state distribution and transitions within the clusters. Thus, $d_{l,m}$ records the number of observations of state m assigned to cluster l and s_{l,m_1,m_2} records the number of transitions from state m_1 to state m_2 within cluster l. Finally, b_l keeps track of the number of observations assigned to luster l. For each iteration, the auxiliary variables document the assignment step. Then, we can re-sample the distributions using the statistics as the new evidence. A summary of the entire inference process is given in Alg. 1. Note, that, while seemingly similar to classic EM-approaches, the Gibbs sampler is based on sampling rather than on ML solutions. Therefore, it can be shown, that under certain conditions the sampler will converge to the global optimum [31].

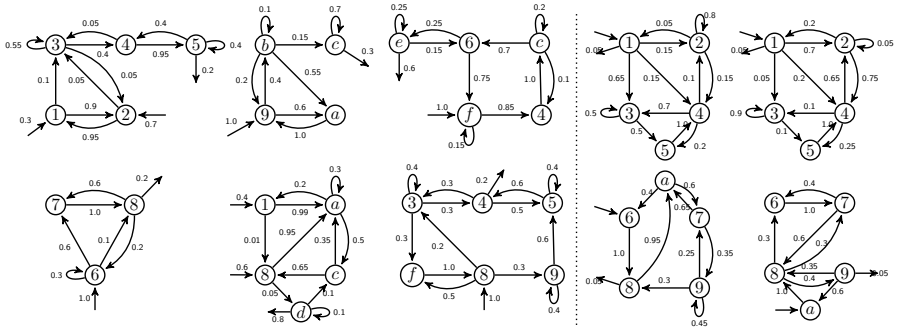

Fig. 2: Generative processes of scenario II (left) and scenario III (right); states are indexed by hexadecimal numbers (1-f).

4 Experiments

We first evaluate the clustering performance of our model in controlled scenarios to understand its effectiveness and to shed light on extreme cases. The synthetic nature of the data allows us to accurately evaluate the clustering performance of our approach. Then we will focus on the interpretability of the clusters and of the extracted patterns. Therefore, we extract usage patterns of users browsing a social network website without prior knowledge. Finally, an analysis of the behavior on an electronic textbook using iMMC will show that the obtained patterns correlate with the success of the corresponding student, suggesting that behavior patterns hold further, sensitive information about students.

4.1 Synthetic data

In this section we compare the clustering performance of our algorithm, the infinite mixture model of Markov chains (iMMC), to the traditional mixture model of Markov chains approach (MMC). We pick the latent Dirichlet allocation (LDA) [32] as an additional baseline to asses the importance of the sequential information contained in the observations. LDA only makes use of the frequency count of events within a sequence.

We generate three synthetic scenarios to generate different sets of clusters. In the context of user behavior, a cluster represents the causal reason for an observed sequence of events: clusters thus serve as proxies for user intention/interest. Their state spaces are the set of events that are associated with one or more clusters. A learning task is simpler when state spaces are disjoint (Scenario I). An example are clusters like 'cooking' and 'driving a car' that have no state spaces of events in common. Learning tasks with fully overlapping state spaces are more difficult (Scenario III, Fig. 2 (right)). Examples are clusters that share many events such as 'cooking' and 'baking' or 'driving a car' and 'driving a motorcycle'. The learning task in Scenario II (Fig. 2 (left)) addresses both characteristics.

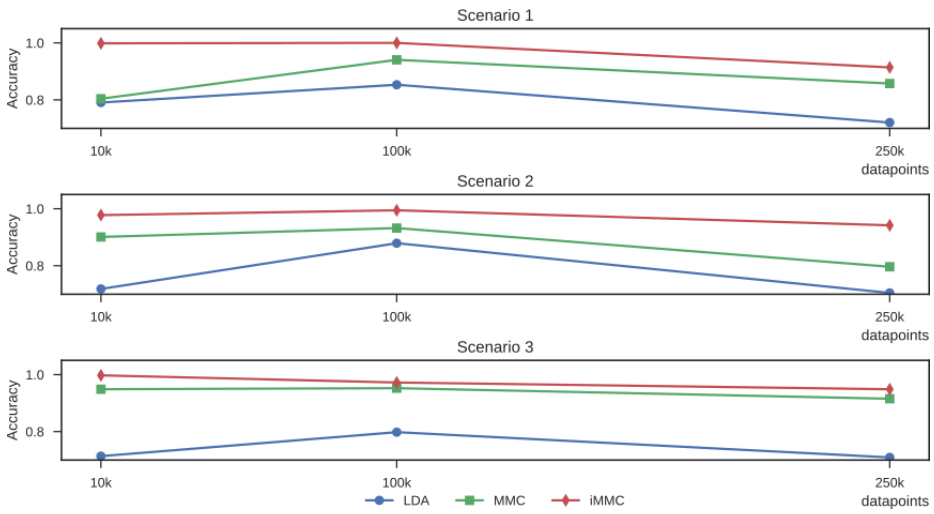

Fig. 3: Accuracy of each method on different scenarios and for dataset sizes.

Table 1: Error rates for the synthetic clustering tasks; each data set consists of $10k$, $100k$, and $250k$ data points (small, medium, large).

	Scenario I			**Scenario II**			**Scenario III**		
	Small	Medium	Large	Small	Medium	Large	Small	Medium	Large
LDA	20.92%	28.14%	28.62%	14.69%	12.09%	20.20%	27.95%	29.54%	29.06%
MMC	19.60%	9.90%	5.13%	5.94%	6.78%	4.77%	14.26%	20.36%	8.47%
iMMC	0.14%	2.23%	0.26%	0.00%	0.54%	2.78%	8.61%	5.82%	5.15%

Given a scenario, we obtain a corresponding data set by selecting uniformly at random one of its clusters. Then we run its generating process which yields a sequence of actions. This procedure is repeated until we have the desired number of actions in the set of generated sequences. For each scenario we evaluate the algorithms on data sets of sizes of $10,000$, $100,000$ and $250,000$ data points. For each combination of scenario and data set size, we generate 10 data sets and report on results of the averaged performances over 5 runs for each of these data sets. While we use a single set of hyperparameter values for our algorithm (each is set to 1), we supply the MMC with the correct number of clusters and apply a soft clustering. For LDA we transform each sequence into a frequency vector of events occurring in the sequence.

Even though MMC was provided with the correct number of clusters and our algorithm had to adjust it to the data, our algorithm is as efficient as MMC. Table 1 and Figure 3 shows the overall clustering performance of both algorithms on all data sets and scenarios. In all cases, our algorithm outperforms MMC.

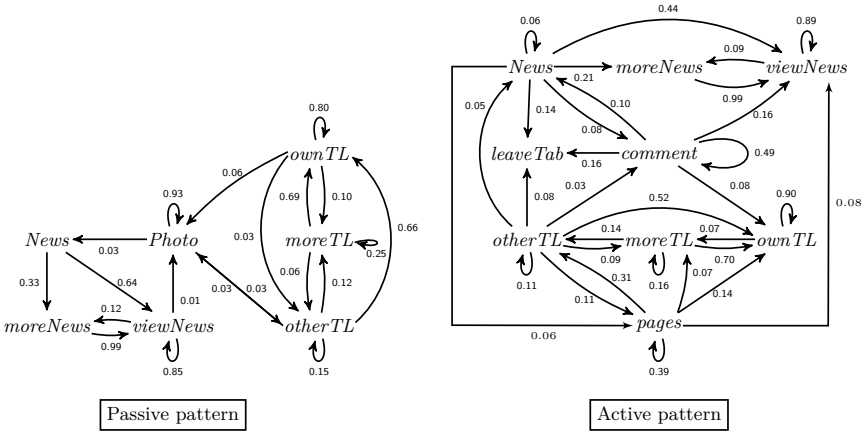

Fig. 4: An examplary solution of the identified clusters; exit states are omitted, their probability equals 1 minus the sum of emission probabilities of a state.

4.2 Facebook data

In this experiment, we demonstrate how the model can be applied for information extraction tasks from huge dataset. This is especially useful for tasks that come with no or only little prior knowledge. The data set for the next evaluation contains user navigation data from Facebook [18]. For each user, the invoked pages are recorded and grouped into sessions. Examples for such invoked pages are *'Login', 'Newsfeed', 'Load more news', 'Like', etc.* The dataset contains 152 unique invoked pages, 49,479 sessions of 2,749 users, and 8,197,308 observations. Every session is interpreted as a sequence of observations.

The most frequently observed behavioral pattern is user's checking for updates on the newsfeed by *waking up* the device and, without performing any additional activity, *put to sleep* shortly after. Figure 4 depicts two more complex patterns of users on Facebook. The first pattern, on the left, describes passive user behavior without any direct communication. Users following this patterns tend to look at their newsfeed (*News*) or at their own timeline (*ownTL*). While updating (represented by the loop on *ownTL*) or scrolling (*moreTL*) their own timeline, they would sometimes be interested in someone else's time-line (*otherTL*). There, they scroll through it but will most likely go back to their own timeline. They tend to look at more entries (*viewNews*) from their newsfeed and interact (self-loop) with them. If they open a gallery (*Photo*), they would look at several pictures (self-loop on *Photo*) before returning to their previous activity.

The pattern in the right part of 4 describes a more active behavior. While also browsing their Facebook universe, users frequently *comment* on newsfeeds' and timelines' entries. Additionally, the users visit fan and company pages (*pages*). The iMMC algorithm successfully distinguished different session behaviors without any prior knowledge on the data, nor dependencies between events.

4.3 Electronic text books

In this section, we present insights on the usage patterns of students interacting with an electronic text book for history called the mBook [6]. We show that identified usage patterns correlate with psychometric scores.

Among others, the mBook has been successfully deployed in the German-speaking community of Belgium. Together with psychologists and didacticians, we aim to evaluate the pros and cons of daily use in classrooms on children and teachers. In addition to an event log that tracks all user actions in the book, demographic variables as well as variables measuring competencies and interest are regularly assessed. Since 2013, about $3,000$ users have created $370,000$ sessions. In this experiment, we focus on 803 sessions of a subset of 286 users between February and March 2017. Our aim is to identify characteristic usage patterns to later search for correlation with psychometric variables.

Related studies reveal that time-on-page and cursor trajectories often serve as indicators for student engagement [21,22]. However, in our case, the text book is mainly used on tablets in class rooms and, hence, cursors or eye tracking are not available. We thus aim to identify alternative indicators that are precise enough to capture characteristic traits of different behavior. We define and differentiate 75 atomic events that a user can trigger, ranging from pressing a button to various scrolling performances. The latter are further divided into 9 events : *scroll.direction.duration*. The direction can be *up*, *down* or *fix* if the movement is of less than 10 pixels. The duration can be *fast*, *medium* or *slow* for event duration of respectively less than 1 second, between 1 and 3 seconds and more than 3 seconds. In the following, node names will be abbreviated using only the first letter. For example a *scroll.down.fast* is reduced to *d.f.*

In contrast to the analysis of the Facebook data set, where the huge amount of data allowed for a deployment of MMC, in this case a MMC would fail due to the lack of a sufficient amount of data; information criteria are known to perform poorly when the sample size is smaller than the number of parameters [20] as shown in Figure 5 (left). The evolution of three information criteria AIC [29], AICc [26], and BIC [30] is depicted for different numbers of clusters where every point in the figure denotes the best result out of 30 repetitions. Theoretically, the minima of these curves are supposed to give the optimal solutions given the involved parameters. Due to the ill-posed optimization problem, however, the criteria grow almost linearly. The AIC curves reaches a mininum for two clusters, what is not really interesting. Thus information criteria do not allow to draw conclusion.

By contrast, our Bayesian approach successfully clusters the data using $\gamma = 2$, $\sigma = 1.5$, $\lambda = 2.4$, $L = 100$ and $10,000$ iterations. After every $1,000$ iterations, an intermediate clustering is computed as the average of the last $1,000$ iterations. The first intermediate clustering is based on 34 clusters, the final solution settles on 32 clusters. The evolution of the solution is shown in Figure 5 (right). The blue line (left scale) represents the evolution of the normalized mutual information (NMI) relative to the final solution. The red line (right scale) refers to the entropy of the clustering for the actual iteration. After $7,000$ iterations the NMI indicates

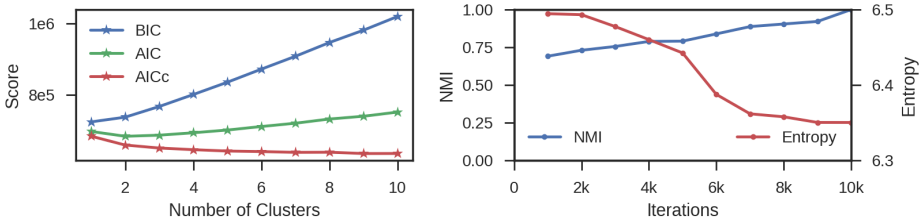

Fig. 5: Left: BIC, AIC and AICc for MMC. Right: NMI and entropy for iMMC

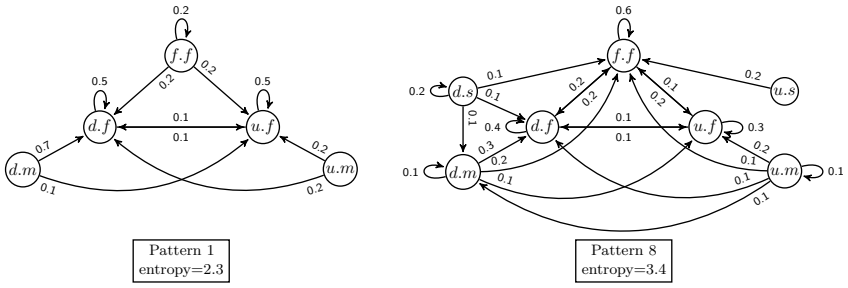

Fig. 6: Two exemplary scrolling patterns.

that the clustering is already 90% similar to the final one. The decrease in entropy shows that the algorithm merges the data into fewer clusters. The plateau after 7,000 iterations indicates fine granular changes of cluster memberships.

There are eight resulting clusters with at least 20 sessions. We focus on the scrolling events and show two patterns in Figure 6 realizing the smallest and highest entropy, respectively. Note that the weights do not sum up to one, as we ignore outgoing edges to non-scroll events in this analysis.

The first thing to notice is that in Pattern 1, *scroll.fix.** cannot be reached from another type of scroll. Either it starts a scrolling sequence or it indicates mis-usage or hesitation of the user. Although Pattern 8 is more complex, it shares

Table 2: The most strongly correlated event transitions for each score.

Score	Max Corr.	Event	Min Corr.	Event
Competence	0.697	$f.f \to u.f$	-0.719	$u.m \to u.s$
Knowledge	0.962	$d.m \to u.f$	-0.947	$d.s \to d.m$
Motivation	0.748	$f.f \to f.f$	-0.714	$f.f \to u.f$
IT Access	0.751	$d.s \to u.f$	-0.735	$f.f \to d.f$
IT Skill	0.837	$d.s \to u.f$	-0.743	$d.m \to d.s$

the fact that users tend to not transit to slower scrolls. This can be interpreted by the observed behavior that 'longer' scrolls are corrected by faster ones. This is typical behavior for users who are scrolling while reading the text on the page. This is also reflected in high self-transition probabilities of *scroll.down.slow* and *scroll.fix.fast*. Multiple ways to reach this last event are likely caused by stopping a scroll with a small scroll and keeping the finger on the tablet.

Psychometric Correlations During the four years of the experiment, the children are assessed at the end of each school year. Five factors are measured. Competency and knowledge in the field are assessed using item response theory [33,34]. Additionally, their motivation, access to digital devices and their skills in the usage of these are assessed by multiple choice questionnaires (advanced skills weight more than simple ones).

To correlate the assessed variables with our clustering, we represent clusters by the average score of all children who have sessions in the cluster. We compute Pearson correlation coefficients [26] that are adjusted for small sample sizes for the 81 possible transition probabilities between scroll events and the 8 resulting clusters with at least 20 elements.

The maximum and minimum correlations for the assessed variables are reported in Table 2. Except for motivation, high correlated transitions for every variable end with a *scroll.up.fast* and a change in direction. Knowledge has a correlation of almost 1.0 with *scroll.down.medium* $\rightarrow$ *scroll.up.fast*, and of nearly -1.0 with *scroll.down.slow* $\rightarrow$ *scroll.down.medium*. Pattern 8 is the only pattern containing these two edges. However, the correlations cancel out in the final result. Figure 7 confirms that cluster 8 loads only weakly on knowledge compared to the others.

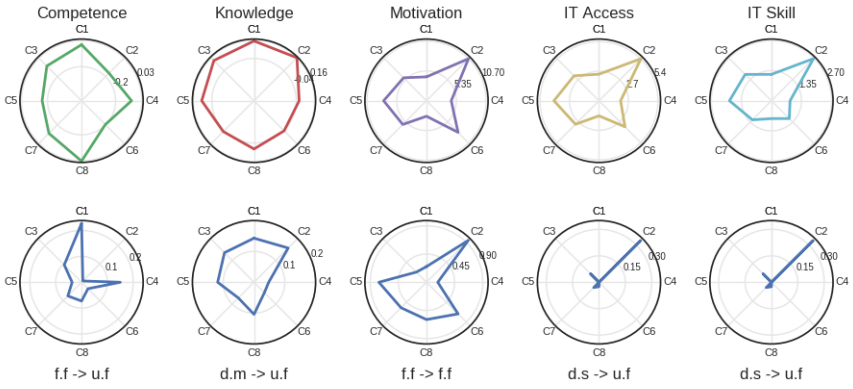

Fig. 7: Scores and probabilities of their most correlated transition for the 8 biggest clusters.

The first row in Figure 7 shows the loadings for the 8 biggest clusters. The clusters are organized from top to bottom according to their entropy.

Patterns 1 and 8 (see Fig. 6) are extracted from clusters 1 and 8, respectively. Both patterns are often observed by pupils with high competencies in history. Therefore, these patterns may serve as behavioral indicators for a user's competency. This finding is supported by the high correlation of cluster 1 with the prior knowledge of the user. Seemingly, knowledgeable children prefer simpler scrolling patterns. By contrast, cluster 2 contains highly motivated children that possess high computer skills. The pupils in cluster 6 are also motivated but do not possess such a high ICT literacy and thus do not know to handle electronic devices that well.

The second row in Figure 7 displays the values among clusters of the most strongly correlated transitions to the corresponding score. Negative correlations are not shown for interpretability. These plots give an impression of the correlations. For knowledge and motivation scores, the probability of *scroll.down.medium* $\rightarrow$ *scroll.up.fast* and *scroll.fix.fast* $\rightarrow$ *scroll.fix.fast* could be used to predict their respective scores in the assessment. With respect to competence, a high transition probability seemingly also implies a high score in the assessment. However the opposite does not hold true. Cluster 8, as also seen in Figure 6, has a smaller probability of transitioning from *scroll.fix.fast* to *scroll.up.fast*, although the average competency score of the cluster is the largest.

Our results show for the first time that behavioral indicators in electronic text books can be identified to discriminate between children. Results like this will have a high impact on the next generations of electronic text books so that they become adaptive and provide individual learning environments for every child.

5 Conclusion

We presented a Bayesian nonparametric approach to modeling user behavior. The nonparametric nature of our approach allowed for the efficient identification of the underlying clusters within user event data. Our model showed significant improvements over related approaches when analyzing such data. We obtained a natural state-duration model by capturing end-state distributions of the clusters. The models allowed us to capture state durations based on the dynamics of the cluster. Furthermore, representing each cluster as a Markov chain led to easily interpretable results that may impact design decisions and future developments of the respective service.

Acknowledgements

This research has been funded in parts by the German Science Foundation DFG under grant GRK/1907 and by the German Federal Ministry of Education and Science BMBF under grant QQM/01LSA1503C.

References

1. Mitchell, A., Olmstead, K., Purcell, K., Rainie, L., Rosenstiel, T: Understanding the participatory news consumer. (2010)
2. Teh, Y. W., Jordan, M. I., Beal, M. J., Blei, D. M.: Hierarchical Dirichlet processes. Journal of the American Statistical Association Vol. 101 No. 476 (2006)
3. Sethuraman, J: A constructive definition of Dirichlet priors. Statistica Sinica, 693–650 (1994)
4. Cadez, I., Heckerman, D., Meek, C., Smyth, P., White, S.: Visualization of navigation patterns on a web site using model-based clustering. Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 280–284 (2000)
5. Ishwaran, H., Zarepour, M.: Exact and approximate sum representations for the Dirichlet process. The Canadian Journal of Statistics/La Revue Canadienne de Statistique, 269–283 (2002)
6. Schreiber, W., Sochatzy, F., Ventzke, M.: Das multimediale Schulbuch - kompetenzorientiert, individualisierbar und konstruktionstransparent. In Analyse von Schulbüchern als Grundlage empirischer Geschichtsdidaktik, 212–232 (2013)
7. Pirolli, P. L., Pitkow, J. E.: Distributions of surfers' paths through the world wide web: Empirical characterizations. World Wide Web, 2(1-2):29–45 (1999)
8. Lau, T., Horvitz, E.: Patterns of search: Analyzing and modeling web query refinement. UM99 User Modeling, Springer Vienna, 119–128 (1999)
9. Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, K.: The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence (UAI), 256–265 (1998)
10. Manavoglu, E., Pavlov, D., Giles, C. L.: Probabilistic user behavior models. ICDM 2003, Third IEEE International Conference on Data Mining. IEEE (2003)
11. Ypma, A., Heskes, T.: Automatic categorization of web pages and user clustering with mixtures of hidden Markov models. In International Workshop on Mining Web Data for Discovering Usage Patterns and Profiles, 35–49 (2002)
12. Deshpande, M., Karypis, G.: Selective Markov models for predicting Web page accesses. ACM Transactions on Internet Technology (TOIT), 4(2), 163–184 (2004)
13. Mochihashi, D., Sumita, E.: The Infinite Markov Model. In NIPS, 1017–1024 (2007)
14. Bühlmann, P., Wyner, A. J.: Variable length Markov chains. The Annals of Statistics, 27(2), 480–513 (1999)
15. Begleiter, R., El-Yaniv, R., Yona, G.: On prediction using variable order Markov models. Journal of Artificial Intelligence Research, 22, 385–421 (2004)
16. Dubey, A., Hwang, S., Rangel, C., Rasmussen, C. E., Ghahramani, Z., Wild, D. L.: Clustering protein sequence and structure space with infinite Gaussian mixture models. In Pacific Symposium on Biocomputing, 399–410 (2003)
17. Brown, D. P.: Efficient functional clustering of protein sequences using the Dirichlet process. Bioinformatics, 24(16), 1765–1771 (2008)
18. Paul, T., Puscher, D., Strufe, T.: Improving the Usability of Privacy Settings in Facebook. In CoRR (2011)
19. Du, N., Farajtabar, M., Ahmed, A., Smola, A. J., Song, L.: Dirichlet-Hawkes processes with applications to clustering continuous-time document streams. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 219–228 (2015)
20. Giraud, C.: Introduction to high-dimensional statistics. Vol. 138. CRC Press, (2014)

21. Cocea, M., Weibelzahl, S.: Cross-system validation of engagement prediction from log files. European Conference on Technology Enhanced Learning. Springer Berlin Heidelberg, (2007)
22. Salmeron-Majadas, S., Santos, O. C., Boticario, J. G.: Exploring indicators from keyboard and mouse interactions to predict the user affective state. Educational Data Mining (2014)
23. Kurihara, K., Welling, M., Teh, Y. W.: Collapsed Variational Dirichlet Process Mixture Models. In IJCAI Vol. 7, pp. 2796-2801 (2007)
24. Fox, E. B., Sudderth, E. B., Jordan, M. I., Willsky, A. S.: A sticky HDP-HMM with application to speaker diarization. The Annals of Applied Statistics, 1020-1056 (2011)
25. Ferguson, T. S.: A Bayesian analysis of some nonparametric problems. The annals of statistics, 209-230 (1973)
26. Olkin, I.; Pratt, J. W.: Unbiased estimation of certain correlation coefficients. The Annals of Mathematical Statistics, S. 201-211 (1958)
27. Haider, P., Chiarandini, L., Brefeld, B.: Discriminative clustering for market segmentation. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, (2012)
28. Dempster, A. P., Laird, N. M. , Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society. Series B (methodological) 1–38 (1977)
29. Akaike, H.: A new look at the statistical model identification. IEEE transactions on automatic control 19.6 716–723 (1974)
30. Schwarz, G.: Estimating the dimension of a model. The annals of statistics 6.2 461–464 (1978)
31. Roberts, G. O., Smith, A.: Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. Stochastic processes and their applications 49.2 207–216 (1994)
32. Blei, D. M., Ng, A. Y., Jordan, M. I.: Latent Dirichlet allocation. Journal of machine Learning research, 3(Jan), 993-1022 (2003)
33. Baker, Frank B.: The basics of item response theory. For full text: http://ericae.net/irt/baker., (2001)
34. DeMars, Christine: Item response theory. Oxford University Press, (2010)