
The Segmented iHMM: A Simple, Efficient Hierarchical Infinite HMM

Ardavan Saeedi

Computer Science and Artificial Intelligence Laboratory, MIT

ARDAVANS@MIT.EDU

Matthew Hoffman

Adobe Research

MATHOFFM@ADOBE.COM

Matthew Johnson

Harvard University

MATTJJ@CSAIL.MIT.EDU

Ryan Adams

Harvard University and Twitter

RPA@SEAS.HARVARD.EDU

Abstract

We propose the segmented iHMM (siHMM), a hierarchical infinite hidden Markov model (iHMM) that supports a simple, efficient inference scheme. The siHMM is well suited to segmentation problems, where the goal is to identify points at which a time series transitions from one relatively stable regime to a new regime. Conventional iHMMs often struggle with such problems, since they have no mechanism for distinguishing between high- and low-level dynamics. Hierarchical HMMs (HHMMs) can do better, but they require much more complex and expensive inference algorithms. The siHMM retains the simplicity and efficiency of the iHMM, but outperforms it on a variety of segmentation problems, achieving performance that matches or exceeds that of a more complicated HHMM.

1. Introduction

The infinite hidden Markov model (iHMM) (Beal et al., 2001) and its variants (e.g., Fox et al. (2008); Saeedi & Bouchard-Côté (2011)) have been among the most successful Bayesian nonparametric models, with applications from speech recognition (Fox et al., 2011) to biology (Beal & Krishnamurthy, 2012). However, despite their success in modeling time series with complicated low-level dynamics, their application to time series with multiple timescales has been limited.

Such hierarchically structured sequences characterized by

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48. Copyright 2016 by the author(s).

multiple timescales arise in many domains, including natural language (Lee et al., 2013), handwriting (Lake et al., 2014), and motion recognition (Heller et al., 2009). For example, it is natural in motion recognition to model the sequence of high-level actions (such as walking to a chair, then sitting down) and steps within the actions (e.g., bending one’s knees then leaning back to sit down) at two different levels.

We will focus on the problem of *segmentation*, in which the goal is to identify points at which a time series transitions from one relatively stable regime to a new regime. In the motion recognition example, the segmentation problem would be to identify when a subject transitioned from one type of action (e.g., walking) to another (e.g., sitting down), without necessarily identifying what they were doing. This is one of the easiest problems in time-series modeling that involves multiple timescales, but (as we will see) it is quite hard for (i)HMMs, which have no mechanism for distinguishing between high- and low-level dynamics.

The hierarchical HMM (HHMM) is a generalization of the HMM that naturally deals with dynamics at multiple timescales (Fine et al., 1998; Murphy & Paskin, 2002). But this generality comes at a price: these models lack the simple predictive distributions and efficient inference schemes that make (i)HMMs so popular. And the available nonparametric versions of the HHMM (e.g., Heller et al., 2009; Stepleton et al., 2009) are complex to implement and not readily amenable to efficient inference.

In this paper, we propose the segmented iHMM (siHMM), a simple extension to the iHMM that does not suffer from the above problems and can discover segment boundaries in time series with dynamics at two timescales. Unlike the HHMM, our model does not explicitly model higher-level state; instead, it assumes dynamics that evolve according

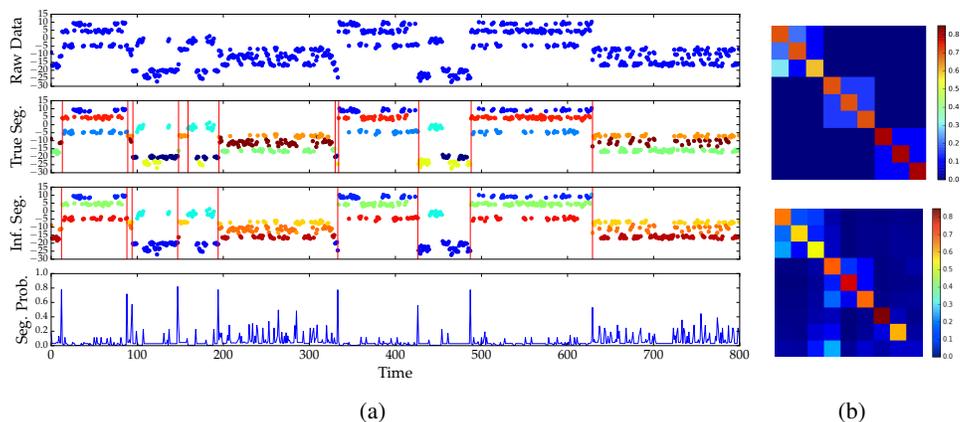


Figure 1. (a) A sample of a synthetic dataset with true and inferred segmentations. Top three plots: the y-axis shows 1-d observations, colors denote true or inferred hidden state, vertical lines denote true or inferred segment boundaries. Bottom plot: inferred posterior probability of a segment boundary. (b) **Top** True and **Bottom** inferred transition matrices

to a standard iHMM except for occasional change-point events that kick the model into a new randomly chosen hidden state, disrupting the low-level dynamics of the iHMM. Because it relies on a very simple model of high-level dynamics, inference in the siHMM has time and implementation complexity similar to that of the iHMM, and well below that of a typical HHMM. We show that this simple change-point extension is sufficient to encourage the iHMM to model time-series data characterized by multiple regimes of low-level dynamics. Although our model is limited by the depth of the hierarchy, in many practical applications of HHMMs (e.g., [Olivera et al. \(2004\)](#); [Nguyen et al. \(2005\)](#); [Xie et al. \(2003\)](#)) a two-level analysis of the dynamics is sufficient.

Below, we describe two versions of the model. The first version, which we call the feature-independent model, enjoys conditional conjugacy and therefore has simple Gibbs and variational inference algorithms. The second version, which we call the feature-based model, can incorporate domain knowledge without requiring a complex new machinery. We present a stochastic variational inference (SVI) scheme for the feature-based model; the derivation for the feature-independent model is similar and straightforward.

We apply the model to three different tasks: a novel task of segmenting traces of user behavior in software applications, automatic behavioral segmentation of fruit fly and biometric sensor data labeling. Segmenting user behavior traces is of significant importance in understanding the behavior of software application users; it can help in identifying and simplifying the complex common patterns among the users (e.g., [Adar et al. \(2014\)](#); [Han et al. \(2007\)](#); [Horvitz et al. \(1998\)](#)). For the fruit fly behavior segmentation, we use a dataset from [Kain et al. \(2013\)](#); the results of this task

can be used to better understand how the nervous system generates behavior. Finally, labeling sensor data gathered in everyday life settings can be used not only to understand physical activities (e.g., [Ermes et al. \(2008\)](#); [Pärkkä et al. \(2006\)](#)), but also to detect psychological and emotional states (e.g., [Picard et al. \(2001\)](#); [Healey & Picard \(2005\)](#)). Such labels can aid in implementing effective health and well-being related interventions, understanding user behavior, and designing affective interfaces.

We empirically compare our model with two main baselines: 1) a two-level Bayesian nonparametric hierarchical HMM (HHMM) introduced by [Johnson \(2014\)](#) that models high-level dynamics as an infinite hidden semi-Markov model (HSMM) and sub-dynamics as an iHMM, and 2) the iHMM. In each of these tasks, we show that our model outperforms the nonparametric HHMM (despite being substantially simpler and faster) and the iHMM.

2. Model

Our model can be viewed as a generalization of an iHMM where the transition probability from each state k is a mixture of two distributions: 1) a state-dependent transition probability distribution π_k , as in an iHMM, and 2) a state-independent probability distribution π_0 . Which of these two distributions generates a hidden state z_t at a given time t depends on the hidden state z_{t-1} and observation y_{t-1} at the previous time $t-1$.

We say that the transitions caused by π_0 define the boundaries of a *segment*. The model implicitly assumes that the low-level dynamics within a segment are more structured and predictable than the higher-level dynamics that govern transitions between segments, since it can throw much

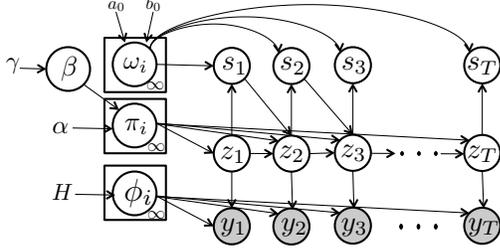


Figure 2. Graphical representation of the feature-independent siHMM.

more modeling power at these low-level dynamics. In motion capture data, for instance, the dynamics of a walk may be highly structured and predictable, whereas the dynamics that govern whether a user transitions from walking to standing, sitting, or running may be much less predictable.

2.1. Feature-independent model

The feature-independent model assumes the following generative process. At time step $t = 0$, we initialize the process by sampling a hidden state z_0 from a distribution π_0 . Given a hidden state z_t , we generate an observation y_t from a conditional observation distribution $F(\phi_{z_t})$ where ϕ_{z_t} is the parameter corresponding to the hidden state z_t : $y_t | z_t \sim F(\phi_{z_t})$.

Next, we sample a variable s_t , which we call the segmentation variable, from a Bernoulli distribution with a parameter ω_{z_t} . This is a state-dependent variable which has a conjugate beta prior with hyperparameters a_0 and b_0 . Here, $s_t = 1$ denotes the beginning of a new segment:

$$\omega_i \sim \text{Beta}(a_0, b_0); \quad s_t | z_t, y_t \sim \text{Bern}(\omega_{z_t}).$$

We denote the probability of creating a new segment at time t by p_t^{seg} . If $s_t = 0$, we sample the next state z_{t+1} from a state-dependent distribution π_{z_t} (as in the iHMM), otherwise, we ignore the current state z_t and sample z_{t+1} from a distribution π_0 :

$$z_{t+1} | z_t, s_t = 0 \sim \pi_{z_t}, \quad z_{t+1} | z_t, s_t = 1 \sim \pi_0.$$

The transition matrix has the same generative process as the iHMM:

$$\beta \sim \text{GEM}(\gamma); \quad \pi_i \sim \text{DP}(\alpha\beta); \quad \phi_i \sim H;$$

where H is the prior distribution over ϕ , $\text{GEM}(\gamma)$ is the stick-breaking distribution with concentration parameter γ , and $\text{DP}(\alpha\beta)$ denotes sampling from a Dirichlet process with concentration parameter α . The graphical model of the feature-independent siHMM is depicted in Fig. 2.

An illustration of the model applied to a synthetic dataset (explained in Section 5.1) is provided in Fig. 1. The model

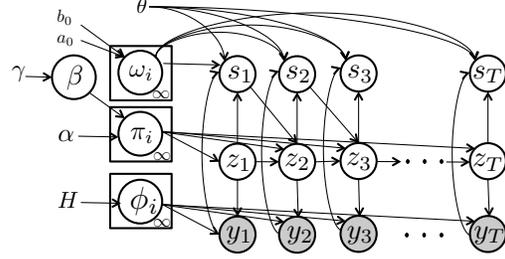


Figure 3. Graphical representation of the feature-based siHMM.

is able to approximately recover the block-diagonal structure of the true transition matrix. Even though the model does not explicitly encourage block-diagonal structure, the sparsity induced by the DP prior on π_k is sufficient to encourage the model to push inter-segment dynamics into π_0 and recover the block-diagonal intra-segment dynamics.

2.2. Feature-based model

In some tasks such as segmenting software user traces or tagging fruit fly behavior, there is a rich domain knowledge available for improving the model. For instance, in segmenting user traces, features like $\mathbb{1}(\text{action} = \text{save})$ or $\mathbb{1}(\text{action} = \text{close})$ may indicate the end of a segment. We modify the model in a way that we can add features declaratively. Although due to lack of conjugacy, deriving the Gibbs sampler is not straightforward anymore, in Section 3, we derive an efficient SVI algorithm for this model.

The difference between this version of the model and the feature-independent version is in the conditional distribution of the segmentation variable s_t (see Fig. 3 for the graphical model). Here, the parameter of the Bernoulli distribution is

$$\sigma(\theta \cdot f(y) + \omega_{z_t}) = \frac{1}{1 + \exp(-\theta \cdot f(y) - \omega_{z_t})}$$

where θ is the weight vector for the data-dependent features, $f(y)$ is the feature function which consists of all observation-dependent features, and ω_{z_t} is the feature weight for hidden state z_t . To simplify the notation, we assume that the observation-dependent features only depend on the observation at a single time step. Hence, we have

$$s_t | z_t, y_t \sim \text{Bern}(\sigma(\theta \cdot f(y_t) + \omega_{z_t})).$$

We do not assume a prior for the feature weights; instead, we use a point estimate for them in our SVI algorithm.

3. Stochastic variational inference

To keep the notation uncluttered, we assume that we have a dataset y of K sequences all with the same length T and

write: $\mathbf{y} = \{y_{1:T}^k\}_{k=1}^K$, $\mathbf{z} = \{z_{1:T}^k\}_{k=1}^K$, $\mathbf{s} = \{s_{1:T}^k\}_{k=1}^K$. For inference, we use the stochastic variational inference (SVI) algorithm (Hoffman et al., 2013) and approximate the posterior with a truncated variational distribution introduced in Johnson & Willsky (2014). We approximate the posterior $p(\mathbf{z}, \mathbf{s}, \beta, \theta, \omega, \pi, \phi | \mathbf{y})$ with a mean-field family distribution $q(\mathbf{z}, \mathbf{s})q(\beta)q(\omega)q(\theta)q(\pi)q(\phi)$. In the language of SVI, \mathbf{z} and \mathbf{s} are local variables and $\beta, \omega, \theta, \pi$, and ϕ are global variables. We maximize the marginal likelihood lower bound \mathcal{L} :

$$\mathcal{L} \triangleq \mathbb{E}_q \left[\ln \frac{p(\mathbf{z}, \mathbf{s}, \beta, \omega, \theta, \pi, \phi, \mathbf{y})}{q(\mathbf{z}, \mathbf{s})q(\beta)q(\omega)q(\theta)q(\pi)q(\phi)} \right]$$

by using stochastic natural gradient ascent over the global factors and standard mean-field updates for the local factors. At each iteration of SVI, we sample a minibatch of M sequences from the dataset and update its local factors; next, given the expectation with respect to the local factors, we update the global factors by taking a step of size ρ in the approximate natural gradient direction. To further simplify the notation, we assume that the minibatch is a single sequence and drop the superscript for y, z and s . Next, we explain the variational factors for each of the variables.

3.1. Variational factors

For $q(z_{1:T}, s_{1:T})$, the ‘‘direct assignment’’ truncation used by Johnson & Willsky (2014), sets $q(z_{1:T}, s_{1:T}) = 0$ if for any of z_1 to z_T we have $z_t = k$ and $k > K$; here K is the truncation level. Since by using this truncation the update to $q(\beta)$ conditioned on the other factors is not conjugate anymore, we use a point estimate for $q(\beta)$: $q(\beta) = \delta_{\beta^*}(\beta)$. We adopt the same point estimate approach for the parameters of the sigmoid function; hence, $q(\theta) = \delta_{\theta^*}(\theta)$ and $q(\omega_z) = \delta_{\omega_z^*}(\omega_z)$.

With this truncation scheme, we can write the prior over π_i as $p((\pi_{i1}, \dots, \pi_{iK}, \pi_{i,\text{rest}})) = \text{Dir}(\alpha\beta_1, \dots, \alpha\beta_K, \alpha\beta_{\text{rest}})$. Here, $\pi_{i,\text{rest}} = 1 - \sum_{k=1}^K \pi_k$ and $\beta_{\text{rest}} = 1 - \sum_{k=1}^K \beta_k$. The optimal $q((\pi_{i1}, \dots, \pi_{iK}, \pi_{i,\text{rest}}))$ is of the form of $\text{Dir}(\tilde{\alpha}_i)$ where $\tilde{\alpha}_i$ is the parameter of the variational distribution (Hoffman et al., 2013).

We assume that the prior over ϕ is in an exponential family with natural parameter η , and it is a conjugate prior for the likelihood function $f(y_t | \phi)$. This implies that the optimal variational distribution $q(\phi)$ is also in the same family with some other natural parameter denoted by $\tilde{\eta}$. More formally, we have: $h(\phi_i) \propto \exp\{\langle \eta_i, t_\phi(\phi_i) \rangle\}$ and $q(\phi_i) \propto \exp\{\langle \tilde{\eta}_i, t_\phi(\phi_i) \rangle\}$ where t_ϕ is the sufficient statistic function of $p(\phi)$.

3.2. SVI update equations

For the variational updates, we need to take expectations with respect to each of the variational distributions. For the

expectations with respect to $q(z_{1:T}, s_{1:T})$, a modification of the standard HMM forward-backward algorithm with the following forward F and backward B messages can be used:

$$\begin{aligned} F(z_t, s_t) &\triangleq f(y_t | \phi_{z_t})p(s_t | z_t, y_t) \times \\ &\sum_{z_{t-1}, s_{t-1}} F(z_{t-1}, s_{t-1})p(z_t | s_{t-1}, z_{t-1}); \\ B(z_t, s_t) &\triangleq \sum_{z_{t+1}, s_{t+1}} B(z_{t+1}, s_{t+1}) \times \\ &f(y_{t+1} | \phi_{z_{t+1}})p(s_{t+1} | z_{t+1}, y_{t+1})p(z_{t+1} | s_t, z_t). \end{aligned} \quad (1)$$

These messages can be computed in $O(TK^2)$. In fact, the augmented transition matrix that we need to compute for these forward-backward messages has the following form:

$$\begin{array}{l} s_t = 0 \\ s_t = 1 \end{array} \left[\begin{array}{c|c} s_{t+1} = 0 & s_{t+1} = 1 \\ \hline (1 - p_{t+1}^{\text{seg}})\Pi & p_{t+1}^{\text{seg}}\Pi \\ \hline (1 - p_{t+1}^{\text{seg}})\mathbf{1}\pi_0^T & p_{t+1}^{\text{seg}}\mathbf{1}\pi_0^T \end{array} \right]$$

where Π is a $K \times K$ transition matrix and $\mathbf{1}$ is an all-ones vector of size K . The matrix operation for computing a message requires $2K^2$ operations for the upper half of the matrix ($s_t = 0$) and $2K + K$ for the lower half ($s_t = 1$). This is because all the rows of each $K \times K$ block in the lower half are the same. Hence, the total number of operations for message passing is $T(2K^2 + 3K)$. Furthermore, the total memory required to compute these operations is $O(TK)$.

For updating the local factors, instead of π_{ij} and $f(y_t | \phi_{z_t})$ in Eq.1, we compute the forward-backward messages using: $\tilde{\pi}_{ij} \triangleq \exp\{\mathbb{E}_{q(\pi)} \ln \pi_{ij}\}$ and $\tilde{L}_{ti} \triangleq \exp\{\mathbb{E}_{q(\phi_i)} \ln f(y_t | \phi_{z_t}, z_t = i)\}$

The expectations of the sufficient statistics with respect to $q(z_{1:T}, s_{1:T})$ are:

$$\begin{aligned} \tilde{t}_{\text{trans}}^{ij} &\triangleq \mathbb{E}_{q(z_{1:T}, s_{1:T})} \sum_{t=2}^T \mathbf{1}[z_{t-1} = i, z_t = j, s_{t-1} = 0, s_t = .]; \\ \tilde{t}_{\text{trans}}^{0j} &\triangleq \mathbb{E}_{q(z_{1:T}, s_{1:T})} \left[\mathbf{1}[z_1 = j] + \sum_{t=2}^T \mathbf{1}[z_t = j, s_{t-1} = 1] \right]; \\ \tilde{t}_y^j &\triangleq \mathbb{E}_{q(z_{1:T}, s_{1:T})} \sum_{t=1}^T \mathbf{1}[z_t = j, s_t = .] t_y^j(y_t). \end{aligned} \quad (3)$$

Given these expected sufficient statistics for S sequences and minibatch size M , we can write the update equations for the parameters of the global variational factors $q(\pi)$ and $q(\phi)$ with a scaling factor $m \triangleq S/M$:

$$\begin{aligned} \tilde{\eta}_i &\leftarrow (1 - \rho)\tilde{\eta}_i + \rho(\eta_i + m.\tilde{t}_y^i) \\ \tilde{\alpha}_i &\leftarrow (1 - \rho)\tilde{\alpha}_i + \rho(\alpha_i + m.\tilde{t}_{\text{trans}}^i) \\ \tilde{\alpha}_0 &\leftarrow (1 - \rho)\tilde{\alpha}_0 + \rho(\alpha_0 + m.\tilde{t}_{\text{trans}}^0). \end{aligned}$$

For the global factors $q(\omega)$, $q(\theta)$ and $q(\beta)$, we use a point estimate; hence, we only need the gradient of \mathcal{L} with respect to β^* , θ^* and ω^* . For ∇_{β^*} we follow the derivation in Johnson & Willsky (2014) and obtain the following gradient to use in a truncated gradient step on β^* :

$$\begin{aligned}\nabla_{\beta^*} \mathcal{L} &= \nabla_{\beta^*} \left\{ \mathbb{E}_{q(\pi)} \left[\ln \frac{p(\beta, \pi)}{q(\beta)q(\pi)} \right] \right\} \\ &= \nabla_{\beta^*} \left\{ \ln p(\beta^*) + \sum_{i=1}^K \mathbb{E}_{q(\pi_i)} \ln p(\pi_i | \beta^*) \right\}\end{aligned}$$

Note that, we need to ensure that $\beta^* \geq 0$ after each update. To estimate θ^* , we have:

$$\begin{aligned}\nabla_{\theta^*} \mathcal{L} &= \nabla_{\theta^*} \left\{ \mathbb{E}_{q(z_{1:T}, s_{1:T})} \left[\ln \frac{p(\theta, z_{1:T}, s_{1:T}, y_{1:T})}{q(\theta)q(z_{1:T}, s_{1:T})} \right] \right\} \\ &= \mathbb{E}_{q(z_{1:T}, s_{1:T})} \left[\sum_{t=1}^T s_t f(y_t) - p(s_t = 1 | y_t, z_t) f(y_t) \right].\end{aligned}$$

We have a similar equation for ω_z^* . This gradient computation has time complexity of $O(TK)$.

4. Related models

There are few models similar to our model in terms of extending iHMM to multiple timescales. Infinite hierarchical HMM (iHHMM), introduced in Heller et al. (2009), is a nonparametric model that allows the HHMM to have potentially unbounded depth. Hence, the model can infer the number of levels in the hierarchy. In iHHMM, the bottom level is the observed sequence and each level is a sequence of hidden variables dependent on the level above. As the authors suggested, more efficient inference algorithms are needed in order to make their model useful for practical applications.

The block-diagonal iHMM (Stepleton et al., 2009) is a generalization of the iHMM that assumes a nearly block-diagonal structure on the transition matrix of the iHMM. Each block corresponds to a ‘‘sub-behavior’’ and the model can partition the data sequences according to these sub-behaviors. The model first partitions the infinite number of hidden states into an infinite number of blocks by using an additional stick-breaking process. Then, it increases the probability of transition between the states of a block by modifying the Dirichlet process prior over the transitions. Hence, as the block size becomes smaller, the model behavior converges to that of iHMM. For inference, as the authors explained, achieving a fast mixing rate in their proposed inference algorithm requires implementing a nontrivial bookkeeping-intensive method. In contrast, our model is much simpler and easier to implement inference for, but it can also discover transition matrices with approximately block-diagonal structure; the segmentation events

provide a mechanism for transitioning from one group of connected states to another.

Another related model is a two-level Bayesian nonparametric HMM introduced in Johnson (2014) that models the high-level dynamics or the superstates as an HDP-HSMM. As a generalization of iHMM, HDP-HSMM can model the dwell time in each state by sampling that from a state-specific duration distribution once a state is entered. For a formal definition of the HDP-HSMM, see Johnson (2014). Given each superstate j , observations are generated according to an iHMM with parameters $\{\beta^j, \pi_k^j, \phi_k^j\}_{k=1}^{\infty}$. where x_t denotes the substate at time step t . Compared to our model, this model is much more flexible; however, the computation of forward-backward messages is less efficient. Moreover, it requires more bookkeeping for all superstates and their substates, and uses separate truncation levels for the superstates and each of the iHMMs corresponding to them. (One can set a large truncation level for all of them, but this means paying a huge computational cost for iHMMs that only require a few states. In contrast, in the siHMM we only need to set a single truncation level for the whole model.) We refer to this model as the sub-iHMM and use it as a baseline since it is a flexible Bayesian nonparametric model that supports two-level dynamics.

Finally, the iHMM or sticky HDP-HMM can be used for segmentation by treating changes in hidden state as segment boundaries (Fox et al., 2011). In the sticky HDP-HMM, a self-transition bias encourages consecutive observations to be associated with the same state. The model can capture segments in tasks such as speaker diarization; however, within a given state there are no dynamics. In other words, in the sticky HDP-HMM, the observations within a segment are independent of each other. This makes the model less appropriate for segmenting sequences with predictable intra-segment dynamics. For instance, in a software application workflow an action like `selection` needs to come before an action like `move selection`. In our iHMM experiments, we include a ‘‘sticky’’ self-transition bias ¹.

5. Experiments

We evaluate the performance of our feature-independent and feature-based models on synthetic and real datasets. We use a synthetic dataset to illustrate the advantages of our model compared to baselines.

In order to further demonstrate the capabilities of our model, we apply it to three real tasks from three different

¹We do not include a separate column in Table 2 for the sticky HPD-HMM; instead, we find the best variational lower bound for all settings of hyperparameters of the iHMM including a hyperparameter for self-transition bias.

domains: human-computer interaction, biology, and sensor data analysis. The properties of the datasets used for these tasks are provided in Table 1. In this table, the maximum number of labels is the total number of segment labels for each dataset². All datasets are accompanied with the “gold standard” labels; hence, the ground truth is available and we can compute the labeling error of our model.

As mentioned in Section 4, two reasonable baselines for our model are the two-level Bayesian nonparametric HMM and iHMM. We report the labeling error (or normalized Hamming distance in the case of the synthetic dataset) and predictive log-likelihood for our model and these baselines. To choose among different hyperparameter settings, we use the variational lower bound (VLB) as our objective measure. We show that our model, while being simpler and efficient in terms of inference, is competitive with or outperforms these baselines. For all experiments on siHMM we try both the feature-independent and feature-based models; we report the results separately in Table 2 to show the effect of including observation features in the model. In the experiments, we only try the hidden state and the observation as the features for a given time step. However, more sophisticated features can be made from the observation(s). For the details of all experiments see the supplementary material.

5.1. Synthetic data

We generate a synthetic dataset with 5000 data points from 3 different transition matrices, each with 3 hidden states. Each row of each transition matrix is sampled from a modified Dir(1) with self-transition bias of 1. The observations are sampled from normal distributions with non-conjugate separate priors $\mathcal{N}(0, 10)$ and $\Gamma^{-1}(2, 1)$ on their mean and variance parameters, respectively. The goal is to find the points where we change regimes and also to determine the dynamics within each segment of the sequence. At each time step with probability 0.05, we switch the regime.

Fig. 1 shows a sample sequence and the result of running 100 passes of SVI over the whole dataset for that sequence. For running SVI, we split the dataset into 20 sequences and use a batch of size 2. We randomly sample a sequence with length 750 to calculate the predictive log-likelihood. We report the error over the dataset for the hyperparameter setting with the highest VLB.

We run SVI with 10 different seeds for 100 iterations over a set of hyperparameters (see the supplementary material for all the settings which we considered). To compute the normalized Hamming distance between the inferred states and the true states, we use the Munkres algorithm

²For the synthetic dataset, maximum number of labels is the number of hidden states used to generate the observations

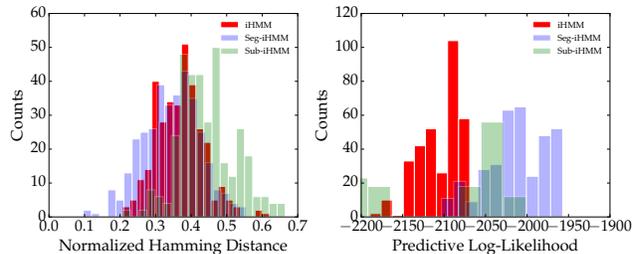


Figure 4. Histogram of normalized Hamming distance and predictive log-likelihood for different settings of hyperparameters for the synthetic data set.

(Munkres, 1957). The algorithm assigns indices to the inferred sequence so that it maximizes the overlap with the true sequence. Table 2 gives the computed distance over the dataset for the hyperparameter setting with the highest VLB. We also report the predictive log-likelihood over the held-out set.

Fig. 4 shows the histogram of the normalized Hamming distance and also the predictive log-likelihood for runs with different hyperparameter settings. In terms of the Hamming distance, the siHMM performs slightly better than the iHMM and outperforms the sub-iHMM for most settings. The same conclusion holds for the predictive log-likelihood. Furthermore, Fig. 1 shows that our model can do reasonably well in finding the regime change points and also the states within each segment of the sequence. We choose a threshold of 0.5 for the posterior segmentation probability to identify a time point as a change point. This threshold is just a natural default; it can be tuned to trade off sensitivity and specificity. Finally, as shown in the last row of Fig. 1, the model is able to provide an estimate for the posterior probability of beginning a segment at time t .

Table 2 shows the normalized Hamming distance for the synthetic dataset; siHMM outperforms both iHMM and sub-iHMM. This might be because the synthetic dataset is specifically generated without any state-dependent high-level dynamics. Our model, which does not assume any dynamics for the segments, performs better than the other baselines which implicitly or explicitly assume that.

Table 1. Datasets used for experiments (description in text)

Dataset	# Points	Held-out	Max. # Labels
Synthetic	$5e^3$	15%	9
Users	$1.4e^4$	10 %	23
Drosophila	$1e^4$	15 %	12
Sensors	$1.2e^4$	10 %	10

The Segmented iHMM

Table 2. Labeling error and predictive log-likelihood for various different datasets. **Key:** iHMM = infinite HMM; sub-iHMM = a two level hierarchical infinite HMM; siHMM (WoF) = feature-independent siHMM; siHMM (WF) = feature-based siHMM; Synthetic = synthetic data segmentation and hidden state inference tasks for which we report the normalized Hamming distance instead of error rate; Users = user trace segmentation task; Sensors = labeling sensor data task; Drosophila = segmenting fruit fly behavior task.

Data Set	(Normalized Hamming Distance / Error% & Predictive LL)			
	iHMM	sub-iHMM	siHMM (WoF)	siHMM (WF)
Synthetic	(0.21, -2.07×10^3)	(0.23, -2.24×10^3)	(0.15, -2.05×10^3)	(0.13, -2.19×10^3)
Users	(30%, -3.87×10^3)	(24%, -2.51×10^3)	(25%, -3.71×10^3)	(16%, -3.52×10^3)
Drosophila	(36%, -6.58×10^4)	(41%, -6.7×10^4)	(37%, -7.18×10^4)	(34%, -6.71×10^4)
Sensors	(27%, -3.25×10^3)	(22%, -3.95×10^3)	(18%, -3.21×10^3)	(34%, -3.35×10^3)

5.2. Segmenting user behavior traces

Modeling user behavior traces is of significant importance in human-computer interaction literature (e.g., Adar et al. (2014); Bigham et al. (2014)). Having a good understanding of the tasks done by users can potentially help in designing better work-flows in software applications. It can also help in providing better guidance to users by predicting user intentions. Log files of software applications contain user actions and their corresponding time-stamps; however, it is not clear only from these log files how many different tasks have been done by a user in a single work session. A task consists of multiple actions and each work session consists of multiple tasks. Our two-level hierarchical model can be used for detecting the boundaries between tasks (i.e., segments). We believe that for the software applications in which the tasks are less predictable compared to the actions within each task, our model is a good fit. For instance, in a photo editing software, the high-level tasks such as adding filters or removing a part of a picture usually do not have a clear order and different users, based on their needs, apply different orderings. However, the actions required to add a filter (e.g., 1- choose a filter, 2- select a part of the photo, and 3- apply the filter) are typically more structured and follow an order.

We collect log files of users who follow 23 different tutorials in a photo editing software. The dataset is in the form of time-stamp and action; it contains 14000 data points and 59 unique actions in total. The actions are directly read from the users log files and an action at time t is the observation at that time point. We randomly choose a sequence of size 1400 and form a held-out set. We split the sequences into subsequences of size 1000 and apply 100 passes of SVI to the dataset with both feature-based and feature-independent models.

The labels (i.e., the tutorial numbers that the user followed) for each segment of the dataset are available; hence, we can test the siHMM on predicting the labels for each segment. This task is more involved than segmentation, as we also need to group the substates within the inferred segments. We use a simple K-means clustering on the empirical transition matrix which is generated from counting the transitions in each inferred segment. This approach works well

in practice; however, more sophisticated methods are possible for grouping the substates. Table 2 provides the prediction error (computed using the Munkres algorithm) for siHMM and the baselines. The performance of our feature-independent model is significantly better than iHMM and comparable with that of the more flexible (and also computationally intensive) sub-iHMM. Adding the observation feature to the model reduces the error to 16%. As mentioned in Section 2.2, in this dataset there are observations that can signal a change-point in the dynamics.

5.3. Segmenting fruit fly behavior

Automating scientific experiments on live animals has attracted significant attention recently (see, for instance, Kain et al. (2013); Wiltshko et al. (2015); Crall et al. (2015); Freeman et al. (2014)). With the advent of high throughput and more accurate devices, the need for automatic analysis of large amounts of collected data is felt more today. In neuroscience and biology, a large amount of behavioral data is collected from live animals in order to understand how the brain generates activity and how the underlying mechanisms have evolved (Kain et al., 2013). Typically the first step in analyzing this data, is finding and categorizing different types of behavior; this step can be done manually by experts but it is time-consuming and sometimes error-prone.

An automatic framework has been proposed in Kain et al. (2013) for tracking the leg movements and classifying the behavior of fruit flies. The behavior is recorded by tracking each leg of a fruit fly moving upon a track ball. The collected raw data is the x and y coordinates of 6 legs and the three rotational components of the rotating ball (i.e., a 15-dimensional vector in real time). After some post-processing and adding some higher-order features (e.g., derivatives of each of the 15 raw data vectors), they expand the dimensions to 45 and apply a KNN classifier to classify each frame as a part of 12 possible behavioral labels. Our goal is to use this dataset and categorize the frames in an unsupervised way. A frequent assumption in the behavioral sciences is that a small set of stereotyped motifs describe most animal activities (Berman et al., 2014). In other words, actions within a behavioral segment (e.g., ac-

The Segmented iHMM

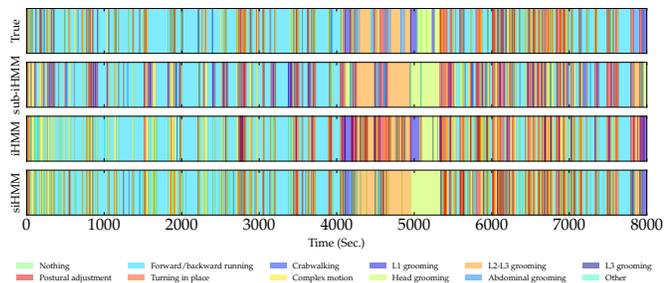


Figure 5. A sample segmentation from the fruit fly dataset. From top to bottom: True, sub-iHMM, iHMM, siHMM.

tions required for grooming) should be more structured, compared to the behavioral segments themselves. Given the capabilities of the siHMM, it is a reasonable choice for applying to this dataset.

The dataset contains 10000 data points; our held-out set is a randomly chosen subsequence with length 1500, and we apply SVI for 100 passes over both the feature-independent and feature-based models. For the observations, we use a multivariate Gaussian likelihood and a conjugate Normal/inverse-Wishart prior.

As in Section 5.2, we group the inferred substates with K-means and assign labels to the segments. The results, presented in Table 2, show that feature-based siHMM performs on a par with the iHMM and outperforms sub-iHMM by a relative error reduction of 17%. This may emphasize the importance of adding data-driven features to the model. Fig. 5, shows a sample of the dataset and its segmentation by different methods.

5.4. Segmenting and classifying human behavior from sensor data

Through the emergence of pervasive computing and affordable wearable sensors, in-situ measurement of different bio-signals has become possible. This powerful source of data can be utilized for several purposes, including activity recognition and task identification. Toward this goal, an efficient algorithm for analyzing this large amount of data – which is gathered 24/7 – is essential. In this section, we use siHMM to model the data collected via Empatica E4 wristband (Empatica, 2015), a wearable device that can collect Electrodermal activity (EDA) (Boucsein, 2012), blood volume pulse (BVP), acceleration, and body temperature. EDA refers to changes in electrical properties of the skin caused by sudomotor innervation (Boucsein, 2012). EDA is an indication of physiological or psychological arousal and has been utilized to objectively measure affective phenomena and sleep quality (Sano et al., 2015).

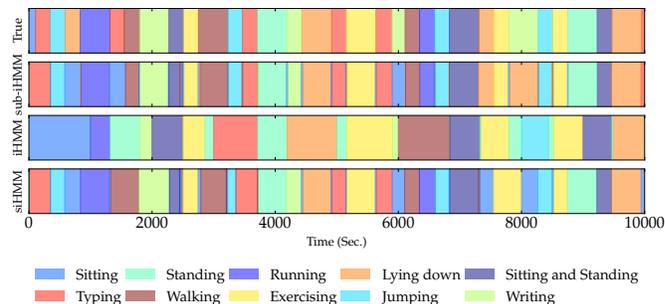


Figure 6. Segments and labels for data collected from sensors. From top to bottom: True, sub-iHMM, iHMM, siHMM.

Segmenting the sensor data can help psychophysiological activity recognition. For instance, it can help in finding stressful periods objectively in order to detect the roots of stress in a person’s lifestyle. However, manual labeling for large amounts of user data (days or months) is time-consuming and even invalid if not reported in a timely manner.

We use a dataset with 12000 time steps, collected from a single user, and model the (normalized) observations (i.e., EDA, BVP and acceleration in 3 dimensions) by a multivariate Gaussian distribution. The hyperparameter setting is similar to that of Section 5.3.

The labeling error in Table 2 shows that feature-independent siHMM, while performing comparably to sub-iHMM, outperforms iHMM by a relative error reduction of 50%. Fig. 5.4 demonstrates the inferred segments for siHMM and the baselines. It seems that the single observation feature that we are using for the experiment does not help in this dataset; however, more sophisticated features may help improving the segmentation.

6. Conclusion

We proposed a new Bayesian nonparametric model, siHMM, for modeling dynamics at two timescales in time series. Our model is a simple extension to the widely used iHMM and has an efficient inference scheme. Although our model is less flexible than other nonparametric models for hierarchical time series, we showed that it can perform reasonably well in practice. One potential application of our model is using the inferred state-independent transition vector (π_0) for summarizing a sequence. For instance, in the user behavior analysis, this vector may represent a user fingerprint and users can be grouped based on it. For a better understanding of this feature and the behavior of our model in other applications, a more comprehensive comparison with other models is useful.

Acknowledgments

Thanks to Asma Ghandeharion for her help with the sensors data and Mira Dontcheva for her feedback on segmenting user behavior traces. RPA is supported by NSF IIS-1421780 and the Alfred P. Sloan Foundation.

References

- Adar, Eytan, Dontcheva, Mira, and Laput, Gierad. Commandspace: Modeling the relationships between tasks, descriptions and features. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pp. 167–176. ACM, 2014.
- Beal, Matthew and Krishnamurthy, Praveen. Gene expression time course clustering with countably infinite hidden markov models. *arXiv preprint arXiv:1206.6824*, 2012.
- Beal, Matthew J, Ghahramani, Zoubin, and Rasmussen, Carl E. The infinite hidden markov model. In *Advances in neural information processing systems*, pp. 577–584, 2001.
- Berman, Gordon J, Choi, Daniel M, Bialek, William, and Shavit, Joshua W. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014.
- Bigam, Jeffrey P, Bernstein, Michael S, and Adar, Eytan. Human-computer interaction and collective intelligence. 2014.
- Boucsein, Wolfram. *Electrodermal activity*. Springer Science & Business Media, 2012.
- Crall, James D, Gravish, Nick, Mountcastle, Andrew M, and Combes, Stacey A. Beetag: a low-cost, image-based tracking system for the study of animal behavior and locomotion. *PLoS one*, 10(9):e0136487, 2015.
- Empatica. E4 sensor. <https://www.empatica.com/e4-wristband>, 2015. Accessed: 2015-12.
- Ermes, Miikka, Parkka, Juha, Mantyjarvi, Jani, and Korhonen, Ilkka. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, 2008.
- Fine, Shai, Singer, Yoram, and Tishby, Naftali. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.
- Fox, Emily B, Sudderth, Erik B, Jordan, Michael I, and Willsky, Alan S. An hdp-hmm for systems with state persistence. In *Proceedings of the 25th international conference on Machine learning*, pp. 312–319. ACM, 2008.
- Fox, Emily B, Sudderth, Erik B, Jordan, Michael I, and Willsky, Alan S. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pp. 1020–1056, 2011.
- Freeman, Jeremy, Vladimirov, Nikita, Kawashima, Takashi, Mu, Yu, Sofroniew, Nicholas J, Bennett, Davis V, Rosen, Joshua, Yang, Chao-Tsung, Looger, Loren L, and Ahrens, Misha B. Mapping brain activity at scale with cluster computing. *Nature methods*, 11(9):941–950, 2014.
- Han, Jiawei, Cheng, Hong, Xin, Dong, and Yan, Xifeng. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.
- Healey, Jennifer A and Picard, Rosalind W. Detecting stress during real-world driving tasks using physiological sensors. *Intelligent Transportation Systems, IEEE Transactions on*, 6(2): 156–166, 2005.
- Heller, Katherine A, Teh, Yee W, and Görür, Dilan. Infinite hierarchical hidden markov models. In *International Conference on Artificial Intelligence and Statistics*, pp. 224–231, 2009.
- Hoffman, Matthew D, Blei, David M, Wang, Chong, and Paisley, John. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- Horvitz, Eric, Breese, Jack, Heckerman, David, Hovel, David, and Rommelse, Koos. 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*, pp. 256–265. Morgan Kaufmann Publishers Inc., 1998.
- Johnson, Matthew and Willsky, Alan. Stochastic variational inference for bayesian time series models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1854–1862, 2014.
- Johnson, Matthew James. *Bayesian time series models and scalable inference*. PhD thesis, Massachusetts Institute of Technology, 2014.
- Kain, Jamey, Stokes, Chris, Gaudry, Quentin, Song, Xi-angzhi, Foley, James, Wilson, Rachel, and de Bivort, Benjamin. Leg-tracking and automated behavioural classification in drosophila. *Nature communications*, 4:1910, 2013.
- Lake, Brenden M, Lee, Chia-ying, Glass, James R, and Tenenbaum, Joshua B. One-shot learning of generative speech concepts. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society*, 2014.
- Lee, Chia-ying, Zhang, Yu, and Glass, James R. Joint learning of phonetic units and word pronunciations for asr. In *EMNLP*, pp. 182–192, 2013.
- Munkres, James. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.
- Murphy, Kevin P and Paskin, Mark A. Linear-time inference in hierarchical hmms. *Advances in neural information processing systems*, 2:833–840, 2002.
- Nguyen, Nam T, Phung, Dinh Q, Venkatesh, Svetha, and Bui, Hung. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pp. 955–960. IEEE, 2005.
- Olivera, Nuria, Gargb, Ashutosh, and Horvitz, Eric. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, 96:163–180, 2004.

- Pärkkä, Juha, Ermes, Miikka, Korpipää, Panu, Mäntyjärvi, Jani, Peltola, Johannes, and Korhonen, Ilkka. Activity classification using realistic data from wearable sensors. *Information Technology in Biomedicine, IEEE Transactions on*, 10(1):119–128, 2006.
- Picard, Rosalind W, Vyzas, Elias, and Healey, Jennifer. Toward machine emotional intelligence: Analysis of affective physiological state. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1175–1191, 2001.
- Saeedi, Ardavan and Bouchard-Côté, Alexandre. Priors over recurrent continuous time processes. In *Advances in Neural Information Processing Systems*, pp. 2052–2060, 2011.
- Sano, A et al. Discriminating high vs low academic performance, selfreported sleep quality, stress level, and mental health using personality traits, wearable sensors and mobile phones. *Body Sensor Networks (BSN)*, 2015.
- Stepleton, Thomas S, Ghahramani, Zoubin, Gordon, Geoffrey J, and Lee, Tai S. The block diagonal infinite hidden markov model. In *International Conference on Artificial Intelligence and Statistics*, pp. 552–559, 2009.
- Wiltchko, Alexander B, Johnson, Matthew J, Iurilli, Giuliano, Peterson, Ralph E, Katon, Jesse M, Pashkovski, Stan L, Abaira, Victoria E, Adams, Ryan P, and Datta, Sandeep Robert. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015.
- Xie, Lexing, Chang, Shih-Fu, Divakaran, Ajay, and Sun, Huifang. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 3, pp. III–29. IEEE, 2003.