Chapter 3

Hierarchical Dirichlet Process Hidden Semi-Markov Models

■ 3.1 Introduction

Given a set of sequential data in an unsupervised setting, we often aim to infer meaningful states present in the data along with characteristics that describe and distinguish those states. For example, in a speaker diarization (or who-spoke-when) problem, we are given a single audio recording of a meeting and wish to infer the number of speakers present, when they speak, and some characteristics governing their individual speech patterns [108, 31]. Or in separating a home power signal into the power signals of individual devices, we could perform the task much better if we were able to exploit our prior knowledge about the levels and durations of each device's power modes [66]. Such learning problems for sequential data are pervasive, and so we would like to build general models that are both flexible enough to be applicable to many domains and expressive enough to encode the appropriate information.

Hidden Markov Models (HMMs) have proven to be excellent general models for approaching learning problems in sequential data, but they have two significant disadvantages: first, state duration distributions are necessarily restricted to a geometric form that is not appropriate for many real-world data; second, the number of hidden states must be set a priori so that model complexity is not inferred from data in a way that scales with the size and complexity of the data.

Recent work in Bayesian nonparametrics has addressed the latter issue. In particular, the Hierarchical Dirichlet Process HMM (HDP-HMM) has provided a powerful framework for representing a posterior over state complexity while maintaining efficient inference algorithms [106, 5]. However, the HDP-HMM does not address the issue of non-Markovianity in real data. The Markovian disadvantage is in fact compounded in the nonparametric setting, since non-Markovian behavior in data can lead to the creation of unnecessary extra states and unrealistically rapid switching dynamics [31].

One approach to avoiding the rapid-switching problem is the Sticky HDP-HMM [31], which introduces a learned global self-transition bias to discourage rapid switching. Indeed, the Sticky model has demonstrated significant performance improvements over the HDP-HMM for several applications. However, it shares the HDP-HMM's restriction to geometric state durations, thus limiting the model's expressiveness regarding duration structure. Moreover, its global self-transition bias is shared among all states, and so it does not allow for learning state-specific duration information. The infinite Hierarchical HMM [50] indirectly induces non-Markovian state durations at the coarser levels of its state hierarchy, but it is difficult to incorporate prior information or otherwise adjust the duration model. Furthermore, constructing posterior samples from any of these models can be computationally expensive, and finding efficient algorithms to exploit problem structure is an important area of research.

These potential limitations to the HDP-HMM motivate this investigation into explicit-duration semi-Markov modeling, which has a history of success in the parametric (and usually non-Bayesian) setting. We combine semi-Markovian ideas with the HDP-HMM to construct a general class of models that allow for both Bayesian nonparametric inference of state complexity as well as general duration distributions. We demonstrate the applicability of our models and algorithms on both synthetic and real data sets.

The remainder of this chapter is organized as follows. In Section 3.2, we describe explicit-duration HSMMs and existing HSMM message-passing algorithms, which we use to build efficient Bayesian inference algorithms. We also provide a brief treatment of the Bayesian nonparametric HDP-HMM and sampling inference algorithms. In Section 3.3 we develop the HDP-HSMM and related models. In Section 3.4 we develop extensions of the weak-limit and direct assignment samplers [106] for the HDP-HMM to our models and describe some techniques for improving the computational efficiency in some settings.

Section 3.5 demonstrates the effectiveness of the HDP-HSMM on both synthetic and real data. In synthetic experiments, we demonstrate that our sampler mixes very quickly on data generated by both HMMs and HSMMs and can recover synthetic parameters. We also show that while an HDP-HMM is unable to capture the statistics of an HSMM-generated sequence, we can build HDP-HSMMs that efficiently learn whether data were generated by an HMM or HSMM. As a real-data experiment, we apply the HDP-HSMM to a problem in power signal disaggregation.

■ 3.2 Background and notation

In this section, we review three main background topics: our notation for Bayesian HMMs, conventions for explicit-duration HSMMs, and the Bayesian nonparametric HDP-HMM.

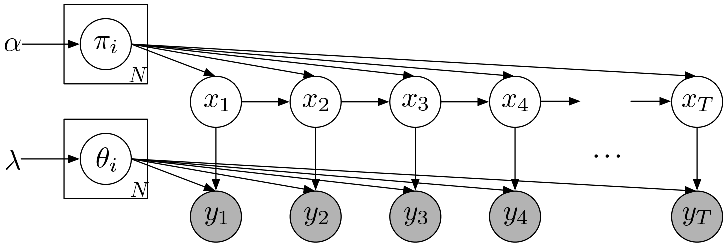


Figure 3.1: Basic graphical model for the Bayesian HMM. Parameters for the transition, emission, and initial state distributions are random variables. The symbol α represents the hyperparameter for the prior distributions on state-transition parameters. The shaded nodes indicate observations on which we condition to form the posterior distribution over the unshaded latent components.

■ 3.2.1 HMMs

Recall from Section 2.4 that the core of the HMM consists of two layers: a layer of hidden *state* variables and a layer of *observation* or *emission* variables, as shown in Figure 3.1. The hidden state sequence, $x = (x_t)_{t=1}^T$, is a sequence of random variables on a finite alphabet, $x_t \in \{1, 2, \dots, N\} = [N]$, that form a Markov chain. We focus on time-homogeneous models, in which the transition distribution does not depend on t. The transition parameters are collected into a row-stochastic transition matrix A where

$$A_{ij} = p(x_{t+1} = j | x_t = i). \tag{3.2.1}$$

We also use $\{\pi^{(i)}\}$ to refer to the set of rows of the transition matrix, and we write $\pi^{(0)}$ for the initial state distribution. We use $p(y_t|\theta^{(i)})$ to denote the conditional emission distribution, where $\theta^{(i)}$ is the emission parameter for the ith state.

■ 3.2.2 HSMMs

There are several approaches to hidden semi-Markov models [78, 118]. We focus on *explicit duration* semi-Markov modeling; that is, we are interested in the setting where each state's duration is given an explicit distribution. The basic idea underlying this HSMM formalism is to augment the generative process of a standard HMM with a random state duration time, drawn from some state-specific distribution when the state is entered. The state remains constant until the duration expires, at which point there is a Markov transition to a new state.

A kind of graphical model for the explicit-duration HSMM is shown in Figure 3.2, from Murphy [78], though the number of nodes in the graphical model is itself random and so it is not a proper graphical model. In this picture, we see there is a Markov chain (without self-transitions) on S *super-state* nodes, $(z_s)_{s=1}^S$, and these super-states

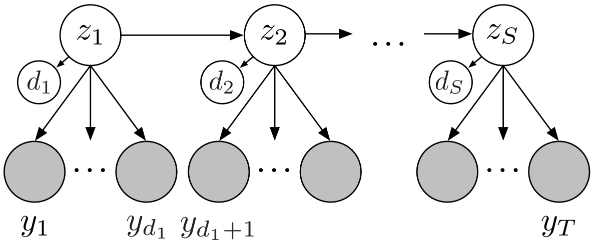


Figure 3.2: HSMM interpreted as a Markov chain on a set of super-states, $(z_s)_{s=1}^{S}$. The number of shaded nodes associated with each z_s, denoted by d_s, is drawn from a state-specific duration distribution.

in turn emit random-length *segments* of observations. We refer to the index s as the *segment index* or the *super-state index*. Here the symbol d_s is used to denote the random length of the observation segment of super-state s, and we call the sequence $(d_s)_{s=1}^{S}$ the *duration sequence*. The "super-state" picture separates the Markovian transitions from the segment durations.

We also define the *label sequence* $(x_t)_{t=1}^{T}$ in terms of the super-state sequence and duration sequence. Writing $x_{t_1:t_2} = (x_{t_1}, x_{t_1+1}, \ldots, x_{t_2})$ to denote the subsequence of x with first and last indices t_1 and t_2, respectively, the label sequence is defined by

$$x_{t(s):t(s+1)-1} = z_s \qquad s = 1, 2, \ldots, S \tag{3.2.2}$$

$$t(s) \triangleq \begin{cases} t(s-1) + d_{s-1} & s > 1 \\ 1 & s = 1 \end{cases} \qquad t = 1, 2, \ldots, T. \tag{3.2.3}$$

Therefore there is one element of the label sequence for each element of the observation sequence $(y_t)_{t=1}^{T}$. While the observation sequence length T is observed, the number of segments S is a latent variable. Note that while in some other work on HSMMs the label sequence $(x_t)_{t=1}^{T}$ is referred to as the "state sequence," it does not capture the full Markov state of the system because it does not include the required duration information; therefore, we take care to distinguish the label sequence (x_t) and the super-state sequence (z_s), and we do not refer to either as the HSMM's state sequence.

When defining an HSMM model, one must also choose whether the observation sequence ends exactly on a segment boundary or whether the observations are *censored* at the end, so that the final segment may possibly be cut off in the observations. In the right-censored case, where the final segment may be cut off at the end, we have $\sum_{s=1}^{S} d_s \leq T$. We focus on the right-censored formulation in this chapter, but our models and algorithms can easily be adapted to the uncensored or even left-censored cases. For a further discussion, see Guédon [47].

It is possible to perform polynomial-time message-passing inference along an HSMM state chain (conditioned on parameters and observations) in a way similar to the standard alpha-beta dynamic programming algorithm for standard HMMs. The "backward" messages are crucial in the development of efficient sampling inference in Section 3.4 because the message values can be used to efficiently compute the posterior information necessary to block-sample the hidden label sequence (x_t), and so we briefly describe the relevant part of the existing HSMM message-passing algorithm. As derived in Murphy [78], we can define and compute the backward messages[1] B and B^* as follows:

$$\begin{aligned}
B_t(i) &\triangleq p(y_{t+1:T}|x_t = i, x_t \neq x_{t+1}) \\
&= \sum_j B^*_t(j) p(x_{t+1} = j | x_t = i), && (3.2.4) \\
B^*_t(i) &\triangleq p(y_{t+1:T}|x_{t+1} = i, x_t \neq x_{t+1}) \\
&= \sum_{d=1}^{T-t} B_{t+d}(i) \underbrace{p(D_{t+1} = d|x_{t+1} = i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|x_{t+1} = i, D_{t+1} = d)}_{\text{likelihood term}} \\
&\quad + \underbrace{p(D_{t+1} > T - t|x_{t+1} = i) p(y_{t+1:T}|x_{t+1} = i, D_{t+1} > T - t)}_{\text{censoring term}}, && (3.2.5) \\
B_T(i) &\triangleq 1, && (3.2.6)
\end{aligned}$$

where we have split the messages into B and B^* components for convenience and used $y_{k_1:k_2}$ to denote $(y_{k_1}, \ldots, y_{k_2})$. D_{t+1} represents the duration of the segment beginning at time $t + 1$. The conditioning on the parameters of the distributions, namely the observation, duration, and transition parameters, is suppressed from the notation.

We write $x_t \neq x_{t+1}$ to indicate that a new segment begins at $t+1$, and so to compute the message from $t+1$ to t we sum over all possible lengths d for the segment beginning at $t + 1$, using the backward message at $t + d$ to provide aggregate future information given a boundary just after $t + d$. The final additive term in the expression for $B^*_t(i)$ is described in Guédon [47]; it constitutes the contribution of state segments that run off the end of the provided observations, as per the censoring assumption, and depends on the survival function of the duration distribution.

Though a very similar message-passing subroutine is used in HMM Gibbs samplers, there are significant differences in computational cost between the HMM and HSMM message computations. The greater expressive power of the HSMM model necessarily increases the computational cost: the above message passing requires $\mathcal{O}(T^2N + TN^2)$

[1] In Murphy [78] and others, the symbols β and β^* are used for the messages, but to avoid confusion with our HDP parameter β, we use the symbols B and B^* for messages.

basic operations for a chain of length T and state cardinality N, while the corresponding HMM message passing algorithm requires only $\mathcal{O}(TN^2)$. However, if the support of the duration distribution is limited, or if we truncate possible segment lengths included in the inference messages to some maximum $d_{\max}$, we can instead express the asymptotic message passing cost as $\mathcal{O}(Td_{\max}N + TN^2)$. Though the increased complexity of message-passing over an HMM significantly increases the cost per iteration of sampling inference for a global model, the cost can be offset when explicit duration modeling is particularly informative, as shown in the experiments in Section 3.5. In Chapter 4, we develop methods to reduce this message-passing complexity for specific duration models.

■ 3.2.3 The HDP-HMM and Sticky HDP-HMM

The HDP-HMM [106] provides a natural Bayesian nonparametric treatment of the classical Hidden Markov Model. The model employs an HDP prior over an infinite state space, which enables both inference of state complexity and Bayesian mixing over models of varying complexity. We provide a brief overview of the HDP-HMM model and relevant inference algorithms, which we extend to develop the HDP-HSMM. A much more thorough treatment of the HDP-HMM can be found in, for example, Fox [30].

The generative process HDP-HMM(γ, α, H) given concentration parameters $\gamma, \alpha > 0$ and base measure (observation prior) H can be summarized as:

$$\beta \sim \mathrm{GEM}(\gamma), \tag{3.2.7}$$

$$\pi^{(i)} \overset{\text{iid}}{\sim} \mathrm{DP}(\alpha, \beta) \qquad \theta^{(i)} \overset{\text{iid}}{\sim} H \qquad i = 1, 2, \ldots, \tag{3.2.8}$$

$$x_t \sim \pi^{(x_{t-1})}, \tag{3.2.9}$$

$$y_t \sim f(\theta^{(x_t)}) \qquad t = 1, 2, \ldots, T, \tag{3.2.10}$$

where GEM denotes a stick breaking process [102] as in Section 2.5 and f denotes an observation distribution parameterized by draws from H. We set $x_1 = 1$. We have also suppressed explicit conditioning from the notation. See Figure 3.3 for a graphical model.

The HDP plays the role of a prior over infinite transition matrices: each $\pi^{(j)}$ is a DP draw and is interpreted as the transition distribution from state j. The $\pi^{(j)}$ are linked by being DP draws parameterized by the same discrete measure β, thus $\mathbb{E}[\pi^{(j)}] = \beta$ and the transition distributions tend to have their mass concentrated around a typical set of states, providing the desired bias towards re-entering and re-using a consistent set of states.

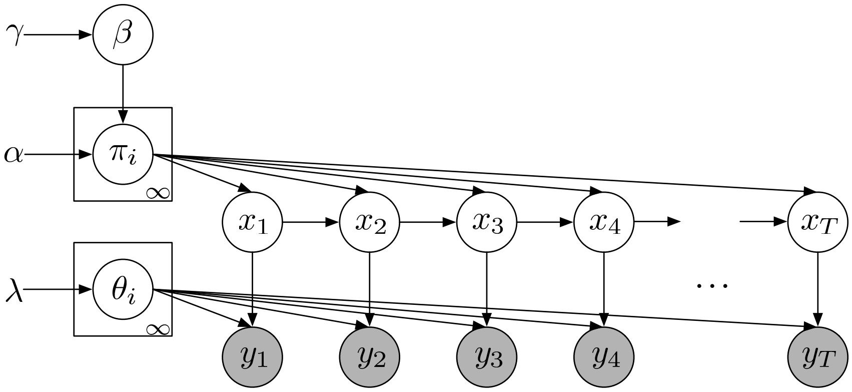


Figure 3.3: Graphical model for the HDP-HMM.

The Chinese Restaurant Franchise and direct-assignment collapsed sampling methods described in Teh et al. [106] and Fox [30] are approximate inference algorithms for the full infinite dimensional HDP, but they have a particular weakness in the sequential-data context of the HDP-HMM: each state transition must be re-sampled individually, and strong correlations within the label sequence significantly reduce mixing rates [30]. As a result, finite approximations to the HDP have been studied for the purpose of providing faster mixing. Of particular note is the weak limit approximation, used in Fox et al. [31], which has been shown to reduce mixing times for HDP-HMM inference while sacrificing little in terms of approximating the infinite-dimensional HDP posterior.

The Sticky HDP-HMM augments the HDP-HMM with an extra parameter $\kappa > 0$ that biases the process towards self-transitions and thus provides a method to encourage longer state durations. The Sticky-HDP-HMM$(\gamma, \alpha, \kappa, H)$ generative process can be written

$$\beta \sim \mathrm{GEM}(\gamma), \tag{3.2.11}$$

$$\pi^{(i)} \overset{\text{iid}}{\sim} \mathrm{DP}(\alpha + \kappa, \beta + \kappa\delta_j) \qquad \theta^{(i)} \overset{\text{iid}}{\sim} H \qquad i = 1, 2, \ldots, \tag{3.2.12}$$

$$x_t \sim \pi^{(x_{t-1})}, \tag{3.2.13}$$

$$y_t \sim f(\theta^{(x_t)}) \qquad t = 1, 2, \ldots, T, \tag{3.2.14}$$

where δ_j denotes an indicator function that takes value 1 at index j and 0 elsewhere. While the Sticky HDP-HMM allows some control over duration statistics, the state duration distributions remain geometric; a goal of this work is to provide a model in which any duration distributions specific to each state may be used.

3.3 HSMM models

In this section, we introduce the explicit-duration HSMM-based models that we use in the remainder of the chapter. We define the finite Bayesian HSMM and the HDP-HSMM and show how they can be used as components in more complex models, such as in a factorial structure. We describe generative processes that do not allow self-transitions in the super-state sequence, but we emphasize that we can also allow self-transitions and still employ the inference algorithms we describe; in fact, allowing self-transitions simplifies inference in the HDP-HSMM, since complications arise as a result of the hierarchical prior and an elimination of self-transitions. However, there is a clear modeling gain by eliminating self-transitions: when self-transitions are allowed, the "explicit duration distributions" do not model the state duration statistics directly. To allow direct modeling of state durations, we must consider the case where self-transitions do not occurr.

3.3.1 Finite Bayesian HSMM

The finite Bayesian HSMM is a combination of the Bayesian HMM approach with semi-Markov state durations and is the model we generalize to the HDP-HSMM. It is instructive to compare this construction with that of the finite model used in the weak-limit HDP-HSMM sampler that will be described in Section 3.4.2, since in that case the hierarchical ties between rows of the transition matrix requires particular care.

The generative process for a Bayesian HSMM with N states and observation and duration parameter prior distributions of H and G, respectively, can be summarized as

$$\pi^{(i)} \overset{\text{iid}}{\sim} \mathrm{Dir}(\alpha(1-\delta_i)) \qquad (\theta^{(i)}, \vartheta^{(i)}) \overset{\text{iid}}{\sim} H \times G \qquad i = 1, 2, \ldots, N, \tag{3.3.1}$$

$$z_s \sim \pi^{(z_{s-1})}, \tag{3.3.2}$$

$$d_s \sim g(\vartheta^{(z_s)}), \qquad s = 1, 2, \ldots, \tag{3.3.3}$$

$$x_{t(s):t(s+1)-1} = z_s, \tag{3.3.4}$$

$$y_{t(s):t(s+1)-1} \overset{\text{iid}}{\sim} f(\theta^{(z_s)}) \tag{3.3.5}$$

where f and g denote observation and duration distributions parameterized by draws from H and G, respectively, and $t(s)$ is defined in (3.2.3). As in Section 3.2.2, we collect the $\pi^{(i)}$ for $i = 1, 2, \ldots, N$ into the rows of a transition matrix A. We use $\mathrm{Dir}(\alpha(1-\delta_i))$ to denote a symmetric Dirichlet distribution with parameter α except with the ith component of the hyperparameter vector set to zero, hence fixing $A_{ii} = 0$ and ensuring there will be no self-transitions sampled in the super-state sequence (z_s).

Note, crucially, that in this definition the $\pi^{(i)}$ are not tied across various i. In the

HDP-HSMM, as well as the weak limit model used for approximate inference in the HDP-HSMM, the $\pi^{(i)}$ will be tied through the hierarchical prior (specifically via β), and that connection is necessary to penalize the total number of states and encourage a small, consistent set of states to be visited in the super-state sequence. However, the interaction between the hierarchical prior and the elimination of self-transitions presents an inference challenge.

■ 3.3.2 HDP-HSMM

The generative process of the HDP-HSMM is similar to that of the HDP-HMM, with some extra work to include duration distributions. The process HDP-HSMM(γ, α, H, G), illustrated in Figure 3.4, can be written as

$$\beta \sim \mathrm{GEM}(\gamma), \tag{3.3.6}$$

$$\pi^{(i)} \overset{\text{iid}}{\sim} \mathrm{DP}(\alpha, \beta) \qquad (\theta^{(i)}, \vartheta^{(i)}) \overset{\text{iid}}{\sim} H \times G \qquad i = 1, 2, \ldots, \tag{3.3.7}$$

$$z_s \sim \bar{\pi}^{(z_{s-1})}, \tag{3.3.8}$$

$$d_s \sim g(\vartheta^{(z_s)}) \qquad s = 1, 2, \ldots, \tag{3.3.9}$$

$$x_{t(s):t(s+1)-1} = z_s, \tag{3.3.10}$$

$$y_{t(s):t(s+1)-1} \overset{\text{iid}}{\sim} f(\theta^{(z_s)}) \tag{3.3.11}$$

where we use $\bar{\pi}^{(i)} \triangleq \frac{\pi_j^{(i)}}{1-\pi_i^{(i)}}(1 - \delta_{ij})$ to eliminate self-transitions in the super-state sequence (z_s).

Note that the atoms we edit to eliminate self-transitions are the same atoms that are affected by the global sticky bias in the Sticky HDP-HMM.

■ 3.3.3 Factorial structure

We can easily compose our sequential models into other common model structures, such as the factorial structure of the factorial HMM [40]. Factorial models are very useful for source separation problems, and when combined with the rich class of sequential models provided by the HSMM, one can use prior duration information about each source to greatly improve performance (as demonstrated in Section 3.5). Here, we briefly outline the factorial model and its uses.

If we use $y \sim$ HDP-HSMM(α, γ, H, G) to denote an observation sequence generated by the process defined in (3.3.6)-(3.3.11), then the generative process for a factorial HDP-HSMM with K component sequences can be written as

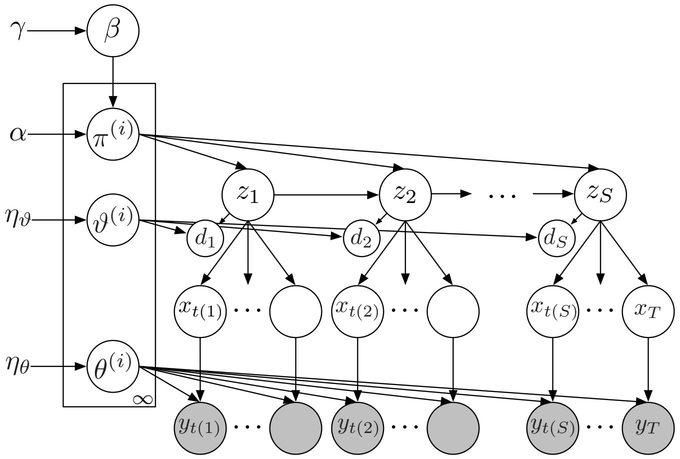


Figure 3.4: A graphical model for the HDP-HSMM in which the number of segments S, and hence the number of nodes, is random.

$$y^{(k)} \sim \text{HDP-HSMM}(\alpha_k, \gamma_k, H_k, G_k) \qquad k = 1, 2, \ldots, K, \tag{3.3.12}$$

$$\bar{y}_t \triangleq \sum_{k=1}^{K} y_t^{(k)} + w_t \qquad t = 1, 2, \ldots, T, \tag{3.3.13}$$

where w_t is a noise process independent of the other components of the model states.

A graphical model for a factorial HMM can be seen in Figure 3.5, and a factorial HSMM or factorial HDP-HSMM simply replaces the hidden state chains with semi-Markov chains. Each chain, indexed by superscripts, evolves with independent dynamics and produces independent emissions, but the observations are combinations of the independent emissions. Note that each component HSMM is not restricted to any fixed number of states.

Such factorial models are natural ways to frame source separation or disaggregation problems, which require identifying component emissions and component states. With the Bayesian framework, we also model uncertainty and ambiguity in such a separation. In Section 3.5.2 we demonstrate the use of a factorial HDP-HSMM for the task of disaggregating home power signals.

Problems in source separation or disaggregation are often ill-conditioned, and so one relies on prior information in addition to the source independence structure to solve the separation problem. Furthermore, representation of uncertainty is often important, since there may be several good explanations for the data. These considerations motivate Bayesian inference as well as direct modeling of state duration statistics.

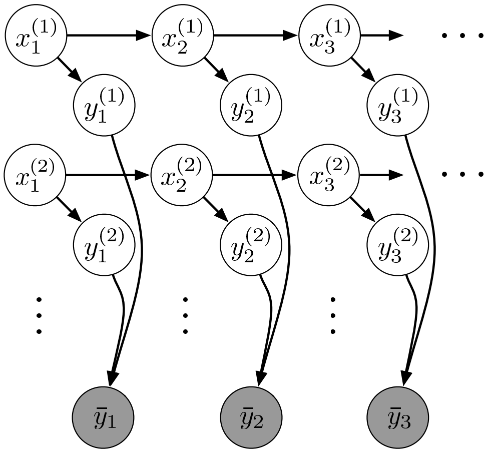


Figure 3.5: A graphical model for the factorial HMM, which can naturally be extended to factorial structures involving the HSMM or HDP-HSMM.

3.4 Inference Algorithms

We describe two Gibbs sampling inference algorithms, beginning with a sampling algorithm for the finite Bayesian HSMM, which is built upon in developing algorithms for the HDP-HSMM in the sequel. Next, we develop a weak-limit Gibbs sampling algorithm for the HDP-HSMM, which parallels the popular weak-limit sampler for the HDP-HMM and its sticky extension. Finally, we introduce a collapsed sampler which parallels the direct assignment sampler of Teh et al. [106]. For both of the HDP-HSMM samplers there is a loss of conjugacy with the HDP prior due to the fact that self-transitions in the super-state sequence are not permitted (see Section 3.4.2). We develop auxiliary variables to form an augmented representation that effectively recovers conjugacy and hence enables fast Gibbs steps.

In comparing the weak limit and direct assignment sampler, the most important trade-offs are that the direct assignment sampler works with the infinite model by integrating out the transition matrix A while simplifying bookkeeping by maintaining part of β, as we make clear in Section 3.4.3; it also collapses the observation and duration parameters. However, the variables in the label sequence (x_t) are coupled by the integration, and hence each element of the label sequence must be resampled sequentially. In contrast, the weak limit sampler represents all latent components of the model (up to an adjustable finite approximation for the HDP) and thus allows block resampling of the label sequence by exploiting HSMM message passing.

We end the section with a discussion of leveraging changepoint side-information to greatly accelerate inference.

■ 3.4.1 A Gibbs sampler for the finite Bayesian HSMM

In this section, we describe a blocked Gibbs sampler for the finite HSMM.

Outline of Gibbs sampler

To perform posterior inference in a finite Bayesian HSMM, we construct a Gibbs sampler resembling that for finite HMMs. Our goal is to construct samples from the posterior

$$p((x_t), \{\theta^{(i)}\}, \{\pi^{(i)}\}, \{\vartheta^{(i)}\} | (y_t), H, G, \alpha) \tag{3.4.1}$$

by drawing samples from the distribution, where G represents the prior over duration parameters. We can construct these samples by following a Gibbs sampling algorithm in which we iteratively sample from the appropriate conditional distributions of (x_t), $\{\pi^{(i)}\}$, $\{\vartheta^{(i)}\}$, and $\{\theta^{(i)}\}$.

Sampling $\{\theta^{(i)}\}$ or $\{\vartheta^{(i)}\}$ from their respective conditional distributions can be easily reduced to standard problems depending on the particular priors chosen. Sampling the transition matrix rows $\{\pi^{(i)}\}$ is straightforward if the prior on each row is Dirichlet over the off-diagonal entries and so we do not discuss it in this section, but we note that when the rows are tied together hierarchically (as in the weak-limit approximation to the HDP-HSMM), resampling the $\{\pi^{(i)}\}$ correctly requires particular care (see Section 3.4.2).

In the following section we develop the algorithm for block-sampling the label sequence (x_t) from its conditional distribution by employing the HSMM message-passing scheme.

Blocked conditional sampling of (x_t) with message passing

To block sample $(x_t)|\{\theta^{(i)}\}, \{\pi^{(i)}\}, \{\vartheta^{(i)}\}, (y_t)$ in an HSMM we can extend the standard block state sampling scheme for an HMM. The key challenge is that to block sample the states in an HSMM we must also be able to sample the posterior duration variables.

If we compute the backward messages B and B^* described in Section 3.2.2, then we can easily draw a posterior sample for the first state according to

$$\begin{aligned} p(x_1 = k | y_{1:T}) &\propto p(x_1 = k) p(y_{1:T} | x_1 = k) && (3.4.2) \\ &= p(x_1 = k) B_0^*(k), && (3.4.3) \end{aligned}$$

where we have used the assumption that the observation sequence begins on a segment boundary and suppressed notation for conditioning on parameters.

We can also use the messages to efficiently draw a sample from the posterior duration distribution for the sampled initial state. Conditioning on the initial state draw, $\bar{x}_1$, the posterior duration of the first state is:

$$p(D_1 = d|y_{1:T}, x_1 = \bar{x}_1) = \frac{p(D_1 = d, y_{1:T}|x_1 = \bar{x}_1)}{p(y_{1:T}|x_1 = \bar{x}_1)} \tag{3.4.4}$$

$$= \frac{p(D_1 = d|x_1 = \bar{x}_1)p(y_{1:d}|D_1 = d, x_1 = \bar{x}_1)p(y_{d+1:T}|D_1 = d, x_1 = \bar{x}_1)}{p(y_{1:T}|x_1 = \bar{x}_1)} \tag{3.4.5}$$

$$= \frac{p(D_1 = d)p(y_{1:d}|D_1 = d, x_1 = \bar{x}_1)B_d(\bar{x}_1)}{B_0^*(\bar{x}_1)}. \tag{3.4.6}$$

We repeat the process by using x_{D_1+1} as our new initial state with initial distribution $p(x_{D_1+1} = i|x_1 = \bar{x}_1)$, and thus draw a block sample for the entire label sequence.

In each step of the forward-sampling process a label is assigned in the label sequence. To compute each label assignment, Eq. (3.4.6) is evaluated in constant time and, when the duration expires, Eq. (3.4.3) is sampled in time proportional to the number of states N. Therefore the overall complexity of the forward-sampling process for an HSMM with N states and sequence length T is $\mathcal{O}(TN)$ after computing the HSMM messages.

■ 3.4.2 A weak-limit Gibbs sampler for the HDP-HSMM

The weak-limit sampler for an HDP-HMM [31] constructs a finite approximation to the HDP transitions prior with finite L-dimensional Dirichlet distributions, motivated by the fact that the infinite limit of such a construction converges in distribution (i.e. weakly) to an HDP:

$$\beta|\gamma \sim \text{Dir}(\gamma/L, \dots, \gamma/L), \tag{3.4.7}$$

$$\pi^{(i)}|\alpha, \beta \sim \text{Dir}(\alpha\beta_1, \dots, \alpha\beta_L) \qquad i = 1, \dots, L, \tag{3.4.8}$$

where we again interpret $\pi^{(i)}$ as the transition distribution for state i and β as the distribution which ties state transition distributions together and encourages shared sparsity. Practically, the weak limit approximation enables the complete representation of the transition matrix in a finite form, and thus, when we also represent all parameters, allows block sampling of the entire label sequence at once. The parameter L gives us control over the approximation quality, with the guarantee that the approximation will become exact as L grows; see Ishwaran and Zarepour [56], especially Theorem 1, for a discussion of theoretical guarantees.

We can employ the weak limit approximation to create a finite HSMM that approximates inference in the HDP-HSMM. This approximation technique often results in greatly accelerated mixing, and hence it is the technique we employ for the experiments in the sequel. However, the inference algorithm of Section 3.4.1 must be modified to incorporate the fact that the $\{\pi^{(i)}\}$ are no longer mutually independent and are instead tied through the shared β. This dependence between the transition rows introduces potential conjugacy issues with the hierarchical Dirichlet prior; the following section explains the difficulty as well as a simple algorithmic solution via auxiliary variables.

The beam sampling technique [110] can be applied here with little modification, as in Dewar et al. [23], to sample over the approximation parameter L, thus avoiding the need to set L a priori while still allowing instantiation of the transition matrix and block sampling of the state sequence. This technique is especially useful if the number of states could be very large and is difficult to bound a priori. We do not explore beam sampling here.

Conditional sampling of $\{\pi^{(i)}\}$ with data augmentation

To construct our overall Gibbs sampler, we need to be able to resample the transition matrix π given the other components of the model. However, by ruling out self-transitions while maintaining a hierarchical link between the transition rows, the model is no longer fully conjugate, and hence resampling is not necessarily easy. To observe the loss of conjugacy using the hierarchical prior required in the weak-limit approximation, note that we can summarize the relevant portion of the generative model as

$$\beta|\gamma \sim \text{Dir}(\gamma, \ldots, \gamma), \tag{3.4.9}$$

$$\pi^{(j)}|\beta \sim \text{Dir}(\alpha\beta_1, \ldots, \alpha\beta_L) \qquad j = 1, \ldots, L, \tag{3.4.10}$$

$$x_t|\{\pi^{(j)}\}, x_{t-1} \sim \bar{\pi}_{x_{t-1}} \qquad t = 2, \ldots, T, \tag{3.4.11}$$

$$\tag{3.4.12}$$

where $\bar{\pi}_j$ represents $\pi^{(j)}$ with the jth component removed and renormalized appropriately:

$$\bar{\pi}_{ji} = \frac{\pi_i^{(j)}(1-\delta_{ij})}{1-\pi_i^{(j)}} \quad \text{where} \quad \delta_{ij} \triangleq \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}. \tag{3.4.13}$$

The deterministic transformation from $\pi^{(j)}$ to $\bar{\pi}^{(j)}$ eliminates self-transitions. Note that we have suppressed the observation parameter set, duration parameter set, and observation sequence for simplicity.

Consider the distribution of $\pi^{(1)}|(x_t), \beta$, the first row of the transition matrix:

$$
\begin{aligned}
p(\pi^{(1)}|(x_t),\beta) &\propto p(\pi^{(1)}|\beta)p((x_t)|\pi^{(1)}) && (3.4.14)\\
&\propto (\pi_1^{(1)})^{\alpha\beta_1-1}\cdots(\pi_L^{(1)})^{\alpha\beta_L-1}\left(\frac{\pi_2^{(1)}}{1-\pi_1^{(1)}}\right)^{n_{12}}\cdots\left(\frac{\pi_L^{(1)}}{1-\pi_1^{(1)}}\right)^{n_{1L}}, && (3.4.15)
\end{aligned}
$$

where n_{ij} are the number of transitions from state i to state j in the super-state sequence (z_s):

$$n_{ij} \triangleq \#\{s \in [S-1] : z_s = i, z_{s+1} = j\}. \tag{3.4.16}$$

Essentially, because of the extra $\frac{1}{1-\pi_1^{(1)}}$ terms from the likelihood without self-transitions, we cannot reduce this expression to the Dirichlet form over the components of $\pi^{(1)}$, and therefore we cannot proceed with sampling m and resampling β and π as in Teh et al. [106].

However, we can introduce auxiliary variables to recover conjugacy [109], also called a completion construction [94, Section 10.1.2]. We define an extended generative model with extra random variables that, when the extra variables are marginalized out, corresponds to the original model. Then we show that conditional distributions simplify with the extra variables, hence allowing us to cycle simple Gibbs updates to produce an efficient sampler.

For simplicity, we focus on the first row of the transition matrix, namely $\pi^{(1)}$, and the draws that depend on it; the reasoning immediately extends to the other rows. We also drop the parameter α for convenience, and to simplify notation, we write n_i for n_{1i}, for $i = 1, 2, \ldots, N$. We also define $n_{.} = \sum_i n_i$. First, we write the relevant portion of the generative process as

$$
\begin{aligned}
\pi^{(1)}|\beta &\sim \mathrm{Dir}(\beta), && (3.4.17)\\
z_i|\bar{\pi}^{(1)} &\sim \bar{\pi}_1 && (3.4.18)\\
y_i|z_i &\sim f(z_i) \qquad\qquad i = 1, \ldots, n_{.}. && (3.4.19)
\end{aligned}
$$

Here, sampling $z_i = k$ represents a transition from state 1 to state k. The $\{y_i\}$ represent the observations on which we condition; in particular, if we have $z_i = k$ then y_i corresponds to an emission from state k in the HSMM. See the graphical model in Figure 3.6(a) for a depiction of the relationship between the variables.

We can introduce auxiliary variables $\{\rho_i\}_{i=1}^{n_{.}}$, where each ρ_i is independently drawn

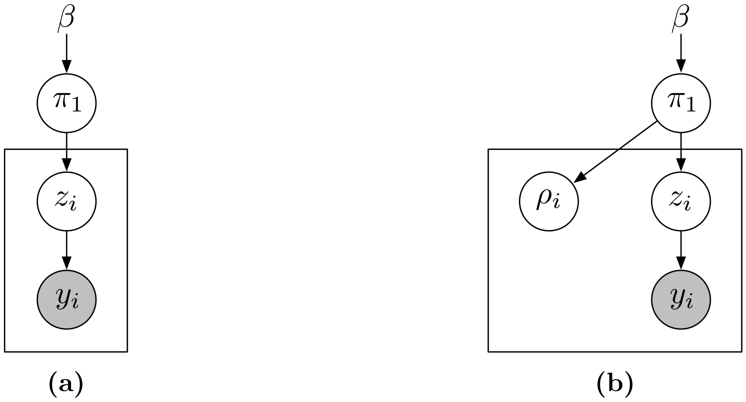


Figure 3.6: Simplified depiction of the relationship between the auxiliary variables and the rest of the model; 3.6(a) depicts the nonconjugate setting and 3.6(b) shows the introduced auxiliary variables $\{\rho_i\}$.

from a geometric distribution supported on $\{0, 1, \ldots\}$ with success parameter $1 - \pi_1^{(1)}$. That is, we introduce $\rho_i | \pi_1^{(1)} \sim \text{Geo}(1 - \pi_1^{(1)})$, shown in Figure 3.6(b). Thus our posterior becomes:

$$
\begin{aligned}
&p(\pi^{(1)} | \{z_i\}, \{\rho_i\}) && (3.4.20)\\
&\quad \propto p(\pi^{(1)}) p(\{z_i\} | \pi^{(1)}) p(\{\rho_i\} | \{\pi_i^{(1)}\}) && (3.4.21)\\
&\quad \propto (\pi_1^{(1)})^{\beta_1 - 1} \cdots (\pi_L^{(1)})^{\beta_L - 1} \left(\frac{\pi_2^{(1)}}{1 - \pi_1^{(1)}} \right)^{n_2} \cdots \left(\frac{\pi_L^{(1)}}{1 - \pi_1^{(1)}} \right)^{n_L} \left(\prod_{i=1}^{n.} (\pi_1^{(1)})^{\rho_i} (1 - \pi_1^{(1)}) \right) && (3.4.22)\\
&\quad = (\pi_1^{(1)})^{\beta_1 + \sum_i \rho_i - 1} (\pi_2^{(1)})^{\beta_2 + n_2 - 1} \cdots (\pi_L^{(1)})^{\beta_L + n_L - 1} && (3.4.23)\\
&\quad \propto \text{Dir}\left(\beta_1 + \sum_i \rho_i, \beta_2 + n_2, \ldots, \beta_L + n_L \right). && (3.4.24)
\end{aligned}
$$

Noting that $n. = \sum_i n_i$, we recover conjugacy and hence can iterate simple Gibbs steps.

We can compare the numerical performance of the auxiliary variable sampler to a Metropolis-Hastings sampler in the model without auxiliary variables. Figure 3.7 shows the sample chain autocorrelations for the first component of π in both samplers. Figure 3.8 compares the Multivariate Scale Reduction Factors of Brooks and Gelman [16] for the two samplers, where good mixing is indicated by achieving the statistic's asymptotic value of unity. For a detailed evaluation, see Appendix B.

We can easily extend the data augmentation to the full HSMM, and once we have augmented the data with the auxiliary variables $\{\rho_s\}_{s=1}^{S}$ we are once again in the

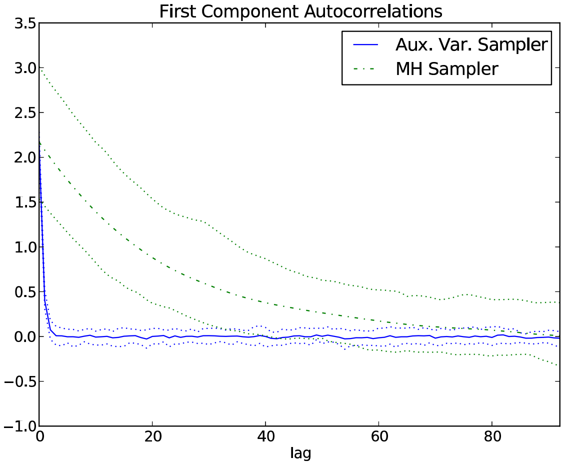


Figure 3.7: Empirical sample chain autocorrelation for the first component of π for both the proposed auxiliary variable sampler and a Metropolis-Hastings sampler. The figure shows mean autocorrelations over 50 randomly-initialized runs for each sampler, with the corresponding dotted lines showing the 10th and 90th percentile autocorrelation chains over those runs. The rapidly diminishing autocorrelation for the auxiliary variable sampler is indicative of fast mixing.

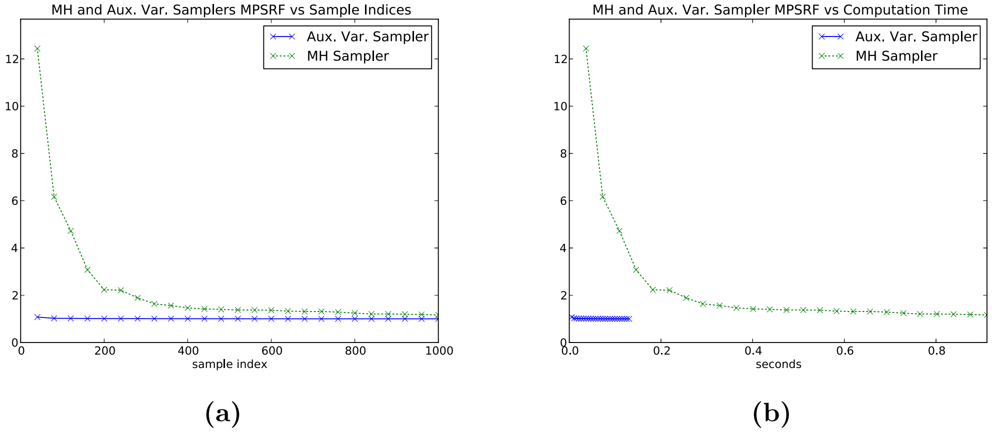


Figure 3.8: Multivariate Potential Scale Reduction Factors for both the proposed auxiliary variable sampler and a Metropolis-Hastings sampler. The auxiliary variable sampler rapidly achieves the statistic's asymptotic value of unity. Note that the auxiliary variable sampler is also much more efficient to execute, as shown in 3.8(b).

conjugate setting. A graphical model for the weak-limit approximation to the HDP-HSMM including the auxiliary variables is shown in Figure 3.9.

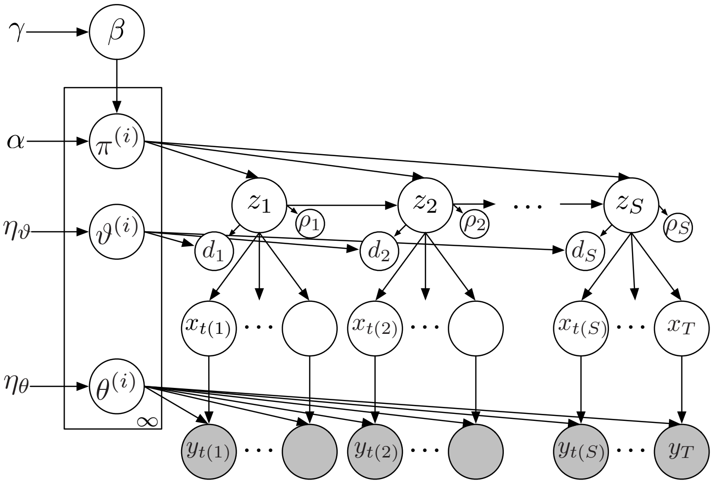


Figure 3.9: Graphical model for the weak-limit approximation including auxiliary variables.

■ 3.4.3 A direct assignment sampler for the HDP-HSMM

Though the main experiments in this chapter are performed with the weak-limit sampler developed in the previous section, we provide a direct assignment (DA) sampler as well for theoretical completeness and because it may be useful in cases where there is insufficient data to inform some latent parameters so that marginalization is necessary for mixing or estimating marginal likelihoods. In the direct assignment sampler for the HDP-HMM the infinite transition matrix π is analytically marginalized out along with the observation parameters. The sampler represents explicit instantiations of the label sequence (x_t) and the prefix of the infinite vector $\beta_{1:K}$ where $K = \#\{x_t : t = 1, \ldots, T\}$. There are also auxiliary variables m used to resample β, but for simplicity we do not discuss them here; see Teh et al. [106] for details.

Our DA sampler additionally represents the auxiliary variables necessary to recover HDP conjugacy (as introduced in the previous section). Note that the requirement for, and correctness of, the auxiliary variables described in the finite setting in Section 3.4.2 immediately extends to the infinite setting as a consequence of the Dirichlet Process's definition in terms of the finite Dirichlet distribution and the Kolmogorov extension theorem [19, Chapter 4]; for a detailed discussion, see Orbanz [87].

Resampling (x_t)

To resample each element of the label sequence (x_t), we perform an update similar to that of the HDP-HMM direct assignment sampler. As described in Fox [30], the corresponding HDP-HMM sampling step for each element x_t of the label sequence is to sample a new label k with probability proportional (over k) to

$$p(x_t = k|(x_{\setminus t}), \beta) \propto \frac{\alpha\beta_k + n_{x_{t-1},k}}{\alpha + n_{x_{t-1},\cdot}} \cdot \frac{\alpha\beta_{x_{t+1}} + n_{k,x_{t+1}} + \mathbb{I}[x_{t-1} = k = x_{t+1}]}{\alpha + n_{k,\cdot} + \mathbb{I}[x_{t-1} = k]} \cdot f_{\text{obs}}(y_t|x_t = k) \tag{3.4.25}$$

for $k = 1, \ldots, K+1$ where $K = \#\{x_t : t = 1, \ldots, T\}$ and where $\mathbb{I}$ is an indicator function taking value 1 if its argument condition is true and 0 otherwise.[2] The variables n_{ij} are transition counts in the label sequence excluding the transition into and out of x_t; that is, $n_{ij} = \#\{x_\tau = i, x_{\tau+1} = j : \tau \in \{1, \ldots, T-1\} \setminus \{t-1, t\}\}$. The function f_{obs} is a predictive likelihood:

$$f_{\text{obs}}(y_t|k) \triangleq p(y_t|x_t = k, \{y_\tau : x_\tau = k\}, H) \tag{3.4.26}$$

$$= \int_{\theta^{(k)}} \underbrace{p(y_t|x_t = k, \theta^{(k)})}_{\text{likelihood}} \underbrace{\prod_{\tau : x_\tau = k} p(y_\tau|x_\tau = k, \theta^{(k)})}_{\text{likelihood of data with same label}} \underbrace{p(\theta^{(k)}|H)}_{\text{observation parameter prior}} d\theta^{(k)} \tag{3.4.27}$$

We can derive this step by writing the complete joint probability $p((x_t), (y_t)|\beta, H)$ leveraging exchangeability; this joint probability value is proportional to the desired posterior probability $p(x_t|(x_{\setminus t}), (y_t), \beta, H)$. When we consider each possible assignment $x_t = k$, we can cancel all the terms that do not depend on k, namely all the transition probabilities other than those to and from x_t and all data likelihoods other than that for y_t. However, this cancellation process relies on the fact that for the HDP-HMM there is no distinction between self-transitions and new transitions: the term for each t in the complete posterior simply involves transition scores no matter the labels of x_{t+1} and x_{t-1}. In the HDP-HSMM case, we must consider segments and their durations separately from transitions.

To derive an expression for resampling x_t in the case of the HDP-HSMM, we can similarly consider writing out an expression for the joint probability $p((x_t), (y_t)|\beta, H, G)$. However, note that as the label assignment of $x_t = k$ varies, the terms in the expression change according to whether $x_{t-1} = k$ or $x_{t+1} = k$. That is, if $x_{t-1} = k$ or $x_{t+1} = k$, the probability expression includes a segment term for entire contiguous run of label k. Hence, since we can only cancel terms that do not depend on k, our score expression must include terms for the adjacent segments into which x_t may merge. See Figure 3.10 for an illustration.

[2] The indicator variables are present because the two transition probabilities are not independent but rather exchangeable.

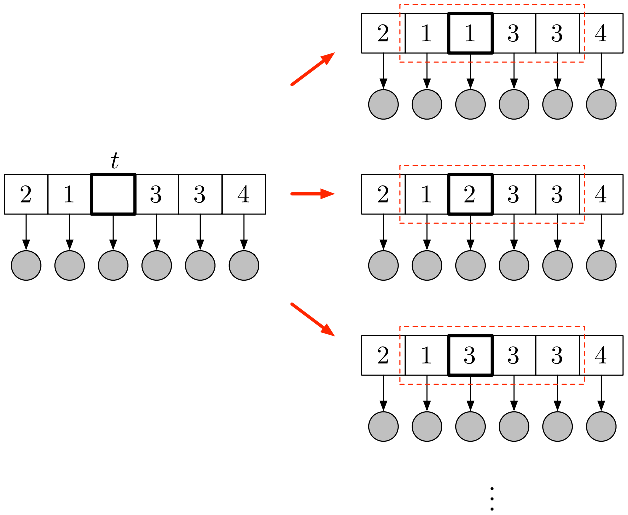


Figure 3.10: Illustration of the Gibbs step to resample x_t for the DA sampler for the HDP-HSMM. The red dashed boxes indicate the elements of the label sequence that contribute to the score computation for $k = 1, 2, 3$ which produce two, three, and two segment terms, respectively. The label sequence element being resample is emphasized in bold.

The final expression for the probability of sampling the new value of x_t to be k then consists of between 1 and 3 segment score terms, depending on merges with adjacent segments, each of which has the form

$$p(x_t = k | (x_{\setminus t}), \beta, H, G) \propto \underbrace{\frac{\alpha\beta_k + n_{x_{\text{prev}},k}}{\alpha(1-\beta_{x_{\text{prev}}}) + n_{x_{\text{prev}},\cdot}}}_{\text{left-transition}} \cdot \underbrace{\frac{\alpha\beta_{x_{\text{next}}} + n_{k,x_{\text{next}}}}{\alpha(1-\beta_k) + n_{k,\cdot}}}_{\text{right-transition}}$$
$$\cdot \underbrace{f_{\text{dur}}(t_2 - t_1 + 1)}_{\text{duration}} \cdot \underbrace{f_{\text{obs}}(y_{t_1:t_2}|k)}_{\text{observation}}, \qquad (3.4.28)$$

where we have used t_1 and t_2 to denote the first and last indices of the segment, respectively.

The function $f_{\text{dur}}(d|k)$ is the corresponding duration predictive likelihood evaluated on a duration d, which depends on the durations of other segments with label k and any duration hyperparameters. The function f_{obs} now represents a *block* or *joint* predictive likelihood over all the data in a segment (see, for example, Murphy [79] for a thorough discussion of the Gaussian case). Note that the denominators in the transition terms are affected by the elimination of self-transitions by a rescaling of the "total mass." The resulting chain is ergodic if the duration predictive score f_{dur} has a support that includes $\{1, 2, \ldots, d_{\max}\}$, so that segments can be split and merged in any combination.

Resampling β and auxiliary variables ρ

To allow conjugate resampling of β, auxiliary variables must be introduced to deal with the conjugacy issue raised in Section 3.4.2. In the direct assignment samplers, the auxiliary variables are not used to resample diagonal entries of the transition matrix π, which is marginalized out, but rather to resample β. In particular, with each segment s we associate an auxiliary count ρ_s which is independent of the data and only serves to preserve conjugacy in the HDP:

$$\pi_{kk}|\alpha,\beta \sim \text{Beta}(\alpha\beta_k, \alpha(1-\beta_k)) \tag{3.4.29}$$

$$\rho_s|\pi_{z_s,z_s} \sim \text{Geo}(1-\pi_{z_s,z_s}) \quad s=1,2,\ldots,S \tag{3.4.30}$$

for $k=1,2,\ldots,K$ where K is the number of used states. The count $n_{i,i}$, which is used in resampling the auxiliary variables m of Teh et al. [106] which in turn are then used to resample β, is defined to be the total of the auxiliary variables for other segments with the same label:

$$n_{i,j} = \begin{cases} \#\{s\ :\ z_s=i, z_{s+1}=j, s=1,2,\ldots,S-1\} & i\neq j \\ \sum_{\{s\,:\,z_s=i\}} \rho_s & i=j \end{cases} \tag{3.4.31}$$

for $i,j=1,2,\ldots,K$ where K. This construction can be interpreted as simply sampling the number of self-transitions we may have seen at segment s given β and the counts of self- and non-self transitions in the super-state sequence. Note the diagonal entries $\pi_i^{(i)}$ are independent of the data given (z_s); as before, this auxiliary variable procedure is a convenient way to integrate out numerically the diagonal entries of the transition matrix.

By using the total auxiliary counts as the statistics for $n_{i,i}$, we can resample $\beta|(x_t),\alpha,\gamma$ according to the procedure for the HDP-HMM as described in Teh et al. [106].

■ 3.4.4 Exploiting changepoint side-information

In many circumstances, we may not need to consider all time indices as possible changepoints at which the super-state may switch; it may be easy to rule out many non-changepoints from consideration. For example, in the power disaggregation application in Section 3.5, we can run inexpensive changepoint detection on the observations to get a list of *possible* changepoints, ruling out many obvious non-changepoints. The possible changepoints divide the label sequence into state *blocks*, where within each block the label sequence must be constant, though sequential blocks may have the same label. By only allowing super-state switching to occur at these detected changepoints, we can greatly reduce the computation of all the samplers considered.

In the case of the weak-limit sampler, the complexity of the bottleneck message-passing step is reduced to a function of the number of possible changepoints (instead of total sequence length): the asymptotic complexity becomes $\mathcal{O}(T_{\text{change}}^2 N + N^2 T_{\text{change}})$, where T_{change}, the number of possible changepoints, may be dramatically smaller than the sequence length T. We simply modify the backward message-passing procedure to sum only over the possible durations:

$$\begin{aligned} B_t^*(i) &\triangleq p(y_{t+1:T}|x_{t+1} = i, x_t \neq x_{t+1}) \\ &= \sum_{d \in \mathbb{D}} B_{t+d}(i) \underbrace{\tilde{p}(D_{t+1} = d|x_{t+1} = i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|x_{t+1} = i, D_{t+1} = d)}_{\text{likelihood term}} \\ &\quad + \underbrace{\tilde{p}(D_{t+1} > T - t|x_{t+1} = i) p(y_{t+1:T}|x_{t+1} = i, D_{t+1} > T - t)}_{\text{censoring term}}, \end{aligned} \tag{3.4.32}$$

where $\tilde{p}$ represents the duration distribution restricted to the set of possible durations $\mathbb{D} \subset \mathbb{N}_+$ and renormalized. We similarly modify the forward-sampling procedure to only consider possible durations. It is also clear how to adapt the DA sampler: instead of re-sampling each element of the label sequence (x_t) we simply consider the block label sequence, resampling each block's label (allowing merging with adjacent blocks).

■ 3.5 Experiments

In this section, we evaluate the proposed HDP-HSMM sampling algorithms on both synthetic and real data. First, we compare the HDP-HSMM direct assignment sampler to the weak limit sampler as well as the Sticky HDP-HMM direct assignment sampler, showing that the HDP-HSMM direct assignment sampler has similar performance to that for the Sticky HDP-HMM and that the weak limit sampler is much faster. Next, we evaluate the HDP-HSMM weak limit sampler on synthetic data generated from finite HSMMs and HMMs. We show that the HDP-HSMM applied to HSMM data can efficiently learn the correct model, including the correct number of states and state labels, while the HDP-HMM is unable to capture non-geometric duration statistics. We also apply the HDP-HSMM to data generated by an HMM and demonstrate that, when equipped with a duration distribution class that includes geometric durations, the HDP-HSMM can also efficiently learn an HMM model when appropriate with little loss in efficiency. Next, we use the HDP-HSMM in a factorial [40] structure for the purpose of disaggregating a whole-home power signal into the power draws of individual devices. We show that encoding simple duration prior information when modeling individual devices can greatly improve performance, and further that a Bayesian treatment of the parameters is advantageous. We also demonstrate how changepoint side-information

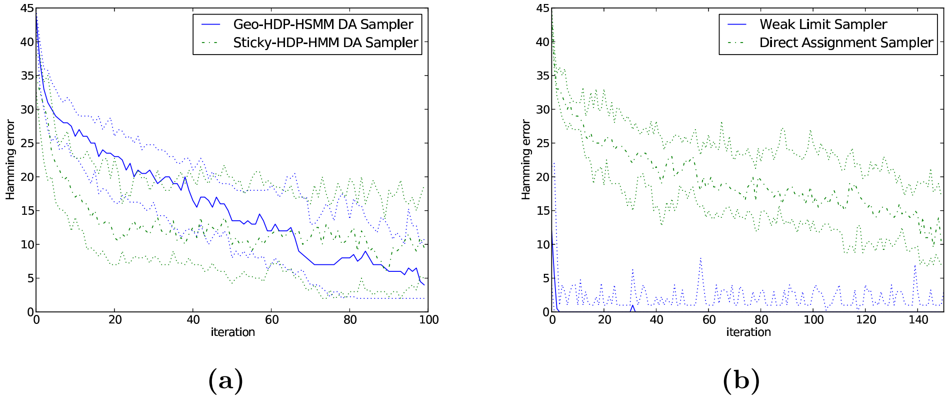


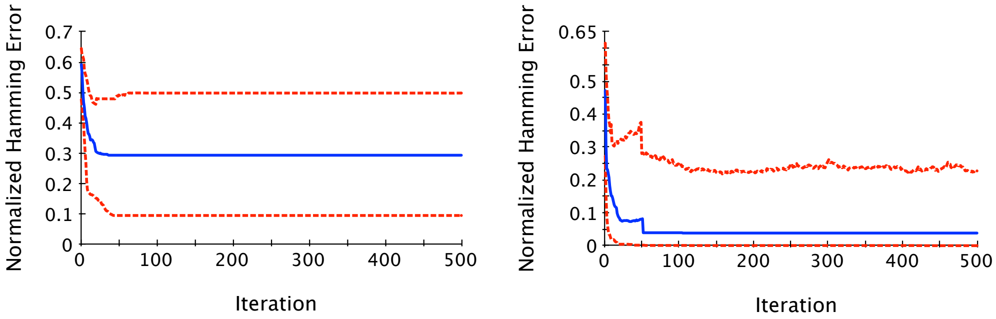
Figure 3.11: (a) compares the HDP-HSMM with geometric durations direct assignment sampler with that of the Sticky HDP-HMM, both applied to HMM data. The sticky parameter κ was chosen to maximize mixing. (b) compares the HDP-HSMM direct assignment sampler with the weak limit sampler. In all plots, solid lines are the median error at each time over 25 independent chains; dashed lines are 25th and 75th percentile errors.

can be leveraged to significantly speed up computation.

■ 3.5.1 Synthetic data

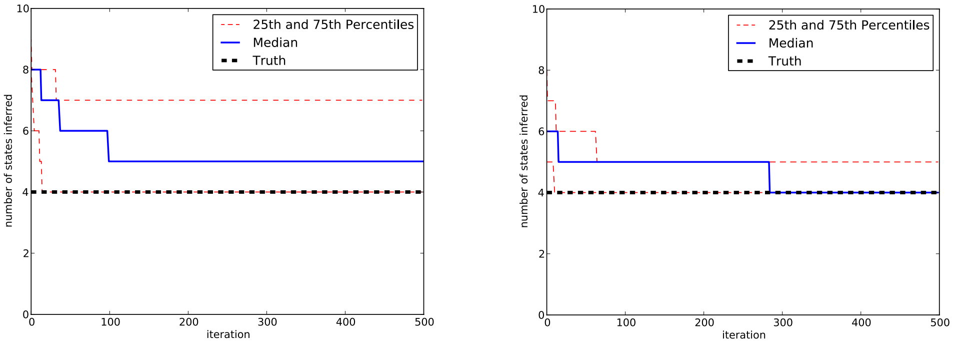
Figure 3.11 compares the HDP-HSMM direct assignment sampler to that of the Sticky HDP-HMM as well as the HDP-HSMM weak limit sampler. Figure 3.11(a) shows that the direct assignment sampler for an HDP-HSMM with geometric durations performs similarly to the Sticky HDP-HSMM direct assignment sampler when applied to data generated by an HMM with scalar Gaussian emissions. Figures 3.11(b) shows that the weak limit sampler mixes much more quickly than the direct assignment sampler. Each iteration of the weak limit sampler is also much faster to execute (approximately 50x faster in our implementations in Python). Due to its much greater efficiency, we focus on the weak limit sampler for the rest of this section; we believe it is a superior inference algorithm whenever an adequately large approximation parameter L can be chosen a priori.

Figure 3.12 summarizes the results of applying both a HDP-HSMM with Poisson durations and an HDP-HMM to data generated from an HSMM with four states, Poisson durations, and 2-dimensional mixture-of-Gaussian emissions. In the 25 Gibbs sampling runs for each model, we applied 5 chains to each of 5 generated observation sequences. The HDP-HMM is unable to capture the non-Markovian duration statistics and so its state sampling error remains high, while the HDP-HSMM equipped with Poisson duration distributions is able to effectively learn the correct temporal model, including



(a) HDP-HMM **(b)** HDP-HSMM

Figure 3.12: State-sequence Hamming error of the HDP-HMM and HDP-HSMM with Poisson durations applied to data from an HSMM with Poisson durations. In each plot, the blue line indicates the error of the chain with the median error across 25 independent Gibbs chains, while the red dashed lines indicate the chains with the 10th and 90th percentile errors at each iteration. The jumps in the plot correspond to a change in the ranking of the 25 chains.

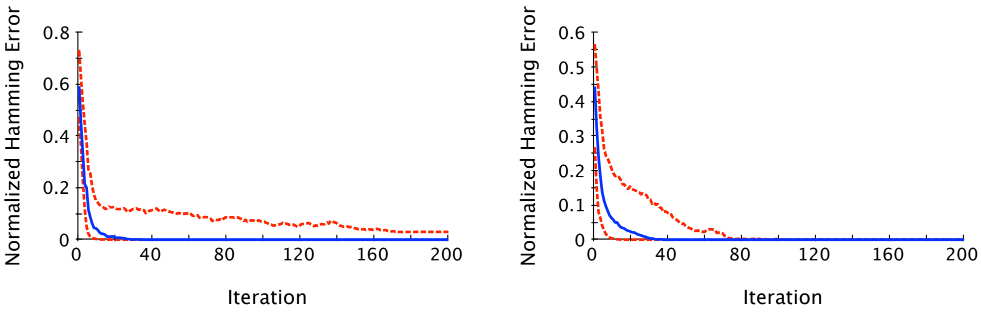


(a) HDP-HMM **(b)** HDP-HSMM

Figure 3.13: Number of states inferred by the HDP-HMM and HDP-HSMM with Poisson durations applied to data from a four-state HSMM with Poisson durations. In each plot, the blue line indicates the error of the chain with the median error across 25 independent Gibbs chains, while the red dashed lines indicate the chains with the 10th and 90th percentile errors at each iteration.

duration, transition, and emission parameters, and thus effectively separate the states and significantly reduce posterior uncertainty. The HDP-HMM also frequently fails to identify the true number of states, while the posterior samples for the HDP-HSMM concentrate on the true number; see Figure 3.13.

By setting the class of duration distributions to be a superclass of the class of geometric distributions, we can allow an HDP-HSMM model to learn an HMM from data when appropriate. One such distribution class is the class of negative binomial distri-



(a) HDP-HMM

(b) HDP-HSMM

Figure 3.14: The HDP-HSMM and HDP-HMM applied to data from an HMM. In each plot, the blue line indicates the error of the chain with the median error across 25 independent Gibbs chains, while the red dashed line indicates the chains with the 10th and 90th percentile error at each iteration.

butions, denoted NegBin(r, p), the discrete analog of the Gamma distribution, which covers the class of geometric distributions when $r = 1$. (We discuss negative binomial duration models further and develop specialized algorithms in Chapter 5.) By placing a (non-conjugate) prior over r that includes $r = 1$ in its support, we allow the model to learn geometric durations as well as significantly non-geometric distributions with modes away from zero. Figure 3.14 shows a negative binomial HDP-HSMM learning an HMM model from data generated from an HMM with four states. The observation distribution for each state is a 10-dimensional Gaussian, with parameters sampled i.i.d. from a Normal-Inverse-Wishart (NIW) prior, as defined in Example 2.2.9 of Section 2.2. The prior over r was set to be uniform on $\{1, 2, \ldots, 6\}$, and all other priors were chosen to be similarly non-informative. The sampler chains quickly concentrated at $r = 1$ for all state duration distributions. There is only a slight loss in mixing time for the HDP-HSMM compared to the HDP-HMM. This experiment demonstrates that with the appropriate choice of duration distribution the HDP-HSMM can effectively learn an HMM model.

■ 3.5.2 Power disaggregation

In this section we show an application of the HDP-HSMM factorial structure to an unsupervised power signal disaggregation problem. The task is to estimate the power draw from individual devices, such as refrigerators and microwaves, given an aggregated whole-home power consumption signal. This disaggregation problem is important for energy efficiency: providing consumers with detailed power use information at the device level has been shown to improve efficiency significantly, and by solving the disaggregation problem one can provide that feedback without instrumenting every individual device with monitoring equipment. This application demonstrates the utility of including duration information in priors as well as the significant speedup achieved with changepoint-based inference.

The power disaggregation problem has a rich history [119] with many proposed approaches for a variety of problem specifications. Some recent work [64] has considered applying factorial HSMMs to the disaggregation problem using an EM algorithm; our work here is distinct in that (1) we do not use training data to learn device models but instead rely on simple prior information and learn the model details during inference, (2) our states are not restricted to binary values and can model multiple different power modes per device, and (3) we use Gibbs sampling to learn all levels of the model. The work in Kim et al. [64] also explores many other aspects of the problem, such as additional data features, and builds a very compelling complete solution to the disaggregation problem, while we focus on the factorial time series modeling itself.

For our experiments, we used the REDD data set [66], which monitors many homes at high frequency and for extended periods of time. We chose the top 5 power-drawing devices (refrigerator, lighting, dishwasher, microwave, furnace) across several houses and identified 18 24-hour segments across 4 houses for which many (but not always all) of the devices switched on at least once. We applied a 20-second median filter to the data, and each sequence is approximately 5000 samples long. See Figure 3.15 for an example of the aggregated sequence data and its constituent sequences.

We constructed simple priors that set the rough power draw levels and duration statistics of the modes for several devices. For example, the power draw from home lighting changes infrequently and can have many different levels, so an HDP-HSMM with a bias towards longer negative-binomial durations is appropriate. Refrigerators tend to exhibit an "off" mode near zero Watts, an "on" mode near 100-140 Watts, and a "high" mode near 300-400 Watts, so our priors biased the refrigerator HDP-HSMM to have fewer modes and set the power levels accordingly. We encode such modes in the prior by adjusting the generative process so that some parameters are drawn from distinct prior distributions. To encode a prior mode near some μ_0, we simply generate a particular emission parameter as $\theta^{(i)} = \mu^{(i)} \sim \mathcal{N}(\mu_0, \sigma_0^2)$ with some uncertainty in the mode value represented by σ_0^2. Other emission parameters are generated independently and identically from a fixed background prior.

Our priors are summarized in Table 3.1. We write $\text{Gauss}(\mu_0, \sigma_0^2; \sigma^2)$ to denote a Gaussian observation distribution prior with a fixed variance of σ^2 and a prior over its mean parameter that is Gaussian distributed with mean μ_0 and variance σ_0^2. We write $\text{NegBin}(\alpha, \beta; r)$ to denote a prior over duration distributions where $p \sim \text{Beta}(\alpha, \beta)$ and r is fixed, and durations are then sampled from $\text{NegBin}(r, p)$. We did not truncate the duration distributions during inference, and we set the weak limit approximation

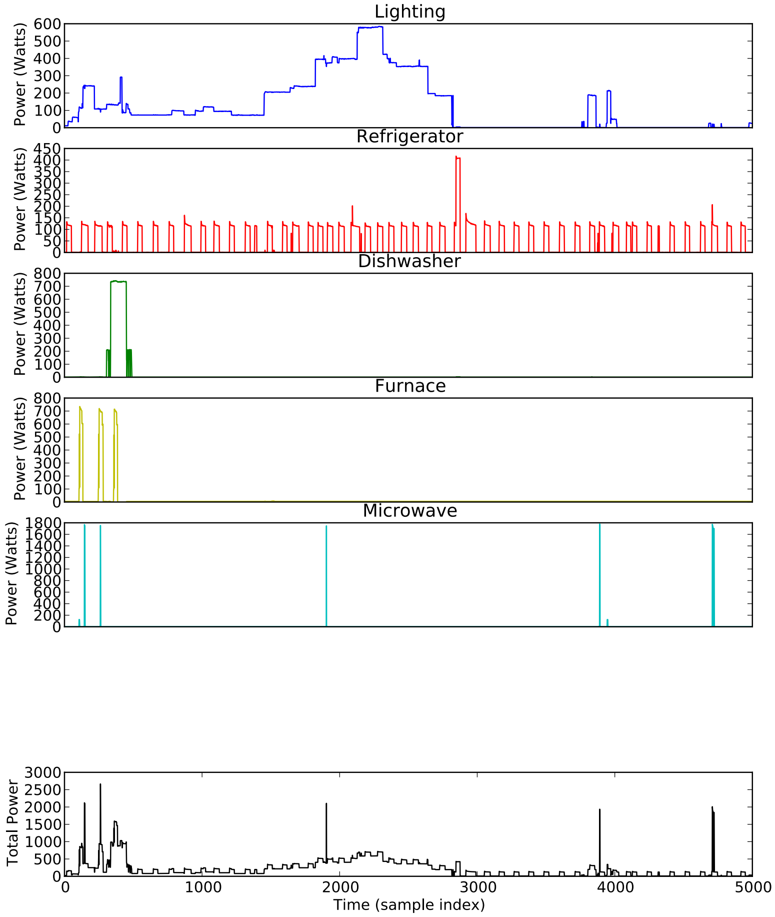


Figure 3.15: Example real data observation sequences for the power disaggregation experiments.

parameter L to be twice the number of expected modes for each device; for example, for the refrigerator device we set $L = 6$ and for lighting we set $L = 20$. We performed sampling inference independently on each observation sequence.

Device	Base Measures		Specific States	
	Observations	Durations	Observations	Durations
Lighting	Gauss$(300, 200^2; 5^2)$	NegBin$(5, 220; 12)$	Gauss$(0, 1; 5^2)$	NegBin$(5, 220; 12)$
Refrigerator	Gauss$(110, 50^2; 10^2)$	NegBin$(100, 600; 10)$	Gauss$(0, 1; 5^2)$ Gauss$(115, 10^2; 10^2)$ Gauss$(425, 30^2; 10^2)$	NegBin$(100, 600; 10)$ NegBin$(100, 600; 10)$ NegBin$(100, 600; 10)$
Dishwasher	Gauss$(225, 25^2; 10^2)$	NegBin$(100, 200; 10)$	Gauss$(0, 1; 5^2)$ Gauss$(225, 25^2; 10^2)$ Gauss$(900, 200^2; 10^2)$	NegBin$(1, 2000; 1)$ NegBin$(100, 200; 10)$ NegBin$(40, 500; 10)$
Furnace	Gauss$(600, 100^2; 20^2)$	NegBin$(40, 40; 10)$	Gauss$(0, 1; 5^2)$	NegBin$(1, 50; 1)$
Microwave	Gauss$(1700, 200^2; 50^2)$	NegBin$(200, 1; 50)$	Gauss$(0, 1; 5^2)$	NegBin$(1, 1000; 1)$

Table 3.1: Power disaggregation prior parameters for each device. Observation priors encode rough power levels that are expected from devices. Duration priors encode duration statistics that are expected from devices.

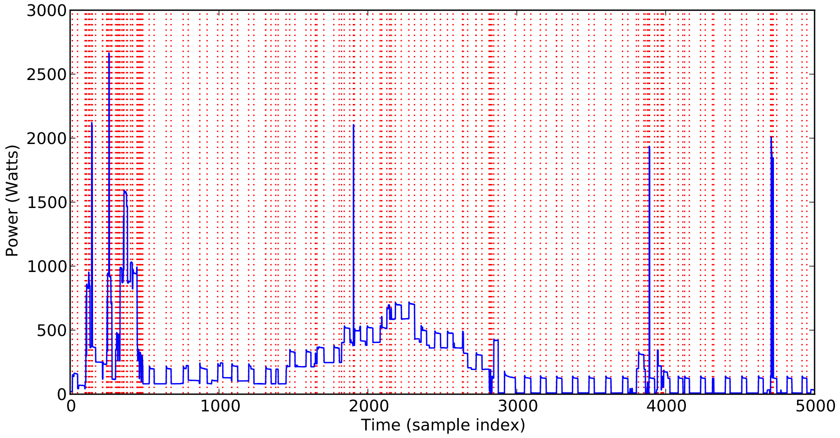


Figure 3.16: An total power observation sequence from the power disaggregation data set. Vertical dotted red lines indicate changepoints detected with a simple first-differences. By using the changepoint-based algorithms described in Section 3.4.4 we can greatly accelerate inference speed for this application.

As a baseline for comparison, we also constructed a factorial sticky HDP-HMM [31] with the same observation priors and with duration biases that induced the same average mode durations as the corresponding HDP-HSMM priors. We also compare to the factorial HMM performance presented in Kolter and Johnson [66], which fit device models using an EM algorithm on training data. For the Bayesian models, we performed inference separately on each aggregate data signal.

The set of possible changepoints is easily identifiable in these data, and a primary task of the model is to organize the jumps observed in the observations into an explanation in terms of the individual device models. By simply computing first differences and thresholding, we are able to reduce the number of potential changepoints we need to consider from 5000 to 100-200, and hence we are able to speed up label sequence resampling by orders of magnitude. See Figure 3.16 for an illustration.

To measure performance, we used the error metric of Kolter and Johnson [66]:

$$\text{Acc.} = 1 - \frac{\sum_{t=1}^{T}\sum_{i=1}^{K}\left|\hat{y}_t^{(i)} - y_t^{(i)}\right|}{2\sum_{t=1}^{T}\bar{y}_t}$$

where $\bar{y}_t$ refers to the observed total power consumption at time t, $y_t^{(i)}$ is the true power consumed at time t by device i, and $\hat{y}_t^{(i)}$ is the estimated power consumption. We produced 20 posterior samples for each model and report the median accuracy of the component emission means compared to the ground truth provided in REDD. We ran our experiments on standard desktop machines, and a sequence with about 200 detected

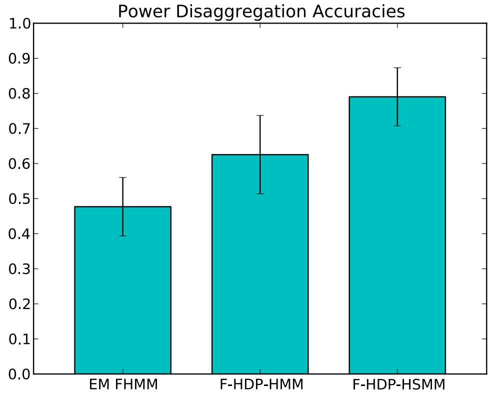


Figure 3.17: Overall accuracy comparison between the EM-trained FHMM of Kolter and Johnson [66], the factorial sticky HDP-HMM, and the factorial HDP-HSMM.

House	EM FHMM	F-HDP-HMM	F-HDP-HSMM
1	46.6%	69.0%	82.1%
2	50.8%	70.7%	84.8%
3	33.3%	67.3%	81.5%
6	55.7%	61.8%	77.7%
Mean	47.7%	67.2%	81.5%

Table 3.2: Performance comparison broken down by house.

changepoints would resample each component chain in 0.1 seconds, including block sampling the label sequence and resampling all observation, duration, and transition parameters. We collected samples after every 50 such iterations.

Our overall results are summarized in Figure 3.17 and Table 3.2. Both Bayesian approaches improved upon the EM-based approach because they allowed flexibility in the device models that could be fit during inference, while the EM-based approach fixed device model parameters that may not be consistent across homes. Furthermore, the incorporation of duration structure and prior information provided a significant performance increase for the HDP-HSMM approach. Detailed performance comparisons between the HDP-HMM and HDP-HSMM approaches can be seen in Figure 3.18. Finally, Figures 3.19 and 3.20 shows total power consumption estimates for the two models on two selected data sequences.

3.6 Summary

We have developed the HDP-HSMM and two Gibbs sampling inference algorithms, the weak limit and direct assignment samplers, uniting explicit-duration semi-Markov modeling with new Bayesian nonparametric techniques. These models and algorithms not

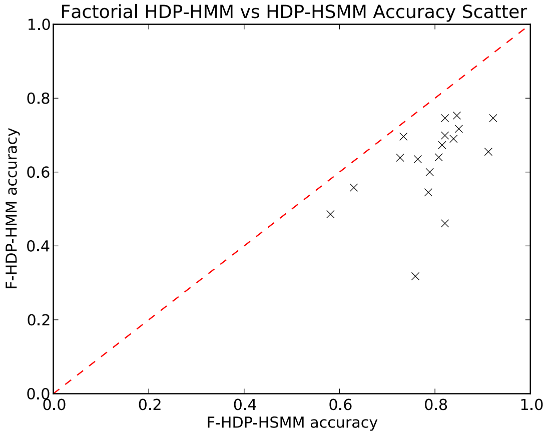


Figure 3.18: Performance comparison between the HDP-HMM and HDP-HSMM approaches broken down by data sequence.

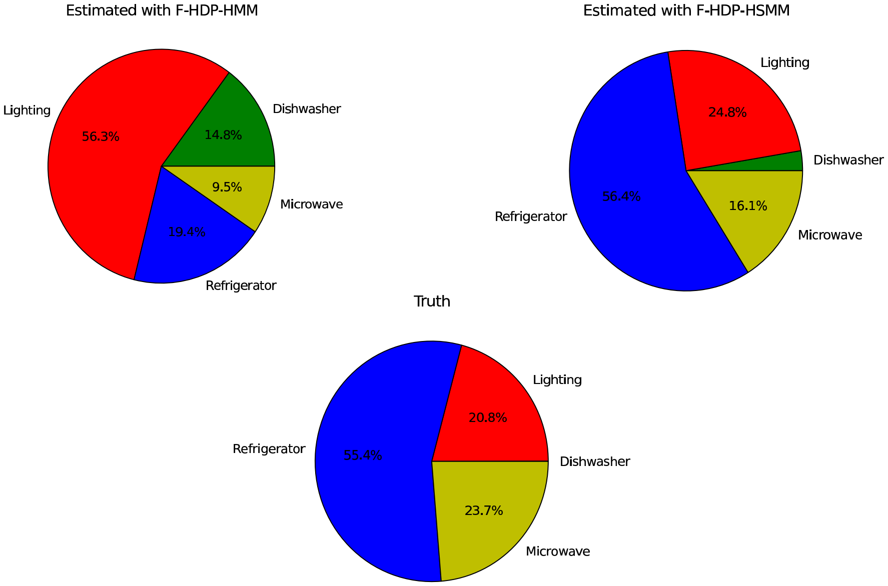


Figure 3.19: Estimated total power consumption for a data sequence where the HDP-HSMM significantly outperformed the HDP-HMM due to its modeling of duration regularities.

only allow learning from complex sequential data with non-Markov duration statistics in supervised and unsupervised settings, but also can be used as tools in constructing and performing infernece in larger hierarchical models. We have demonstrated the utility of the HDP-HSMM and the effectiveness of our inference algorithms with real and synthetic experiments.

In the following chapters, we build on these models and algorithms in several ways.

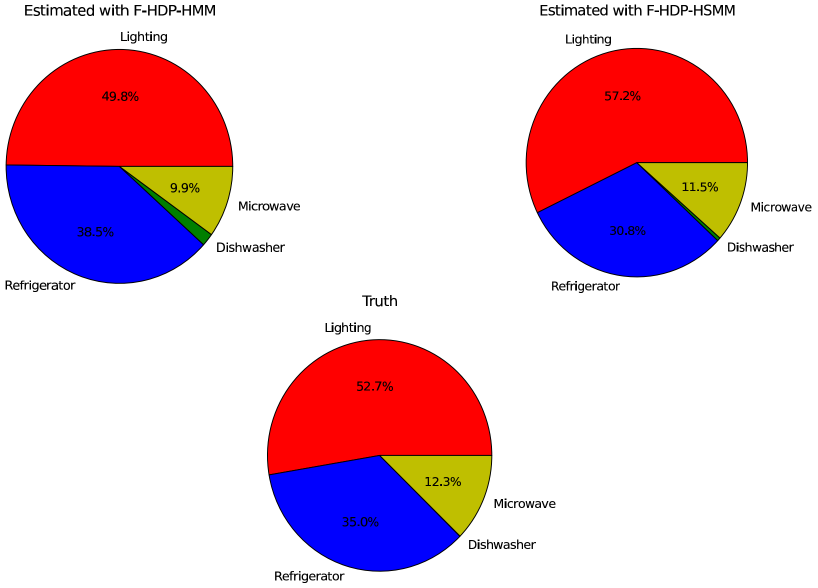


Figure 3.20: Estimated total power consumption for a data sequence where both the HDP-HMM and HDP-HSMM approaches performed well.

In the next two chapters, we address issues of scalability in HSMM and HDP-HSMM inference algorithms, including both the challenge of scaling HSMM message passing to longer sequence lengths and the challenge of scaling inference to large datasets with many observation sequences. In Chapter 6, we build on the HDP-HSMM and these scalable inference techniques to define a new Bayesian nonparametric model.