Bayesian Nonparametric Learning with semi-Markovian Dynamics

by

Matthew J Johnson

B.S. in Electrical Engineering and Computer Sciences
University of California at Berkeley, 2008

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author . . . .
Department of Electrical Engineering and Computer Science
May 21, 2010

Certified by . . . .
Alan S. Willsky
Edwin Sibley Webster Professor of Electrical Engineering
Thesis Supervisor

Accepted by . . . .
Terry P. Orlando
Chairman, Department Committee on Graduate Students

Bayesian Nonparametric Learning with semi-Markovian Dynamics

by

Matthew J Johnson

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2010, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

There is much interest in the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) as a natural Bayesian nonparametric extension of the ubiquitous Hidden Markov Model for learning from sequential and time-series data. However, in many settings the HDP-HMM's strict Markovian constraints are undesirable, particularly if we wish to learn or encode non-geometric state durations. We can extend the HDP-HMM to capture such structure by drawing upon explicit-duration semi-Markovianity, which has been developed in the parametric setting to allow construction of highly interpretable models that admit natural prior information on state durations.

In this thesis we introduce the explicit-duration Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) and develop posterior sampling algorithms for efficient inference. We also develop novel sampling inference for the Bayesian version of the classical explicit-duration Hidden semi-Markov Model. We demonstrate the utility of the HDP-HSMM and our inference methods on synthetic data as well as experiments on a speaker diarization problem and an example of learning the patterns in Morse code.

Thesis Supervisor: Alan S. Willsky
Title: Edwin Sibley Webster Professor of Electrical Engineering

Contents

Chapter 1

Introduction

Given a set of sequential data in an unsupervised setting, we often aim to infer meaningful states, or "topics," 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 speech patterns [2]. In analyzing DNA sequences, we may want to identify and segment region types using prior knowledge about region length distributions [7, 12]. 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 such learning problems in sequential data, but they have two significant disadvantages: (1) state duration distributions are necessarily restricted to a geometric form that is not appropriate for many real-world data, and (2) the number of hidden states must be set a priori so that model complexity is not inferred from data in a Bayesian way.

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 inferring arbitrarily large state complexity from data [14]. However, the HDP-HMM does not address the issue of non-Markovianity in real data. The

Markovian disadvantage is even 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 [2].

One approach to avoiding the rapid-switching problem is the Sticky HDP-HMM [2], which introduces a learned 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 [5] induces non-Markovian state durations at the coarser levels of its state hierarchy, but even the coarser levels are constrained to have a sum-of-geometrics form, and hence it can be difficult to incorporate prior information.

These potential improvements to the HDP-HMM motivate the investigation into explicit-duration semi-Markovianity, which has a history of success in the parametric setting (e.g. [16]). In this thesis, 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 incorporation of general duration distributions. In addition, the sampling techniques we develop for the Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) provide new approaches to inference in HDP-HMMs that can avoid some of the difficulties which result in slow mixing rates.

The remainder of this thesis is organized as follows. In Chapter 2, we provide background information relevant to this thesis. In particular, we describe HMM modeling and the salient points of Bayesian learning and inference. We also provide a description of explicit-duration HSMMs and existing HSMM message-passing algorithms, which we use to build an efficient Bayesian inference algorithm in the sequel. Chapter 2 also provides background on the nonparametric priors and inference techniques we use to extend the classical HSMM: the Dirichlet Process, the Hierarchical Dirichlet Process, and the Hierarchical Dirichlet Process Hidden Markov Model.

In Chapter 3 we develop new models and inference methods. First, we develop a Gibbs sampling algorithm for inference in Bayesian constructions of finite HSMMs. Next, we describe the HDP-HSMM, which combines Bayesian nonparametric priors with semi-Markovian expressiveness. Finally, we develop efficient Gibbs sampling algorithms for inference in the HDP-HSMM, including both a collapsed sampler, in which we analytically marginalize over the nonparametric Dirichlet Process priors, and a practical approximate blocked sampler based on the standard weak-limit approximation to the Dirichlet Process.

Chapter 4 demonstrates the effectiveness of the HDP-HSMM on both synthetic and real data using the blocked sampling inference algorithm. In synthetic experiments, we demonstrate that our sampler mixes very quickly on data generated by both HMMs and HSMMs and accurately learns parameter values and state cardinality. 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. Next, we present an experiment on Morse Code audio data, in which the HDP-HSMM is able to learn the correct state primitives while an HDP-HMM confuses short- and long-tone states because it is unable to incorporate duration information appropriately. Finally, we apply the HDP-HSMM to a speaker diarization problem, for which we achieve competitive performance and rapid mixing.

In Chapter 5 we conclude the thesis and discuss some avenues for future investigation.

Chapter 2

Background

2.1 Bayesian Hidden Markov Models (HMMs)

The Hidden Markov Model is a general model for sequential data. Due to its versatility and tractability, it has found wide application and is extensively treated in both textbooks, including [1], and tutorial papers, particularly [10]. This section provides a brief introduction to the HMM, with emphasis on the Bayesian treatment.

2.1.1 Model Specification

The core of the HMM consists of two layers: a layer of hidden *state* variables and a layer of *observation* or *emission* variables. The relationships between the variables in both layers is summarized in the graphical model in Figure 2-1. Each layer consists of a sequence of random variables, and the indexing corresponds to the sequential aspect of the data (e.g. time indices).

The hidden state sequence, $x = (x_t)_{t=1}^{T}$ for some length $T \in \mathbb{N}$, is a sequence of random variables on a finite alphabet, i.e. $x_t \in \mathcal{X} = [N] \triangleq \{1, 2, \ldots, N\}$, that forms a Markov chain:

$$\forall t \in [T-1] \quad p(x_{t+1}|x_1, x_2, \ldots, x_t) = p(x_{t+1}|x_t). \tag{2.1}$$

Thus, if the indices are taken to be time indices, the state variable summarizes the

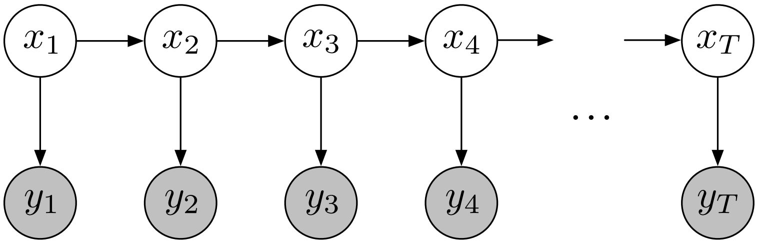


Figure 2-1: Basic graphical model for the HMM. Parameters for the transition, emission, and initial state distributions are not shown as random variables, and thus this diagram is more appropriate for a Frequentist framework.

relevant history of the process in the sense that the future is statistically independent of the past given the present. It is the Markovian assumption that is at the heart of the simplicity of inference in the HMM: if the future were to depend on more than just the present, computations of interest would be more complex.

It is necessary to specify the conditional relationship between sequential hidden states via a transition distribution $p(x_{t+1}|x_t, \pi)$, where π represents parameters of the conditional distribution. Since the states are taken to be discrete in an HMM (as opposed to, for example, a linear dynamical system), the transition distribution is usually multinomial and is often parameterized by a row-stochastic matrix $\pi = (\pi_{ij})_{i,j=1}^N$ where $\pi_{ij} = p(x_{t+1} = j|x_t = i)$ and N is the *a priori* fixed number of possible states. The ith row gives a parameterization of the transition distribution out of state i, and so it is natural to think of π in terms of its rows:

$$\pi = \begin{bmatrix} \qquad \pi_1 \qquad \\ \pi_2 \\ \vdots \\ \pi_N \end{bmatrix} \tag{2.2}$$

We also must specify an initial state distribution, $p(x_1|\pi_0)$, where the π_0 parameter is often taken to be a vector directly encoding the initial state probabilities. We will use the notation $\{\pi_i\}_{i=0}^N$ to collect both the transition and initial state parameters into a single set, though we will often drop the explicit index set.

The second layer of the HMM is the observation (or emission) layer, $y = (y_t)_{t=1}^T$.

However, the variables do not form a Markov chain. In fact, there are no marginal independence statements for the observation variables: the undirected graphical model that corresponds to marginalizing out the hidden state variables is fully connected. This result is a feature of the model: it is able to explain very complex statistical relationships in data, at least with respect to conditional independencies. However, the HMM requires that the observation variables be conditionally independent given the state sequence. More precisely, it requires

$$\forall t \in [T] \quad y_t \perp\!\!\!\perp \{y_{\backslash t}\} \cup \{x_{\backslash t}\} | x_t \tag{2.3}$$

where the notation $(y_{\backslash t})$ denotes the sequence excluding the t^{th} element, and $a \perp\!\!\!\perp b|c$ indicates random variables a and b are independent given random variable c. Given the corresponding state variable at the same time instant an observation is rendered independent from all other observations and states, and in that sense the state "fully explains" the observation.

One must specify the conditional relationship between the states and observations, i.e. $p(y_t|x_t, \theta)$, where θ represents parameters of the emission distribution. These distributions can take many forms, particularly because the observations themselves can be taken from any (measurable) space. As a concrete example, one can take the example that the observation space is some Euclidean space $\mathbb{R}^k$ for some k and the emission distributions are multidimensional Gaussians with parameters indexed by the state, i.e. in the usual Gaussian notation[1], $\theta = \{\theta_i\}_{i=1}^N = \{(\mu_i, \Sigma_i)\}_{i=1}^N$.

With the preceding distributions defined, we can write the joint probability of the hidden states and observations in an HMM as

$$p((x_t), (y_t)|\{\pi_i\}, \theta) = p(x_1|\pi_0) \left(\prod_{t=1}^{T-1} p(x_{t+1}|x_t|\pi) \right) \left(\prod_{t=1}^{T} p(y_t|x_t|\theta) \right). \tag{2.4}$$

The Bayesian and Frequentist formulations of the HMM diverge in their treatment of the *parameters* $\{\pi_i\}$ and θ. A Frequentist framework would treat the parameters

[1] By usual Gaussian notation, we mean μ is used to represent the mean parameter and Σ the covariance matrix parameter, i.e. $\mathcal{N}(\mu, \Sigma)$.

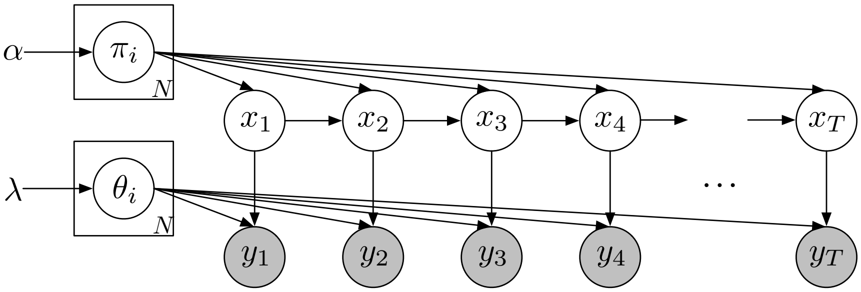


Figure 2-2: Basic graphical model for the Bayesian HMM. Parameters for the transition, emission, and initial state distributions are random variables. The λ and α symbols represent hyperparameters for the prior distributions on state-transition parameters and emission parameters, respectively.

as deterministic quantities to be estimated while a Bayesian framework would model them as random variables with prior distributions, which are themselves parameterized by *hyperparameters*. This thesis is concerned with utilizing a nonparametric Bayesian framework, and so we will use the Bayesian HMM formulation and treat parameters as random variables.

Thus we can write the joint probability of our Bayesian HMM as

$$\begin{aligned} p((x_t),(y_t),\{\pi_i\},\theta|\alpha,\lambda) &= p(\theta|\lambda)p(\{\pi_i\}|\alpha)p((x_t),(y_t)|\{\pi_i\},\theta) && (2.5)\\ &= \prod_{i=1}^{N} p(\theta_i|\lambda)p(\{\pi_i\}|\alpha)p(x_1|\pi_0)\left(\prod_{t=1}^{T-1} p(x_{t+1}|x_t|\pi)\right)\left(\prod_{t=1}^{T} p(y_t|x_t|\theta)\right) && (2.6)\end{aligned}$$

for some observation parameter distribution $p(\theta_i|\lambda)$ and a prior on transitions $p(\{\pi_i\}|\lambda)$. A graphical model is given in Figure 2-2, where parameter random variables are shown in their own nodes. Hyperparameters are shown without nodes and assumed to be fixed and known a priori.

2.1.2 Posterior Inference via Gibbs Sampling

So far we have specified the HMM as a restricted class of probability distributions over sequences. From a practical standpoint, we are interested in the issues that arise when applying the model to data, i.e. finding some representation of the posterior distribution over states and parameters when conditioning on the observations:

$$p((x_t), \theta, \{\pi_i\}|(y_t), \alpha, \lambda). \tag{2.7}$$

We can perform posterior inference in the HMM with a Gibbs sampling algorithm, which allows us to construct samples from the posterior distribution by iteratively re-sampling some variables conditioned on their Markov blanket. We do not provide a general background for Gibbs sampling in this thesis, but the reader can find a thorough discussion in [1].

An iteration of our Gibbs sampler samples the following conditional random variables, which are ordered arbitrarily:

- $(x_t)|\theta, \{\pi_i\}, (y_t)$
- $\{\pi_i\}|\alpha, \theta, (x_t), (y_t)$
- $\theta|\lambda, (x_t), (y_t)$

For example, when we sample the conditional random variable $\theta|\lambda, (x_t), (y_t)$, that means we update the value of θ to be a draw from its conditional distribution given the current values of $\lambda, (x_t), y_t$. Sampling $\{\pi_i\}$ and θ from their conditional distributions is a standard problem which depends on the specific model distributions chosen; such issues are thoroughly described in, e.g., [1]. However, sampling (x_t) is of particular importance and is not a standard procedure, and so we describe it in detail in the next section.

Block Sampling $(x_t)|\theta, \{\pi_i\}, (y_t)$ using Message Passing

To draw a conditional sample of the entire (x_t) sequence at once, we exploit the Markov structure and use dynamic programming on the chain with the well-known "forwards-backwards" (or "alpha-beta") message passing algorithm for the HMM. We define the messages as

$$\begin{aligned} \alpha_t(x_t) &\triangleq p(y_1, \ldots, y_t, x_t) && t = 1, 2, \ldots, T && (2.8) \\ \beta_t(x_t) &\triangleq p(y_{t+1}, \ldots, y_T | x_t) && t = 1, 2, \ldots, T-1 && (2.9) \\ \beta_T(x_T) &\triangleq 1 && && (2.10) \end{aligned}$$

where we have dropped the notation for conditioning on parameters θ and $\{\pi_i\}$ for convenience. The $\alpha_t(x_t)$ message is the probability of the data ending in state x_t, and the $\beta_t(x_t)$ message is the probability of future data given a starting state of x_t. We also note that for any t we have

$$p(x_t | y_1, \ldots, y_T) \propto \alpha_t(x_t)\beta_\tau(x_t). \tag{2.11}$$

The α and β messages can be computed recursively through an easy derivation, which can be found in [6]:

$$\begin{aligned} \alpha_t(x_{t+1}) &= p(y_1, \ldots, y_{t+1}, x_{t+1}) && (2.12) \\ &= p(y_1, \ldots, y_t | x_{t+1}) p(y_{t+1} | x_{t+1}) p(x_{t+1}) && (2.13) \\ &= \sum_{x_t} p(y_1, \ldots, y_t | x_t) p(x_{t+1} | x_t) p(x_t) p(y_{t+1} | x_{t+1}) && (2.14) \\ &= \sum_{x_t} \alpha_t(x_t) p(x_{t+1} | x_t) p(y_{t+1} | x_{t+1}). && (2.15) \end{aligned}$$

With similar manipulations, we can derive

$$\beta_t(x_t) = \sum_{x_{t+1}} \beta_t(x_{t+1}) p(x_{t+1} | x_t) p(y_{t+1} | x_{t+1}). \tag{2.16}$$

Again, we have dropped the notation for explicitly conditioning on parameters.

We can efficiently draw $(x_t)|\theta, \{\pi_i\}, (y_t)$ using only the β messages as follows.

First, we note

$$p(x_1|\theta, \{\pi_i\}, (y_t)) \propto p(x_1|\{\pi_i\})p((y_t)|x_1, \theta, \{\pi_i\}) \tag{2.17}$$

$$= p(x_1|\pi_0)\beta_1(x_1) \tag{2.18}$$

and hence we can draw $x_1|\theta, \{\pi_i\}, (y_t)$ by drawing from the normalized element-wise product of β_1 and π_0. Supposing we draw $x_1 = \bar{x}_1$, we can write

$$p(x_2|\theta, \{\pi_i\}, (y_t), x_1 \propto \bar{x}_1) \propto p(x_2|\{\pi_i\}, x_1 = \bar{x}_1)p((y_{2:T})|x_2, \theta) \tag{2.19}$$

$$= p(x_2|x_1 = \bar{x}_1, \pi_{\bar{x}_1})\beta_2(x_2) \tag{2.20}$$

and hence we can draw $x_2|\theta, \{\pi_i\}, (y_t), x_1 = \bar{x}_1$ by drawing from the normalized element-wise product of $\pi_{\bar{x}_1}$ and β_2. We can recurse this procedure to draw a block sample of $(x_t)|\theta, \{\pi_i\}, (y_t)$.

2.1.3 Summary

In this section we have described the Bayesian treatment of the well-known Hidden Markov Model as well as the salient points of posterior inference with Gibbs sampling. There are two main disadvantages to the HMM that this thesis seeks to address simultaneously: the lack of explicit duration distributions and the issue of choosing the number of states from the data. The following background sections on Hidden semi-Markov Models and Bayesian nonparametric methods describe separate approaches to each of these two issues.

2.2 Explicit-Duration Hidden Semi-Markov Models (HSMMs)

There are several modeling approaches to semi-Markovianity [8], but here we focus on *explicit duration* semi-Markovianity; i.e., we are interested in the setting where each state's duration is given an explicit distribution.

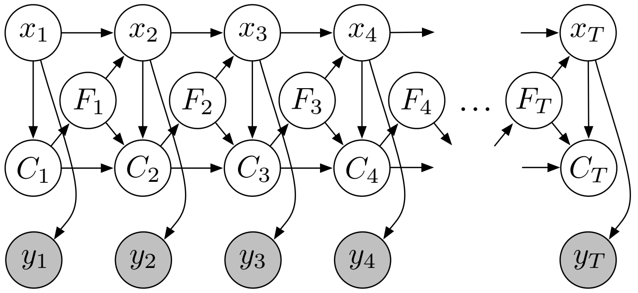


Figure 2-3: A graphical model for the HSMM with explicit counter and finish nodes.

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. It can be cumbersome to draw the process into a proper graphical model, but one compelling representation in [8] is to add explicit "timer" and "finish" variables, as depicted in Figure 2-3. The (C_t) variables serve to count the remaining duration times, and are deterministically decremented to zero. The (F_t) variables are binary-valued, $F_t = 1$ if and only if there is a Markov transition at time $t + 1$. Hence, $F_t = 1$ causes C_{t+1} to be sampled from the duration distribution of x_{t+1}, and the subsequent values $C_{t+2}, C_{t+3}, \ldots$ count down until reaching zero, at which point the process repeats.

An equivalent and somewhat more intuitive picture is given in Figure 2-4 (which also appears in [8]), though the number of nodes in the model is itself random. In this picture, we see there is a standard Markov chain on "super-state" nodes, $(z_s)_{s=1}^{S}$, and these super-states in turn emit random-length segments of observations, of which we observe the first T. The symbol D_i is used to denote the random length of the observation segment of super-state i for $i = 1, \ldots, S$. The "super-state" picture separates the Markovian transitions from the segment durations, and is helpful in building sampling techniques for the generalized models introduced in this thesis.

It is often taken as convention that state self-transitions should be ruled out in an HSMM, because if a state can self-transition then the duration distribution does not fully capture a state's possible duration length. We adopt this convention, which

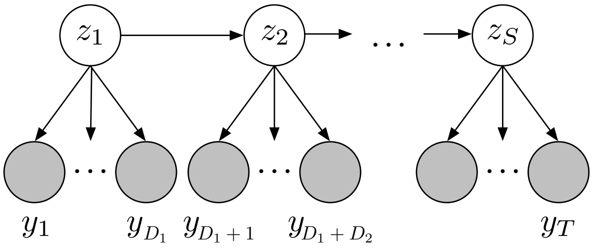


Figure 2-4: HSMM interpreted as a Markov chain on a set of super-states, $(z_s)_{s=1}^S$. The number of shaded nodes associated with each z_s is random, drawn from a state-specific duration distribution.

has a significant impact on the inference algorithms described in Section 3. 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. This censoring convention allows for slightly simpler formulae and computations, and thus is adopted in this paper. We do, however, assume the observations begin on a segment boundary. For more details and alternative conventions, see [4].

It is possible to perform efficient 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 "backwards" messages are crucial in the development of efficient sampling inference in Section 3 because the message values can be used to efficiently compute the posterior information necessary to block-sample the hidden state sequence (x_t), and so we briefly describe the relevant part of the existing HSMM message-passing algorithm. As derived in [8], we can define and compute the backwards message from t to $t+1$ as follows:

$$\beta_t(i) \triangleq p(y_{t+1:T}|x_t = i, F_t = 1) \tag{2.21}$$

$$= \sum_j \beta_t^*(j) p(x_{t+1} = j | x_t = i) \tag{2.22}$$

$$\beta_t^*(i) \triangleq p(y_{t+1:T}|x_{t+1} = i, F_t = 1) \tag{2.23}$$

$$= \sum_{d=1}^{T-t} \beta_{t+d}(i) \underbrace{p(D_{t+1} = d | x_{t+1} = i)}_{\text{duration prior term}} \cdot \underbrace{p(y_{t+1:t+d}|x_{t+1} = i, D_{t+1} = d)}_{\text{likelihood term}} \tag{2.24}$$

$$+ \underbrace{p(D_{t+1} > T - t | x_{t+1} = i) p(y_{t+1:T}|x_{t+1} = i, D_{t+1} > T - t)}_{\text{censoring term}} \tag{2.25}$$

$$\beta_T(i) \triangleq 1 \tag{2.26}$$

where we have split the messages into β and β^* components for convenience and used $y_{k_1:k_2}$ to denote $(y_{k_1}, \ldots, y_{k_2})$. Also note that we have used D_{t+1} to represent the duration of the segment beginning at time $t+1$. The conditioning on the parameters of the distributions is suppressed from the notation. This backwards message-passing recursion is similar to that of the HMM, and we will find that we can use the values of $p(y_{t+1:T}|x_t = i, F_t = 1)$ for each possible state i in the efficient forward-sampling algorithm of Section 3.

The $F_t = 1$ condition indicates 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 backwards message at $t+d$ to provide aggregate future information given a boundary just after $t+d$. The final additive term in the expression for $\beta_t^*(i)$ is described in [4]; 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.

This message passing algorithm will be a subroutine in our Gibbs sampling algorithm; more specifically, it will be a step in block-resampling the state sequence (x_t) from its posterior distribution. Though a very similar technique is used in HMM Gibbs samplers, it is important to note the significant differences in computational cost between the HMM and HSMM message computations. The greater expressivity of the HSMM model necessarily increases the computational cost of the message passing algorithm: the above message passing requires $\mathcal{O}(T^2N + TN^2)$ 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 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 $\mathcal{O}(Td_{\max}N^2)$. Such truncations are often natural because both the duration prior term and the segment likelihood term contribute to the product rapidly vanishing with sufficiently large d. 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 is offset because HSMM samplers often need far fewer total iterations to converge (see Section 3).

Bayesian inference in an HDP-HSMM via Gibbs sampling is a novel contribution of this thesis, and it is discussed in Section 3.1.

2.3 The Dirichlet Process

The Dirichlet Process is a random process that has found considerable use as a prior in Bayesian nonparametrics. It is an extension of the Dirichlet distribution to general measurable spaces, and it possesses several desirable properties. In particular, it describes a distribution over infinitely many "clusters" or "topics," allowing us to infer and mix over arbitrary degrees of model complexity that grow with the amount of data.

In this section, we define the Dirichlet Process, outline its properties, and describe statistical inference procedures based on sampling. We also describe the prototypical Dirichlet Process Mixture Model.

2.3.1 Defining the Dirichlet Process

Definition (Implicit) Let $(\Omega, \mathcal{B})$ be a measurable space, H be a probability measure on that space, and α_0 be a positive real number. A Dirichlet Process (DP) is the distribution of a random probability measure G over $(\Omega, \mathcal{B})$ if and only if for any finite (disjoint) partition $(A_1, \ldots, A_r)$ of Ω we have

$$(G(A_1), \ldots, G(A_r)) \sim \mathrm{Dir}(\alpha_0 H(A_1), \ldots, \alpha_0 H(A_r)) \tag{2.27}$$

where Dir denotes the standard Dirichlet distribution. We write $G \sim DP(\alpha_0, H)$ and call α_0 the *concentration parameter* and H the *base measure* [13].

Note that the preceding definition is implicit rather than constructive. However, it implies the following properties:

Property 1. $\mathbb{E}[G(A)] = H(A) \ \ \forall A \in \mathcal{B}$. This property follows immediately from the expectation properties of the standard Dirichlet distribution.

Property 2. If we have samples $\{\theta_i\}_{i=1}^{N}$ drawn according to

$$G|H, \alpha_0 \sim \mathrm{DP}(H, \alpha_0) \tag{2.28}$$

$$\theta_i|G \sim G \qquad\qquad i = 1, \ldots, N, \tag{2.29}$$

i.e., we draw a random probability measure G from a Dirichlet Process and then draw samples $\{\theta_i\}$ from G, then we have

$$(G(A_1), \ldots, G(A_r))|\theta_1, \ldots, \theta_n \sim \mathrm{Dir}(\alpha_0 H(A_1) + N_1, \ldots, \alpha_0 H(A_r) + N_r) \tag{2.30}$$

for any partition, where N_k counts the number of samples that fall into partition element k. This property follows from the standard Dirichlet conjugacy [13].

Property 3. Since each A_k partition element can be arbitrarily small around some θ_i that falls into it, we note that the posterior must have atoms at the observed $\{\theta_i\}$ values. Furthermore, by conjugacy it still respects the same finite partition Dirichlet property as in the definition, and so the posterior is also a DP:

$$G|\{\theta_i\}, \alpha_0, H \sim DP\left(\alpha_0 + \sum_k n_k, \frac{1}{\alpha + N}\left(\alpha_0 H + \sum_i \delta_{\theta_i}\right)\right) \tag{2.31}$$

where δ_{θ_i} represents an atom at θ_i.

Property 4.

$$\lim_{N\to\infty} \mathbb{E}\left[G(T)|\{\theta_i\}, \alpha_0, H\right] = \sum_k \pi_k \delta_{\bar{\theta}_k}(T) \tag{2.32}$$

where $\{\bar{\theta}_k\}$ are the distinct values of $\{\theta_i\}$ and $\pi_k \triangleq \lim_{N\to\infty} \frac{N_k}{N}$. This property follows from noting that, due to Property 3, the atomic empirical measure component grows in total mass compared to the base measure in the posterior parameter, and since the expectation of a DP is equivalent in measure to its parameter (i.e., if $G \sim \mathrm{DP}(H, \alpha_0)$ then $\mathbb{E}[G(A)] = H(A)$ for any measurable A), we have that the expectation of the posterior must go to the empirical distribution in the limit.

Property 5. $G \sim DP(\alpha_0, H)$ is discrete almost surely. This result is stated without justification here, but it is described in [13].

These properties, which are derived directly from the implicit definition, already illuminate several desirable properties about the DP: namely, its relationship to the standard Dirichlet distribution (but with more general conjugacy), its consistency in the sense that it is clear how the posterior converges to the empirical distribution, and the fact that draws from a DP are discrete with probability 1. Furthermore, we can also see the reinforcement property in which more common samples increasingly dominate the posterior. However, while several properties are apparent, the preceding definition is implicit and does not inform us how to construct a sample G distributed according to $G|H, \alpha_0 \sim \mathrm{DP}(H, \alpha_0)$.

There is a constructive definition of the Dirichlet Process, which is equivalent with probability 1, referred to as the *stick breaking* construction. First, we define a stick breaking process with parameter α_0 as

$$\rho_k|\alpha_0 \sim \text{Beta}(1, \alpha_0) \qquad k = 1, 2, \ldots \tag{2.33}$$

$$\beta_k \triangleq \rho_k \prod_{\ell=1}^{k-1}(1 - \rho_\ell). \tag{2.34}$$

For a definition of the Beta distribution, see [1]. To describe this process we write simply $\beta \sim \text{GEM}(\alpha_0)$. The preceding is referred to as a stick breaking process because it can be visualized as breaking a stick of unit length into pieces, with each piece a $\text{Beta}(1, \alpha_0)$ proportion of the remaining length. We can then use these weights to construct $G \sim \text{DP}(H, \alpha_0)$ [3]:

$$\theta_k|H \sim H \qquad k = 1, 2, \ldots \tag{2.35}$$

$$G \triangleq \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k} \tag{2.36}$$

where δ_{θ_k} represents an atom at θ_k. To summarize, we have the following definition:

Definition (Stick Breaking) Let $(\omega, \mathcal{B})$ be a measurable space, H be a probability measure on that space, and α_0 be a positive real number. A Dirichlet Process is the distribution of a random probability measure G constructed as

$$\beta|\alpha_0 \sim \text{GEM}(\alpha_0) \tag{2.37}$$

$$\theta_k|H \sim H \qquad k = 1, 2, \ldots \tag{2.38}$$

$$G \triangleq \sum_{k=1}^{\infty} \beta_k \delta_{\theta_k}. \tag{2.39}$$

The stick breaking construction of a Dirichlet Process draw provides an alternative parameterization of the DP in terms of unique atom components. This parameterization can be useful not only to explicitly construct a measure drawn from a DP, but also in the design of sampling procedures which marginalize over G, as we discuss in the next section.

2.3.2 Drawing Samples from a DP-distributed Measure

We can draw samples from a measure G distributed according to the Dirichlet Process without instantiating the measure itself. That is, we can generate samples $\{\theta_i\}_{i=1}^N$ that are distributed as

$$G|H, \alpha_0 \sim \mathrm{DP}(H, \alpha_0) \tag{2.40}$$

$$\theta_i|G \sim G \qquad\qquad i = 1, \ldots, N \tag{2.41}$$

where we effectively marginalize out G. Using the posterior properties described previously, we can consider the predictive distributions on samples when we integrate out the measure G [13]:

$$\mathbb{P}(\theta_{N+1} \in A|\theta_1, \ldots, \theta_N) = \mathbb{E}\left[G(A)|\theta_1, \ldots, \theta_N\right] \tag{2.42}$$

$$= \frac{1}{\alpha_0 + N}\left(\alpha_0 H(A) + \sum_{i=1}^{N} \delta_{\theta_i \in A}\right) \tag{2.43}$$

The final line describes a *Polya urn* scheme [13] and thus allows us to draw samples from a Dirichlet process. First, we draw $\theta_1|H \sim H$. To draw $\theta_{i+1}|\theta_1, \ldots, \theta_i, H$, we choose to sample a new value with probability $\frac{\alpha_0}{\alpha_0+i}$, in which case we draw $\theta_{i+1}|\theta_1, \ldots, \theta_i, H \sim H$, or we choose to set $\theta_{i+1} = \theta_j$ for all $j = 1, \ldots, i$ with equal probability $\frac{1}{\alpha_0+i}$. This Polya urn procedure will generate samples as if they were drawn from a measure drawn from a Dirichlet Process, but clearly we do not need to directly instantiate all or part of the (infinite) measure.

However, the Polya urn process is not used in practice because it exhibits very slow mixing rates in typical models. This issue is a consequence of the fact that there may be repeated values in the $\{\theta_i\}$, leading to fewer conditional independencies in the model.

We can derive another sampling scheme that avoids the repeated-value problem by following the stick breaking construction's parameterization of the Dirichlet Process. In particular, we examine predictive distributions for both a label sequence,

$\{z_i\}_{i=1}^N$, and a sequence of distinct atom locations, $\{\bar{\theta}_k\}_{k=1}^\infty$. We equivalently write our sampling scheme for $\{\theta_i\}_{i=1}^N$ as:

$$\beta|\alpha_0 \sim \text{GEM}(\alpha_0) \tag{2.44}$$

$$\bar{\theta}_k|H \sim H \qquad k = 1, 2, \ldots \tag{2.45}$$

$$z_i|\beta \sim \beta \qquad i = 1, 2, \ldots, N \tag{2.46}$$

$$\theta_i \triangleq \theta_{z_i} \qquad i = 1, 2, \ldots, N \tag{2.47}$$

where we have interpreted β to be a measure over the natural numbers.

If we examine the predictive distribution on the labels $\{z_i\}$, marginalizing out β, we arrive at a description of the *Chinese Restaurant Process* (CRP) [13]. First, we set $z_1 = 1$, representing the first customer sitting at its own table, in the language of the CRP. When the $(i+1)$th customer enters the restaurant (equivalently, when we want to draw $z_{i+1}|z_1, \ldots, z_i$), it sits at a table proportional to the number of customers already at that table or starts its own table with probability $\frac{\alpha_0}{\alpha_0+i}$. That is, if the first i customers occupy K tables labeled as $1, 2, \ldots, K$, then

$$p(z_{i+1} = k) = \begin{cases} \frac{N_k}{\alpha_0+i} & k = 1, 2, \ldots, K \\ \frac{\alpha_0}{\alpha_0+i} & k = K+1 \end{cases} \tag{2.48}$$

where N_k denotes the number of customers at table k, i.e. $N_k = \sum_{j=1}^i \mathbf{1}[z_j = k]$. Each table is served a dish sampled i.i.d. from the prior, i.e. $\bar{\theta}_i|H \sim H$, and all customers at the table share the dish.

The Chinese Restaurant Process seems very similar to the Polya urn process, but since we separate the labels from the parameter values, we have[2] that $\theta_i \perp\!\!\!\perp \theta_j$ if $z_i \neq z_j$. In terms of sampling inference, this parameterization allows us to re-sample entire tables (or components) at a time by re-sampling $\bar{\theta}_i$ variables, whereas with the Polya urn procedure the θ_i for each data point had to be moved independently.

[2] For this independence statement we also need H such that $\theta_i \neq \theta_j$ a.s. for $i \neq j$, i.e. that independent draws from H yield distinct values with probability 1.

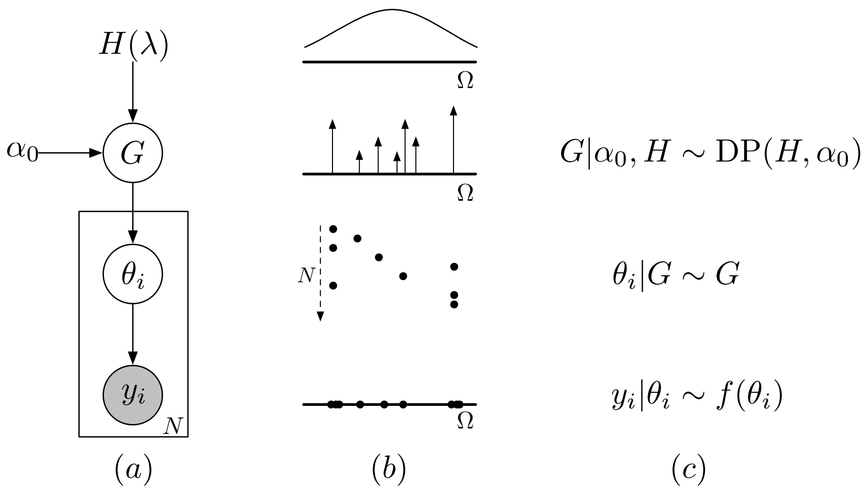


Figure 2-5: Dirichlet Process Mixture Model: (a) graphical model, where the observation nodes are shaded; (b) depiction of sampled objects in the DPMM; (c) corresponding generative process.

2.3.3 The Dirichlet Process Mixture Model

We can construct a DP Mixture Model (DPMM) much as we construct a standard Dirichlet mixture model [1], except if we use the Dirichlet process as the prior over both component labels and parameter values we can describe an arbitrary, potentially infinite number of components.

We can write the generative process for the standard DPMM as

$$G|H, \alpha_0 \sim \mathrm{DP}(H, \alpha_0) \tag{2.49}$$

$$\theta_i|G \sim G \qquad i = 1, 2, \ldots, N \tag{2.50}$$

$$y_i|\theta_i \sim f(\theta_i) \qquad i = 1, 2, \ldots, N \tag{2.51}$$

where f is a class of observation distributions parameterized by θ. The graphical model for the DPMM is given in Figure 2-5(a). For concreteness, we may consider f to be the class of scalar, unit-variance normal distributions with a mean parameter, i.e. $f(\theta_i) = \mathcal{N}(\theta_i, 1)$. The measure H could then be chosen to be the conjugate prior, also a normal distribution, with hyperparameters $\lambda = (\mu_0, \sigma_0^2)$. Possible samples from this setting are sketched in Figure 2-5(b).

We may also write the DPMM generative process in the stick breaking form, keeping track of the label random variables $\{z_k\}$:

$$\begin{aligned} \beta|\alpha_0 &\sim \mathrm{GEM}(\alpha_0) && && (2.52)\\ \theta_k|H &\sim H && k = 1, 2, \ldots && (2.53)\\ z_i|\beta &\sim \beta && i = 1, 2, \ldots, N && (2.54)\\ y_i|z_i, \{\theta_k\} &\sim f(\theta_{z_i}). && i = 1, 2, \ldots, N && (2.55) \end{aligned}$$

A graphical model is given in Figure 2-6.

To perform posterior inference in the model given a set of observations $\{y_i\}_{i=1}^N$, we are most interested in conditionally sampling the label sequence $\{z_i\}$. If we choose our observation distribution f and the prior over its parameters H to be a conjugate pair, we can generally represent the posterior of $\{\theta_k\}_{k=1}^K|\{y_i\}_{i=1}^N, \{z_i\}_{i=1}^N, H$ in closed form, where K counts the number of unique labels in $\{z_i\}$ (i.e., the number of components present in our model for a fixed $\{z_i\}$). Hence, our primary goal is to be able to re-sample $\{z_i\}|\{y_i\}, H, \alpha_0$, marginalizing out the $\{\theta_k\}$ parameters.

We can create a Gibbs sampler to draw such samples by following the Chinese Restaurant Process. We iteratively draw $z_i|\{z_{\setminus i}\}, \{y_i\}, H, \alpha_0$, where $\{z_{\setminus i}\}$ denotes all other labels, i.e. $\{z_j : j \neq i\}$. To re-sample the ith label, we exploit the exchangeability of the process and consider z_i to be the last customer to enter the restaurant. We then draw its label according to

$$p(z_i = k) \propto \begin{cases} N_k \hat{f}(y_i|\{y_j : z_j = k\}) & k = 1, 2, \ldots, K \\ \alpha_0 & k = K + 1 \end{cases} \qquad (2.56)$$

where K counts the number of unique labels in $\{z_{\setminus i}\}$ and $\hat{f}(y_i|\{y_j : z_j = k\})$ represents the predictive likelihood of y_i given the other observation values with label k, integrating out the table's parameter θ_k. This process both instantiates and deletes mixture components ("tables") and allows us to draw posterior samples of $\{z_i\}$.

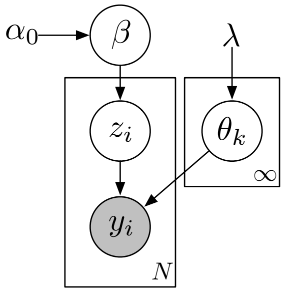


Figure 2-6: Alternative graphical model for the DPMM, corresponding to the stick breaking parameterization. The observation nodes are again shaded.

2.4 The Hierarchical Dirichlet Process

The Hierarchical Dirichlet Process (HDP) is a hierarchical extension of the Dirichlet Process which constructs a set of dependent DPs. Specifically, the dependent DPs share atom locations and have similar, but not identical, weights on their corresponding atoms. As described in this section, such a set of Dirichlet Processes allows us to build a Bayesian nonparametric extension of the Hidden Markov Model with the same desirable model-order inference properties as seen in the Dirichlet Process Mixture Model.

2.4.1 Defining the Hierarchical Dirichlet Process

Definition Let H be a probability measure over a space $(\Omega, \mathcal{B})$ and α_0 and γ be positive real number. We say the set of probability measure $\{G_j\}_{j=1}^J$ are distributed according to the Hierarchical Dirichlet Process if

$$G_0 | H, \alpha_0 \sim \mathrm{DP}(H, \alpha_0) \tag{2.57}$$

$$G_j | G_0, \gamma \sim \mathrm{DP}(G_0, \gamma) \qquad j = 1, 2, \ldots, J \tag{2.58}$$

$$\tag{2.59}$$

for some positive integer J which is fixed a priori.

Note that by Property 1 of the Dirichlet Process, we have $\mathbb{E}[G_j(A)|G_0] = G_0(A)$ for $j = 1, 2, \ldots, J$ for all $A \in \mathcal{B}$. Hence, G_0 can be interpreted as the “average” distribution shared by the dependent DPs. The γ parameter is an additional concentration parameter, which controls the dispersion of the dependent DPs around their mean. Furthermore, note that since G_0 is discrete with probability 1, G_j is discrete with the same set of atoms.

There is also a stick breaking representation of the Hierarchical Dirichlet Process:

Definition (Stick Breaking) We say $\{G_j\}_{j=1}^J$ are distributed according to a Hierarchical Dirichlet Process with base measure H and positive real concentration parameters α_0 and γ if

$$
\begin{align}
\beta|\alpha_0 &\sim \mathrm{GEM}(\alpha_0) \tag{2.60}\\
\bar{\theta}_k|H &\sim H & k &= 1, 2, \ldots \tag{2.61}\\
G_0 &\triangleq \sum_{k=1}^{\infty} \beta_k \delta_{\bar{\theta}_k} \tag{2.62}\\
& \tag{2.63}\\
\tilde{\pi}_j|\gamma &\sim \mathrm{GEM}(\gamma) & j &= 1, 2, \ldots, J \tag{2.64}\\
\theta_{ji}|G_0 &\sim G_0 & i &= 1, 2, \ldots, N_j \tag{2.65}\\
G_j &\triangleq \sum_{i=1}^{\infty} \tilde{\pi}_{ji} \delta_{\theta_{ij}}. \tag{2.66}
\end{align}
$$

Here, we have used the notation $\bar{\theta}_k$ to identify the distinct atom locations; the θ_{ji} are non-distinct with positive probability, since they are drawn from a discrete measure. Similarly, note that we use $\tilde{\pi}_{ji}$ to note that these are the weights corresponding to the non-distinct atom locations θ_{ji}; the total mass at that location may be a sum of several $\tilde{\pi}_{ji}$.

2.4.2 The Hierarchical Dirichlet Process Mixture Model

In this section, we briefly describe a mixture model based on the Hierarchical Dirichlet Process. The Hierarchical Dirichlet Process Mixture Model (HDPMM) expresses a set of J separate mixture models which share properties according to the HDP.

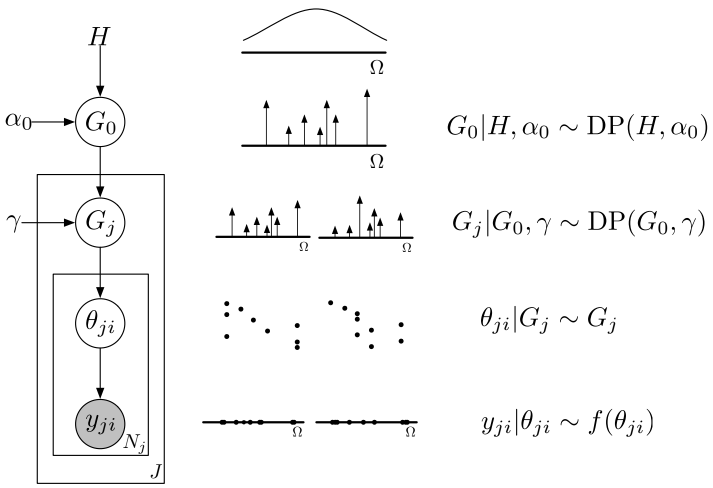


Figure 2-7: The Hierarchical Dirichlet Process Mixture Model: (a) graphical model; (b) depiction of sampled objects; (c) generative process.

Specifically, each mixture model is parameterized by one of the dependent Dirichlet Processes, and so the models share mixture components and are encouraged to have similar weights.

One parameterization of the HDPMM is summarized in Figure 2-7. However, the parameterization that is most tractable for inference follows the stick breaking construction but eliminates the redundancy in the parameters:

$$\begin{aligned}
\beta|\alpha_0 &\sim \mathrm{DP}(\beta, \alpha_0) && && (2.67)\\
\pi_j|\beta, \gamma &\sim \mathrm{DP}(\beta, \gamma) && j = 1, 2, \ldots, J && (2.68)\\
z_{ji}|\pi_j &\sim \pi_j && i = 1, 2, \ldots, N_j && (2.69)\\
\theta_k|H &\sim H && k = 1, 2, \ldots && (2.70)\\
y_{ji}|\{\theta_k\}, z_{ji} &\sim f(\theta_{z_{ji}}) && && (2.71)
\end{aligned}$$

This third parameterization of the HDP is equivalent [3] to the other parameterizations, and recovers the distinct values[3] of the $\{\theta_k\}$ parameters while providing a label sequence $\{z_{ji}\}$ that is convenient for resampling. A graphical model for this parameterization of the mixture model is given in Figure 2-8.

[3] Here we have assumed that H is such that two independent draws are distinct almost surely. This assumption is standard and allows for this convenient reparameterization.

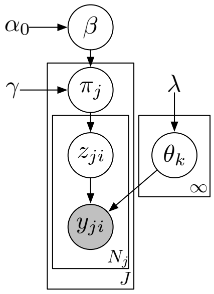


Figure 2-8: A graphical model for the stick breaking parameterization of the Hierarchical Dirichlet Process Mixture Model with unique atom locations.

To perform posterior inference in this mixture model, there are several sampling schemes based on a generalization of the Chinese Restaurant Process, the Chinese Restaurant Franchise. A thorough discussion of these schemes can be found in [14].

2.5 The Hierarchical Dirichlet Process Hidden Markov Model

The HDP-HMM [14] provides a natural Bayesian nonparametric treatment of the classical Hidden Markov Model approach to sequential statistical modeling. 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. Thus the HDP-HMM subsumes the usual model selection problem, replacing other techniques for choosing a fixed number of HMM states such as cross-validation procedures, which can be computationally expensive and restrictive. Furthermore, the HDP-HMM inherits many of the desirable properties of the HDP prior, especially the ability to encourage model parsimony while allowing complexity to grow with the number of observations. We provide a brief overview of the HDP-HMM model and relevant inference techniques, which we extend to develop the HDP-HSMM.

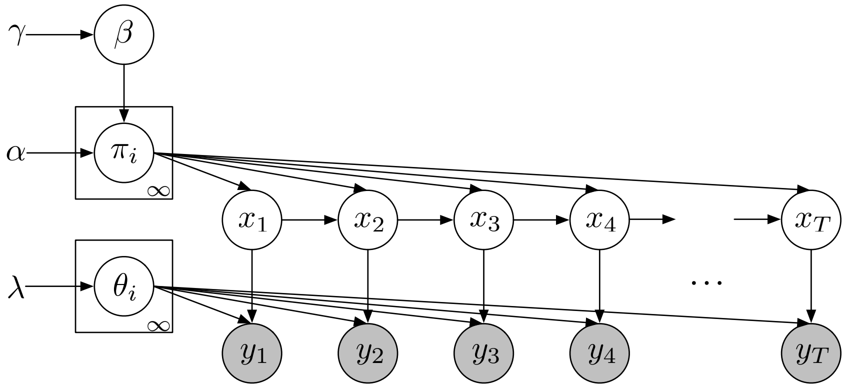


Figure 2-9: Graphical model for the HDP-HMM.

The generative HDP-HMM model (Figure 2-9) can be summarized as:

$$\beta|\gamma \sim \mathrm{GEM}(\gamma) \tag{2.72}$$

$$\pi_j|\beta, \alpha \sim \mathrm{DP}(\alpha, \beta) \qquad j = 1, 2, \ldots \tag{2.73}$$

$$\theta_j|H, \lambda \sim H(\lambda) \qquad j = 1, 2, \ldots \tag{2.74}$$

$$x_t|\{\pi_j\}_{j=1}^{\infty}, x_{t-1} \sim \pi_{x_{t-1}} \qquad t = 1, \ldots, T \tag{2.75}$$

$$y_t|\{\theta_j\}_{j=1}^{\infty}, x_t \sim f(\theta_{x_t}) \qquad t = 1, \ldots, T \tag{2.76}$$

where GEM denotes a stick breaking process [11]. We define $\pi_{x_0} \triangleq \pi_0$ to be a separate distribution.

The variable sequence (x_t) represents the hidden state sequence, and (y_t) represents the observation sequence drawn from the observation distribution class f. The set of state-specific observation distribution parameters is represented by $\{\theta_j\}$, which are draws from the prior H parameterized by λ. The HDP plays the role of a prior over infinite transition matrices: each π_j is a DP draw and is interpreted as the transition distribution from state j, i.e. the jth row of the transition matrix. The π_j are linked by being DP draws parameterized by the same discrete measure β, thus $E[\pi_j] = \beta$ 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.

The Chinese Restaurant Franchise sampling methods provide us with effective approximate inference for the full infinite-dimensional HDP, but they have a particular weakness in the context of the HDP-HMM: each state transition must be re-sampled individually, and strong correlations within the state sequence significantly reduce mixing rates for such operations [3]. As a result, finite approximations to the HDP have been studied for the purpose of providing alternative approximate inference schemes. Of particular note is the popular weak limit approximation, used in [2], which has been shown to reduce mixing times for HDP-HMM inference while sacrificing little of the "tail" of the infinite transition matrix. In this thesis, we describe how the HDP-HSMM with geometric durations can provide an HDP-HMM sampling inference algorithm that maintains the "full" infinite-dimensional sampling process while mitigating the detrimental mixing effects due to the strong correlations in the state sequence, thus providing a novel alternative to existing HDP-HMM sampling methods.

Chapter 3

New Models and Inference Methods

In this chapter we develop new models and sampling inference methods that extend the Bayesian nonparametric approaches to sequential data modeling.

First, we develop a blocked Gibbs sampling scheme for finite Bayesian Hidden semi-Markov Models; Bayesian inference in such models has not been developed previously. We show that a naive application of HMM sampling techniques is not possible for the HSMM because the standard prior distributions are no longer conjugate, and we develop an auxiliary variable Gibbs sampler that effectively recovers conjugacy and provides very efficient, accurate inference. Our algorithm is of interest not only to provide Bayesian sampling inference for the finite HSMM, but also to serve as a sampler in the weak-limit approximation to the nonparametric extensions.

Next, we define the nonparametric Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM) and develop a Gibbs sampling algorithm based on the Chinese Restaurant Franchise sampling techniques used for posterior inference in the HDP-HMM. As in the finite case, issues of conjugacy require careful treatment, and we show how to employ latent history sampling [9] to provide clean and efficient Gibbs sampling updates. Finally, we describe a more efficient approximate sampling inference scheme for the HDP-HSMM based on a common finite approximation to the HDP, which connects the sampling inference techniques for finite HSMMs to the

Bayesian nonparametric theory.

The inference algorithms developed in this chapter not only provide for efficient inference in the HDP-HSMM and Bayesian HSMM, but also contribute a new procedure for inference in HDP-HMMs.

3.1 Sampling Inference in Finite Bayesian HSMMs

In this section, we develop a sampling algorithm to perform Bayesian inference in finite HSMMs. The existing literature on HSMMs deals primarily with Frequentist formulations, in which parameter learning is performed by applications of the Expectation Maximization algorithm [1]. Our sampling algorithm for finite HSMMs contributes a Bayesian alternative to existing methods.

3.1.1 Outline of Gibbs Sampler

To perform posterior inference in a finite Bayesian Hidden semi-Markov model (as defined in Section 2.2), we can construct a Gibbs sampler resembling the sampler described for finite HMMs in Section 2.1.2.

Our goal is to construct a particle representation of the posterior

$$p((x_t), \theta, \{\pi_i\}, \{\omega_i\} | (y_t), \alpha, \lambda) \tag{3.1}$$

by drawing samples from the distribution. This posterior is comparable to the posterior we sought in the Bayesian HMM formulation of Eq. 2.7, but note that in the HSMM case we include the duration distribution parameters, $\{\omega_i\}$. We can construct these samples by following a Gibbs sampling algorithm in which we iteratively sample from the distributions of the conditional random variables:

$$(x_t)|\theta, \{\pi_i\}, \{\omega_i\}, (y_t) \tag{3.2}$$

$$\{\pi_i\}|\alpha, (x_t) \tag{3.3}$$

$$\{\omega_i\}|(x_t), \eta \tag{3.4}$$

$$\theta|\lambda, (x_t), (y_t) \tag{3.5}$$

where η represents the hyperparameters for the priors over the duration parameters $\{\omega_i\}$.

Sampling θ or $\{\omega_i\}$ from their respective conditional distributions can be easily reduced to standard problems depending on the particular priors chosen, and further discussion for common cases can be found in [1]. However, sampling $(x_t)|\theta, \{\pi_i\}, (y_t)$ and $\{\pi_i\}|\alpha, (x_t)$ in a Hidden semi-Markov Model has not been previously developed. In the following sections, we develop (1) an algorithm for block-sampling the state sequence (x_t) from its conditional distribution by employing the HSMM message-passing scheme of Section 2.2 and (2) an auxiliary variable sampler to provide easily resampling of $\{\pi_i\}$ from its conditional distribution.

3.1.2 Blocked Conditional Sampling of (x_t) with Message Passing

To block sample $(x_t)|\theta, \{\pi_i\}, \{\omega_i\}, (y_t)$ in an HSMM we can extend the standard block state sampling scheme for an HMM, as described in Section 2.1.2. 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 backwards messages β and β^* described in Section 2.2, then we can easily draw a posterior sample for the first state according to:

$$\begin{aligned} p(x_1 = i|y_{1:T}) &\propto p(x_1 = i)p(y_{1:T}|x_1 = i, F_0 = 1) \qquad (3.6)\\ &= p(x_1 = i)\beta_0^*(i) \qquad (3.7)\end{aligned}$$

where we have used the assumption that the observation sequence begins on a segment boundary ($F_0 = 1$) and suppressed notation for conditioning on parameters. This first step is directly analogous to the first step of sampling (x_t) for an HMM.

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, $\bar{x}_1$, we can draw a sample of $D_1 | y_{1:T}, x_1 = \bar{x}_1$ (suppressing notation for conditioning on parameters), the posterior duration of the first state is:

$$p(D_1 = d | y_{1:T}, x_1 = \bar{x}_1, F_0 = 1) = \frac{\mathbb{P}(D_1 = d, y_{1:t} | x_1 = \bar{x}_1, F_0)}{p(y_{1:t} | x_1 = \bar{x}_1, F_0)} \tag{2}$$

$$= \frac{\mathbb{P}(D_1 = d | x_1 = \bar{x}_1, F_0) p(y_{1:d} | D_1 = d, Q_1, F_0) p(y_{d+1:t} | D_1 = d, x_1 = \bar{x}_1, F_0)}{p(y_{1:t} | x_1 = \bar{x}_1, F_0)} \tag{3.8}$$

$$= \frac{p(D_1 = d) p(y_{1:d} | D_1 = d, x_1 = \bar{x}_1, F_0 = 1) \beta_d(\bar{x}_1)}{\beta_0^*(\bar{x}_1)}. \tag{3.9}$$

We can repeat the process by then considering x_{D_1+1} to be our new initial state with initial distribution given by $p(x_{D_1+1} = i | x_1 = \bar{x}_1)$, analogous to the HMM case.

3.1.3 Conditional Sampling of $\{\pi_i\}$ with Auxiliary Variables

In the standard construction of an HSMM, as described in Section 2.2, self-transitions are ruled out. However, in the Bayesian setting, this restriction means that the Dirichlet distribution is not a conjugate prior for the transition parameters, $\{\pi_i\}$, as it is in the Bayesian HMM construction.

To observe the loss of conjugacy, note that we can summarize the relevant portion of the generative model as

$$\begin{aligned} \pi_j | \beta &\sim \mathrm{Dir}(\beta_1, \ldots, \alpha\beta_L) & j &= 1, \ldots, L \\ x_t | \{\pi_j\}, x_{t-1} &\sim \bar{\pi}_{x_{t-1}} & t &= 2, \ldots, T \end{aligned}$$

where $\bar{\pi}_j$ represents π_j with the jth component removed and renormalized appropriately, i.e.:

$$\bar{\pi}_{ji} \propto \pi_{ji}(1 - \delta_{ij})$$
$$\sum_{i=1}^{L} \bar{\pi}_{ji} = 1$$

with $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. The deterministic transformation from π_j to $\bar{\pi}_j$ eliminates self-transitions. Note that we have suppressed the observation parameter set, duration parameter set, and observation sequence sampling for simplicity.

Consider the distribution of $\pi_1|(x_t), \beta$:

$$\begin{aligned} p(\pi_1|(x_t), \beta) &\propto p(\pi_1|\beta)p((x_t)|\pi_1) \\ &\propto \pi_{11}^{\beta_1 - 1}\pi_{12}^{\beta_2 - 1} \cdots \pi_{1L}^{\beta_L - 1} \left(\frac{\pi_{12}}{1 - \pi_{11}}\right)^{n_{12}} \left(\frac{\pi_{13}}{1 - \pi_{11}}\right)^{n_{13}} \cdots \left(\frac{\pi_{1L}}{1 - \pi_{11}}\right)^{n_{1L}} \end{aligned}$$

where n_{ij} are the number of transitions from state i to state j in the state sequence (x_t). Essentially, because of the extra $\frac{1}{1-\pi_{11}}$ terms from the likelihood without self-transitions, we cannot reduce this expression to the Dirichlet form over the components of π_1.

However, we can introduce auxiliary variables to recover conjugacy. For notational convenience, we consider the simplified model:

$$\begin{aligned} &\pi|\beta \sim \text{Dir}(\beta) \\ &z_i|\bar{\pi} \sim \bar{\pi} \quad i = 1, \dots, n \\ &y_i|z_i \sim f(z_i) \quad i = 1, \dots, n \end{aligned}$$

where $\bar{\pi}$ is formed by removing the first component of π and re-normalizing. Here, the $\{z_i\}$ directly represent the multinomial transitions of the state sequence (x_t), and the $\{y_i\}$ embody the effect of the observation sequence (y_t). See the graphical model in Figure 3-1 for a depiction of the relationship between the variables.

We wish to draw samples from $\pi, \{z_i\}|\{y_i\}$ by iterating Gibbs sampling steps

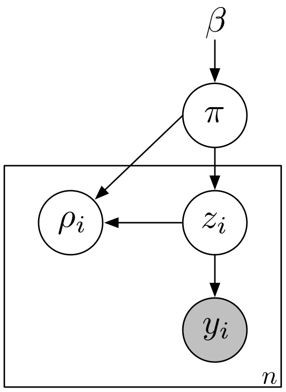


Figure 3-1: Simplified depiction of the relationship between the auxiliary variables and the rest of the model.

between drawing $\{z_i\}|\pi, \{y_i\}$ and $\pi|\{z_i\}$, but as before the latter step is difficult because we do not have a conjugate Dirichlet distribution over π:

$$p(\pi|\{z_i\}) \propto \pi_1^{\beta_1-1}\pi_2^{\beta_2-1}\cdots\pi_L^{\beta_L-1}\left(\frac{\pi_2}{1-\pi_1}\right)^{n_2}\left(\frac{\pi_3}{1-\pi_1}\right)^{n_3}\cdots\left(\frac{\pi_L}{1-\pi_1}\right)^{n_L}$$

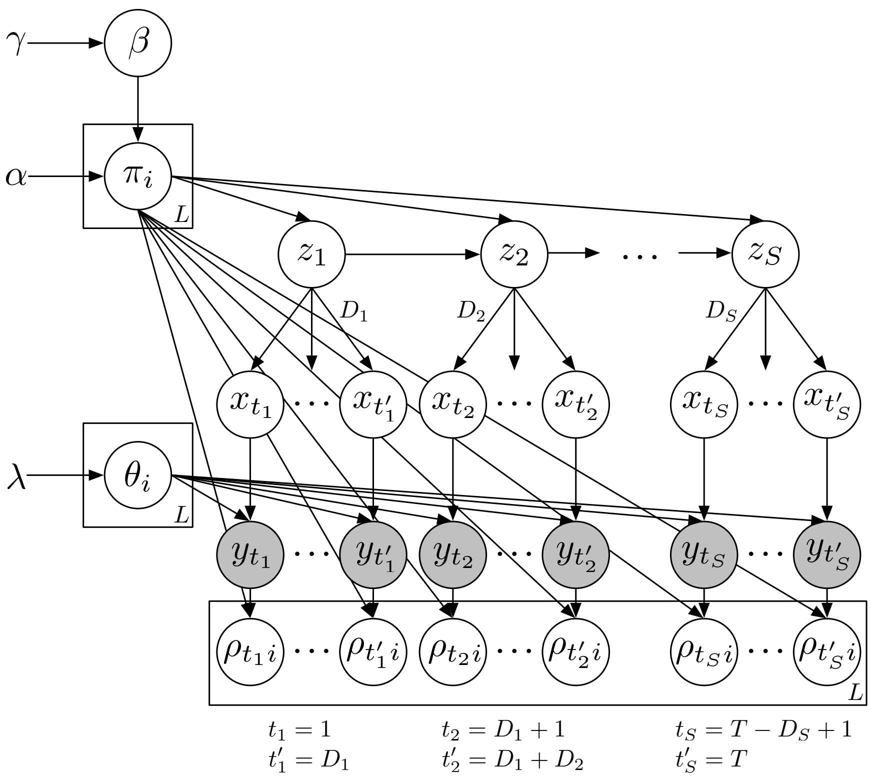
However, we can introduce the auxiliary variables $\{\rho_i\}_{i=1}^n$, where each ρ_i is independently drawn from a geometric distribution supported on $\{0, 1, \ldots\}$ with success parameter $1 - \pi_1$, i.e. $\rho_i \sim \text{Geo}(1 - \pi_1)$. Thus our posterior becomes:

$$\begin{aligned}
p(\pi|\{z_i\}, \{\rho_i\}) &\propto p(\pi)p(\{z_i\}|\pi)p(\{\rho_i\}|\{\pi_i\}) \\
&\propto \pi_1^{\beta_1-1}\pi_2^{\beta_2-1}\cdots\pi_L^{\beta_L-1}\left(\frac{\pi_2}{1-\pi_1}\right)^{n_2}\left(\frac{\pi_3}{1-\pi_1}\right)^{n_3}\cdots\left(\frac{\pi_L}{1-\pi_1}\right)^{n_L}\left(\prod_{i=1}^n \pi_1^{\rho_i}(1-\pi_1)\right) \\
&= \pi_1^{\beta_1+\sum_i \rho_i-1}\pi_2^{\beta_2+n_2-1}\cdots\pi_L^{\beta_L+n_L-1} \\
&\propto \text{Dir}(\beta_1 + \sum_i \rho_i, \beta_2 + n_2, \ldots, \beta_L + n_L)
\end{aligned}$$

and so, noting that $n = \sum_i n_i$, we recover conjugacy.

Intuitively, we are able to fill in the data to include self-transitions because before each transition is sampled, we must sample and reject $\text{Geo}(1 - \pi_1)$ self-transitions. Note that this procedure depends on the fact that, by construction, the $\{\rho_i\}$ are conditionally independent of the data, $\{y_i\}$, given $\{z_i\}$.

We can easily extend these auxiliary variables to the HSMM, and once we have



γ
β
α
π_i
L
z_1
z_2
$\cdots$
z_S
D_1
D_2
D_S
x_{t_1} $\cdots$ $x_{t'_1}$
x_{t_2} $\cdots$ $x_{t'_2}$
x_{t_S} $\cdots$ $x_{t'_S}$
λ
θ_i
L
y_{t_1} $\cdots$ $y_{t'_1}$
y_{t_2} $\cdots$ $y_{t'_2}$
y_{t_S} $\cdots$ $y_{t'_S}$
$\rho_{t_1 i}$ $\cdots$ $\rho_{t'_1 i}$
$\rho_{t_2 i}$ $\cdots$ $\rho_{t'_2 i}$
$\rho_{t_S i}$ $\cdots$ $\rho_{t'_S i}$
L
$t_1 = 1$
$t'_1 = D_1$
$t_2 = D_1 + 1$
$t'_2 = D_1 + D_2$
$t_S = T - D_S + 1$
$t'_S = T$

Figure 3-2: Graphical model for the weak-limit approximation including auxiliary variables.

completed the data with the auxiliary variables, we are once again in the conjugate setting and can use standard sampling methods. A graphical model for a finite HSMM including the auxiliary variables is shown in Figure 3-2. Note that each ρ_{ti} depends on the state label x_t. Also note that Figure 3-2 draws the parameters β and $\{\pi_i\}$ as coupled according to the weak limit approximation, which will be discussed in Section 3.2.3.

3.2 The Hierarchical Dirichlet Process Hidden Semi-Markov Model

In this section we introduce the Hierarchical Dirichlet Process Hidden semi-Markov Model (HDP-HSMM), the nonparametric extension of the finite HSMM, and develop efficient inference techniques.

First, we define the generative process of the HDP-HSMM, which augments the HDP-HMM generative process with general state duration distributions, just as the finite HSMM generative process augments that of the finite HMM. Next, we develop collapsed Gibbs inference algorithm for the HDP-HSMM, which can be viewed as an extension to the direct assignment Chinese Restaurant Franchise sampler of the HDP-HMM (Section 2.5). We must again consider the challenges posed by a loss of prior conjugacy, analogous to those described in Section 3.1.3 but this time in the setting where the infinite transition parameters of the DP are marginalized, in accord with the CRF. Finally, we describe an approximate, finite sampler for the HDP-HSMM based on the standard weak-limit approximation.

3.2.1 Model Definition

The generative process of the HDP-HSMM is similar to that of the HDP-HMM, with some extra work to include duration distributions:

$$\beta|\gamma \sim \mathrm{GEM}(\gamma) \tag{3.10}$$

$$\pi_j|\beta,\alpha \sim \mathrm{DP}(\alpha,\beta) \qquad j = 1, 2, \ldots \tag{3.11}$$

$$\theta_j|H,\lambda \sim H(\lambda) \qquad j = 1, 2, \ldots \tag{3.12}$$

$$\omega_j|\Omega \sim \Omega \qquad j = 1, 2, \ldots \tag{3.13}$$

$\tau := 0$, $s := 1$, while $\tau < T$ do:

$$z_s|\{\pi_j\}_{j=1}^{\infty}, z_{s-1} \sim \tilde{\pi}_{z_{s-1}} \tag{3.14}$$

$$D_s|\omega \sim D(\omega_{z_s}) \tag{3.15}$$

$$y_s = y_{\tau+1:\tau+D_s+1}|\{\theta_j\}_{j=1}^{\infty}, z_s, D_s \overset{\text{iid}}{\sim} F(\theta_{z_s}) \tag{3.16}$$

$$\tau := \tau + D_s \tag{3.17}$$

$$s := s + 1 \tag{3.18}$$

where we have used (z_s) as a *super-state sequence* indexed by s and $\{\omega_j\}_{j=1}^{\infty}$ to represent the parameters for the duration distributions of each of the states, with D

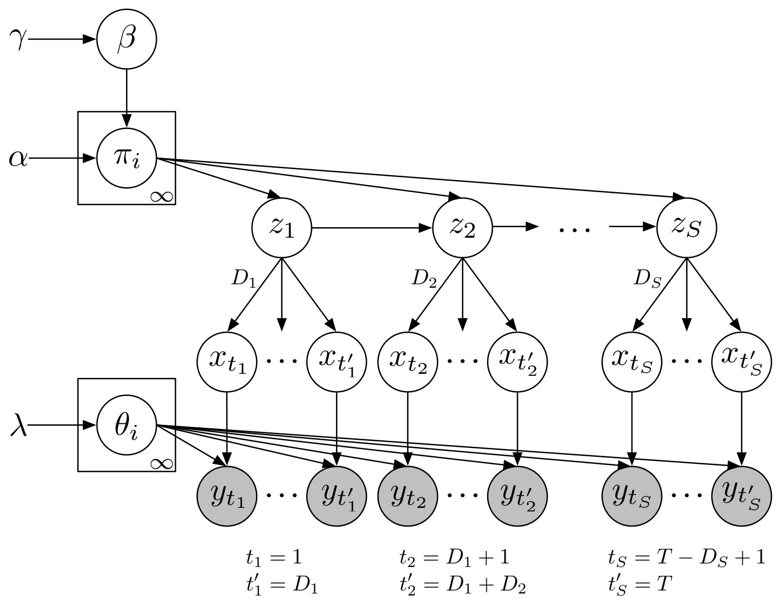


Figure 3-3: A graphical model for the HDP-HSMM in which the number of nodes is random. We will call $(z_s)_{s=1}^S$ the *super-state sequence*, $(x_t)_{t=1}^T$ the *label sequence*, and $(y_t)_{t=1}^T$ the *observation sequence*. The (D_s) variables are the random durations (or segment lengths), and here they control the "fan-out" from their respective super-states.

representing the class of duration distributions. At the end of the process, we censor the observations to have length T exactly, cutting off any excess observations if necessary, so as to generate $y_{1:T}$. It is also convenient to refer to x_t as the *label* of observation y_t; it identifies to which super-state the observation belongs. We refer to the sequence (x_t) as the *label sequence*, and note that the label sequence contains the same information as the pair $((z_s), (D_s))$, the super-state and duration sequences. Note also that we have previously referred to (x_t) as a state sequence, while we now distinguish it as a label sequence and refer to (z_s) as the super-state sequence. A graphical model is given in Figure 3-3.

Note, most importantly, that we draw $z_s | \{\pi_j\}, z_{s-1}$ from $\tilde{\pi}_{z_{s-1}}$, which we use to denote the conditional measure constructed from π_j by removing the atom corresponding to z_{s-1} and re-normalizing appropriately. This part of the construction, which is valid with probability 1, effectively rules out self-transitions.

If D, the duration distribution class, is geometric, we effectively recover the HDP-HMM (just as we would recover a standard HMM from an HSMM with geometric duration distributions) but the resulting inference procedure remains distinct from the HDP-HMM. Thus the HDP-HSMM sampling inference methods described in the next section provide a novel alternative to existing HDP-HMM samplers with some potentially significant advantages.

3.2.2 Sampling Inference via Direct Assignments

In this section, we develop an HDP-HSMM direct assignment sampler based on the direct assignment Chinese Restaurant Franchise sampler for the HDP-HMM. This Gibbs sampling algorithm provides a method for inference in the HDP-HSMM while marginalizing over the infinite-dimensional Hierarchical Dirichlet Process prior.

To create a direct assignment sampler based on the HDP-HMM direct assignment sampler of [14], we can leverage the viewpoint of an HSMM as an HMM on super-state segments and split the sampling update into two steps. First, conditioning on a segmentation (which defines super-state boundaries but not labels), we can view blocks of observations as atomic with a single predictive likelihood score for the entire block. We can then run an HDP-HMM direct assignment sampler on the super-state chain with the caveat that we have outlawed self-transitions. Second, given a super-state sequence we can efficiently re-sample the segmentation boundaries.

To deal with the outlawed self-transition caveat, we must first note that it eliminates exchangeability and hence the Dirichlet Processes's (DP's) convenient posterior properties. The result that the posterior for a Dirichlet Process is also a DP, parameterized by the base measure combined with atoms at observation values, does not apply when we only observe a subset of the draws.

However, we can rule out self-transitions in the super-state sequence while maintaining a complete sample of transitions by running a rejection sampler. The rejections serve a similar purpose to the auxiliary variables we introduced for purposes of inference in the finite HSMM in Section 3.1. However, we do not construct the same auxiliary variables in this sampler because we do not explicitly represent transition probabilities; instead, we effectively marginalize over them with the Chinese Restaurant Franchise. To sample the auxiliary self-transition counts, we sample super-state transitions without any constraints, and we reject any samples that result in self-transitions while counting the number of such rejections for each state. These "dummy" self-transitions, which are not represented in the super-state sequence and hence are independent of the observations, allow us to sample posterior super-state transitions according to the standard HDP direct assignment sampler. This technique is an instance of *latent history* sampling, as described in [9].

Hence, to perform sampling inference in the HDP-HSMM we iterate between two steps. In the first step, we fix the segmentation and re-sample the super-state labels (z_s) according to an HDP-HMM direct assignment sampler. That is, we fix the duration times (D_s) and view the segments of observations associated to each super-state $(y_{t_s:t'_s})$ as atomic, scoring the predictive likelihood for each segment according to the product of the predictive likelihoods of all its observations. To deal with the caveat that we do not allow self-transitions, we also keep counts of rejected self-transitions. We refer to this step as "the HDP-HMM sampling step."

In the second step, we fix the super-state sequence (z_s) and wish to re-sample the segmentation, i.e. the sequence (D_s). We can equivalently re-sample the label sequence (x_t) to get the durations, so long as we enforce that the sequence of labels matches the same super-state sequence as conditioned on in (z_s). We refer to this step as "the segment sampling step" or "the label sampling step."

We can re-sample the label sequence while matching the super-state sequence by constructing a finite HSMM and using the messages-backward, sample-forward posterior sampling technique described in Section 3.1. First, we sample the posterior observation and duration parameters for each unique super-state: $\{\theta_i\}_{i=1}^S$ and $\{\omega_i\}_{i=1}^S$, respectively. Then, we construct an S-state finite HSMM on states $(x_t)_1^T$ with a transition matrix with 1s along its first superdiagonal and 0s elsewhere:

$$(A^*)_{ij} = \begin{cases} 1 & j = i+1 \\ 0 & \text{otherwise} \end{cases} \qquad \text{for } i, j = 1, 2, \ldots, S.$$

We also identify the observation distribution of the finite HSMM's state $s \in \{1, \ldots, S\}$

to be $f(\theta_{z_s})$ and the duration distribution of state s to be $D(\omega_{z_s})$. This construction, along with sampling x_1 deterministically as z_1, forces the sampled label sequence to follow the super-state sequence.

However, the construction is not quite complete because the label sequence may not match the T observations exactly to the S instantiated super-states; instead, we may either not assign observations to all the S super-states, or require more than S labels (or segments) in our label sequence.

For the first case, if we sample a label sequence in which we do not use all the super-states, i.e. the label sequence ends with $x_T = S' < S$, we simply consider the unused states to have no observations assigned to them, and thus we eliminate their explicit representation in the next iteration of the HDP-HMM sampling step. The states without observations assigned during the label sampling step are merged with the "new table" event for the HDP-HMM sampling step, just as in the standard CRF procedure when an instantiated table loses its last customer.

For the second case, we must provide for sampling a label sequence in which we use more than S labels, where S is the number of super-states with observations assigned. To allow sampling of a label sequence with more than S labels, our finite HSMM requires more than S states. Indeed, we must extend the finite HSMM described previously to a representation with T states to be able to encode a label sequence with up to T different labels (or, equivalently, T different segments). Only the first $S \leq T$ super-states have observations assigned to them, and so the remaining $T - S$ states have observation and duration parameters instantiated from their respective priors. We similarly revise our definition of the deterministic transition matrix A^*:

$$(A^*)_{ij} = \begin{cases} 1 & j = i + 1 \\ 0 & \text{otherwise} \end{cases} \quad \text{for } i, j = 1, 2, \ldots, T.$$

Both of the above cases provide necessary behavior: they ensure the sampler can increase and decrease the total number of segments, and hence it can mix from, e.g., a one-segment initialization to a many-segment sample.The two sampling steps are summarized in Figure 3-4.

It is interesting to consider how this sampling algorithm differs from the standard HDP-HMM procedure when geometric duration distributions are used. From a generative standpoint the model classes are identical, but in the HDP-HSMM Gibbs sampling algorithm the CRF steps re-sample super-states at each step, which corresponds to moving an entire block of observation labels. The CRF is slow to mix for the HDP-HMM exactly because adjacent observation labels are highly correlated with one another, and hence sampling new values for adjacent labels one-by-one requires many proposals and rejections. Our HDP-HSMM Gibbs sampling algorithm mitigates this effect by moving entire blocks of labels in single moves, thus effectively achieving efficient block-move proposals while remaining in the simple Gibbs sampling framework. Thus the HDP-HSMM sampling method can be useful not only for the case of non-geometric duration distributions, but also as an HDP-HMM sampler to avoid the usual mixing issues.

3.2.3 Sampling Inference with a Weak Limit Approximation

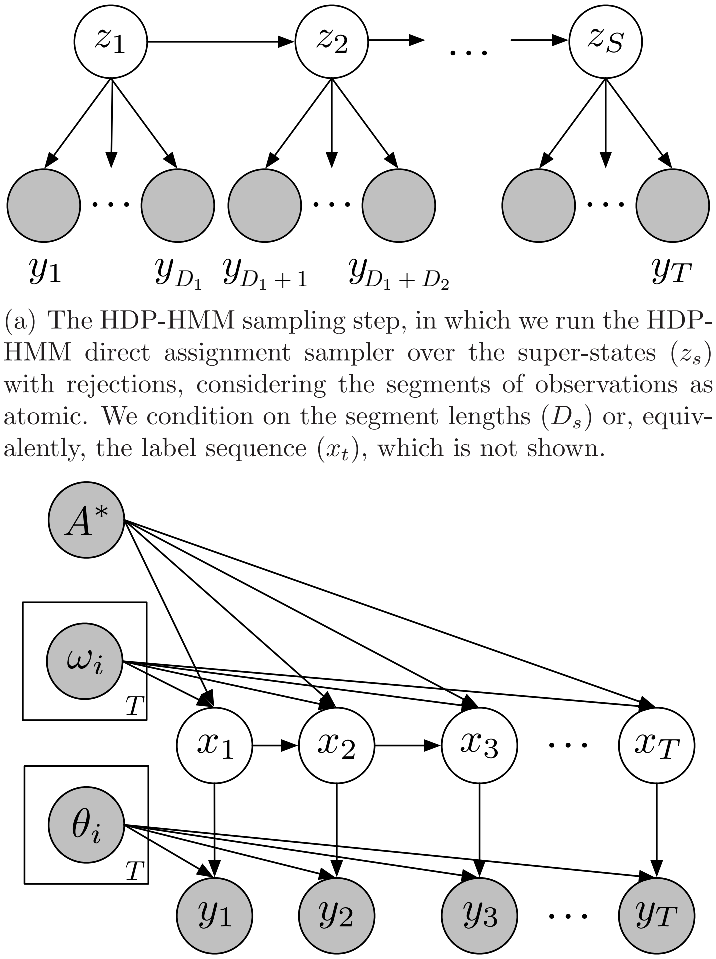
The weak-limit sampler for an HDP-HMM [2] constructs a finite approximation to the HDP transitions prior with finite L-dimensional Dirichlet distributions, motivated by the fact that the infinite limit of such a construction converges in distribution to a true HDP:

$$\beta|\gamma \sim \mathrm{Dir}(\gamma/L, \ldots, \gamma/L) \tag{3.19}$$

$$\pi_j|\alpha, \beta \sim \mathrm{Dir}(\alpha\beta_1, \ldots, \alpha\beta_L) \qquad j = 1, \ldots, L \tag{3.20}$$

where we again interpret π_j as the transition distribution for state j and β as the distribution which ties state distributions together and encourages shared sparsity. Practically, the weak limit approximation enables the instantiation of the transition matrix in a finite form, and thus allows block sampling of the entire label sequence at once, resulting in greatly accelerated mixing.

We can employ the same technique to create a finite HSMM that approximates



(a) The HDP-HMM sampling step, in which we run the HDP-HMM direct assignment sampler over the super-states (z_s) with rejections, considering the segments of observations as atomic. We condition on the segment lengths (D_s) or, equivalently, the label sequence (x_t), which is not shown.

(b) The segment sampling step, where A^*, $\{\omega_i\}$, and $\{\theta_i\}$ encode the requirement that the label sequence (x_t) follows the conditioned super-state sequence (z_s), which is not shown. Note that $(x_t)|A^*, \{\omega_i\}$ forms a semi-Markov chain, though it is (inaccurately) drawn as a Markov chain for simplicity.

Figure 3-4: An illustration of the two sampling steps in the HDP-HSMM direct assignment sampler.

the HDP-HSMM in the weak-limit sense, and hence employ the inference algorithm for finite HSMMs described in Section 3.1. A graphical model for a weak-limit approximate model is given in Figure 3-2. This approximation technique results in much more efficient inference, and hence it is the technique we employ for the experiments in the sequel.

Chapter 4

Experiments

In this chapter, we apply our HDP-HSMM weak-limit sampling algorithm to both synthetic and real data. These experiments demonstrate the utility of the HDP-HSMM and the inference methods developed in this thesis, particularly compared to the standard HDP-HMM.

First, we evaluate HDP-HSMM inference 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 well. Furthermore, we apply HDP-HSMM inference 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 compare the HDP-HSMM with the HDP-HMM on a problem of learning the patterns in Morse Code from an audio recording of the alphabet. This experiment provides a straightforward example of a case in which the HDP-HMM is unable to effectively model the duration statistics of data and hence unable to learn the appropriate state description, while the HDP-HSMM exploits duration information to learn the correct states.

Finally, we apply HDP-HSMM inference to a speech-processing problem using a standard dataset. This experiment shows the real-world effectiveness of the HDP-

HSMM and highlights the mixing-time gains that our HDP-HSMM inference algorithm can provide.

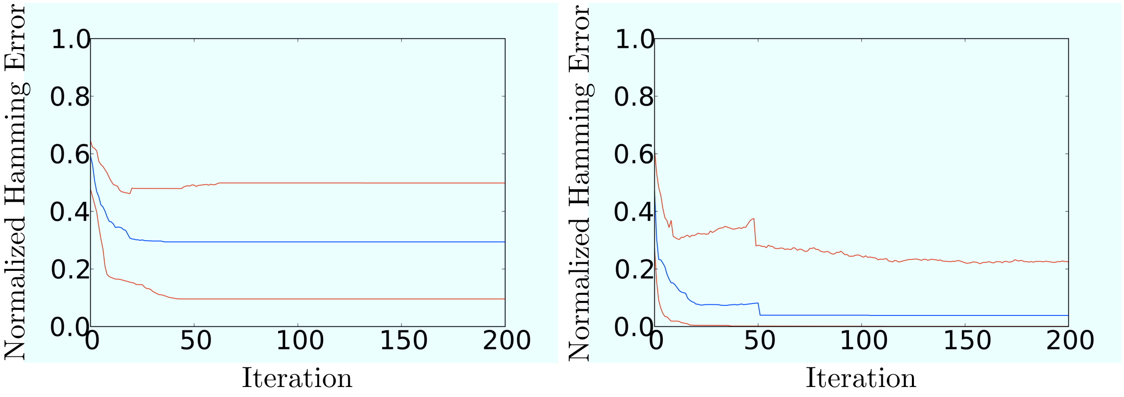
4.1 Synthetic Data

We evaluated the HDP-HSMM model and inference techniques by generating observations from both HSMMs and HMMs and comparing performance to the HDP-HMM. The models learn many parameters including observation, duration, and transition parameters for each state. We generally present the normalized Hamming error of the sampled state sequences as a summary metric, since it involves all learned parameters (e.g., if parameters are learned poorly, the inferred state sequence performance will suffer). In these plots, the blue line indicates the median error across 25 independent Gibbs sampling runs, while the red lines indicate 10th and 90th percentile errors.

Figure 4-1 summarizes the results of applying both an HDP-HSMM and an HDP-HMM to data generated from an HSMM with four states and Poisson durations. The observations for each state are mixtures of 2-dimensional Gaussians with significant overlap, with parameters for each state sampled i.i.d. from a Normal Inverse-Wishart (NIW) prior. In the 25 Gibbs sampling runs for each model, we applied 5 chains to each of 5 generated observation sequences. All priors were selected to be non-informative.

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 capture the correct temporal model 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. Figure 4-2 shows the number of states inferred by each model across the 25 runs.

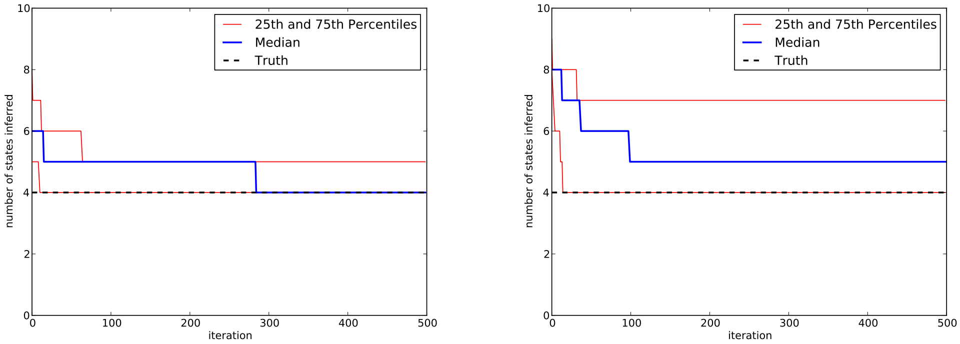
By setting the class of duration distributions to be a strict superclass of the geometric distribution, we can allow an HDP-HSMM model to learn an HMM from data when appropriate. One such distribution class is the class of negative binomial



(a) HDP-HMM

(b) HDP-HSMM

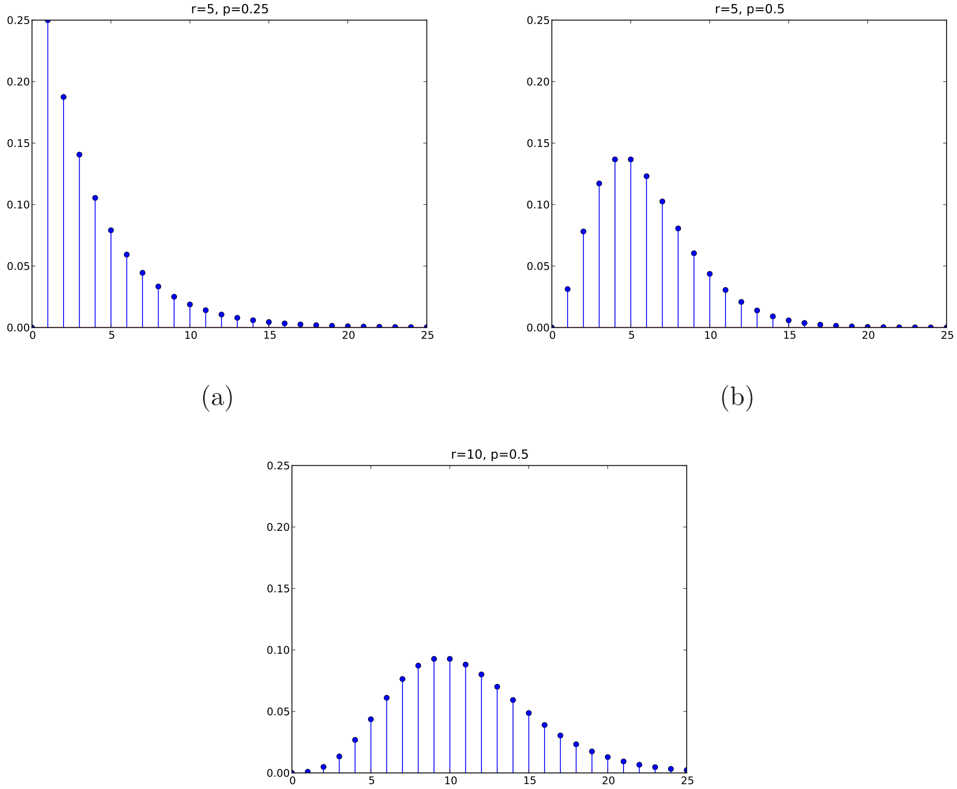
Figure 4-1: State-sequence Hamming error of the HDP-HMM and Poisson-HDP-HSMM applied to data from a Poisson-HSMM.



(a) HDP-HSMM

(b) HDP-HMM

Figure 4-2: Number of states inferred by the HDP-HMM and Poisson-HDP-HSMM applied to data from a four-state Poisson-HSMM.



(a)

(b)

(c)

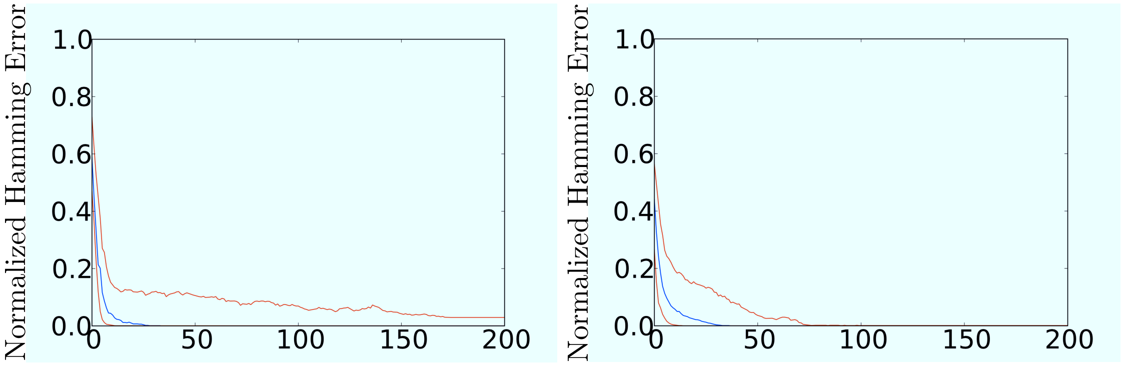
Figure 4-3: Plots of the Negative Binomial PMF for three values of the parameter pair (r, p).

distributions, denoted $\text{NegBin}(r, p)$, the discrete analog of the Gamma distribution, which covers the class of geometric distributions when $r = 1$. The probability mass function (PMF) for the Negative Binomial is given by

$$p(k|r, p) = \binom{k + r - 1}{r - 1} (1 - p)^r p^k \quad k = 0, 1, 2, \ldots \tag{4.1}$$

Plots of the PMF for various choices of the parameters r and p are given in Figure 4-3. 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 4-4 shows a negative binomial HDP-HSMM learning an HMM model from data generated from an HMM with four states. The observation distribution for each



(a) HDP-HMM

(b) HDP-HSMM

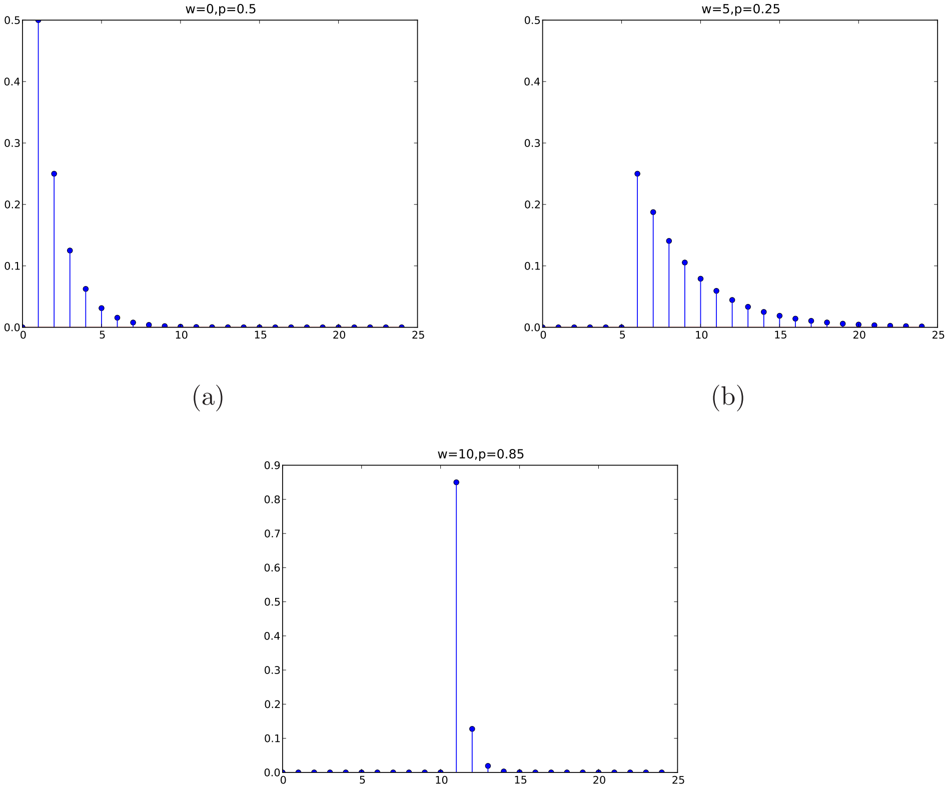
Figure 4-4: The HDP-HSMM and HDP-HMM applied to data from an HMM.

state is a 10-dimensional Gaussian, again with parameters sampled i.i.d. from a NIW prior. 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 negative binomial HDP-HSMM compared to the HDP-HMM on this data. The lower 90th-percentile error for the HDP-HSMM is attributed to the fact that our HDP-HSMM inference scheme resamples states in segment blocks and thus is less likely to explore newly instantiated states. This experiment demonstrates that with the appropriate choice of duration distribution the HDP-HSMM can effectively learn an HMM model when appropriate.

4.2 Learning Morse Code

As an example of duration information disambiguating states, we also applied both an HDP-HSMM and an HDP-HMM to spectrogram data from audio of the Morse code alphabet (see Figure 4-6). The data can clearly be partitioned into "tone" and "silence" clusters without inspecting any temporal structure, but only by incorporating duration information can we disambiguate the "short tone" and "long tone" states and thus correctly learn the state representation of Morse code.

In the HDP-HSMM we employ a delayed geometric duration distribution, in which



(a)

(b)

(c)

Figure 4-5: Plots of the delayed geometric PMF for three values of the parameter pair (w, p).

a state's duration is chosen by first waiting some w samples and then sampling a geometric. Both the wait w and geometric parameter p are learned from data, with a uniform prior over the set $\{0, 1, \ldots, 20\}$ for w and a $\text{Beta}(1, 1)$ uniform prior over p. This duration distribution class is also a superset of the class of geometric distributions, since the wait parameter w can be learned to be 0. Plots of the PMF for various choices of the parameters w and p are shown in Figure 4-5

We applied both the HDP-HSMM and HDP-HMM to the spectrogram data and found that both quickly concentrate at single explanations: the HDP-HMM finds only two states while the HDP-HSMM correctly disambiguates three, shown in Figure 4-7. The two "tone" states learned by the HDP-HSMM have w parameters that closely capture the near-deterministic pulse widths, with p learned to be near 1. The "silence" segments are better explained as one state with more variation in its duration

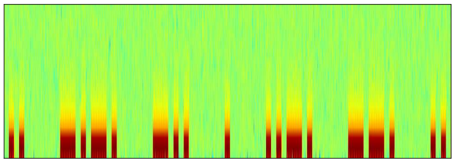
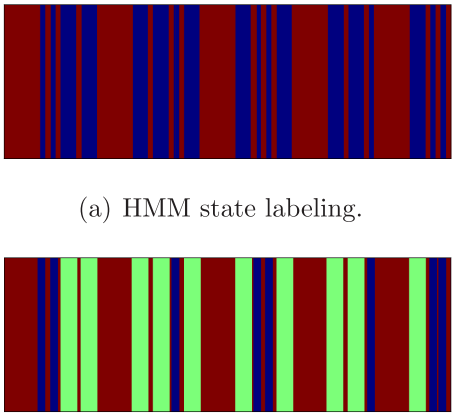


Figure 4-6: A spectrogram segment of Morse code audio.



(a) HMM state labeling.

(b) HSMM state labeling.

Figure 4-7: Each model applied to Morse code data.

statistics. Hence, the HDP-HSMM correctly uncovers the Morse Code alphabet as a natural explanation for the statistics of the audio data.

On the other hand, the HDP-HMM only learns "silence" and "tone" states; it is unable to separate the two types of tone states because they are only disambiguated by duration information. The HDP-HMM is constrained to geometric state durations, and since the geometric PMF is a strictly decreasing function over the support, any state that places significant probability on the long-tone duration places even higher probability on the short-tone duration, and so the two cannot be separated. Hence the HDP-HMM's inability to identify the Morse Code dynamics is a direct result of its strict Markovian restriction to geometric durations. Incorporating a duration distribution class that is able to learn both geometric and non-geometric durations allows us to learn a much more desirable model for the data.

4.3 Speaker Diarization

We also applied our model to a *speaker diarization*, or who-spoke-when, problem. Given a single, un-labeled audio recording of an unknown number of people speaking in a meeting, the task is to identify the number of speakers and segment the audio according to when each participant speaks. This problem is a natural fit for our Bayesian nonparametric HDP-HSMM because we wish to infer the number of speakers (state cardinality), and using non-geometric duration distributions not only allows us to rule out undesirably short speech segments but also provides accelerated mixing.

The NIST Rich Transcriptions Database is a standard dataset for the speaker diarization problem. It consists of audio recordings for each of 21 meetings with various numbers of participants. In working with this dataset, our focus is to demonstrate how the differences in the HDP-HSMM sampling algorithm manifest themselves on real data; state-of-the-art performance on this dataset has already been demonstrated by the Sticky HDP-HMM [2].

We first preprocessed the audio data into Mel Frequency Cepstral Coefficients (MFCCs) [15], the standard real-valued feature vector for the speaker diarization problem. We computed the largest 19 MFCCs over 30ms windows spaced every 10ms as our feature vectors, and reduced the dimensionality from 19 to 4 by projecting onto the first four principle components. We used mixtures of multivariate Gaussians as observation distributions, and we placed a Gaussian prior on the mean parameter and independent (non-conjugate) Inverse-Wishart prior on the covariance. The prior hyperparameters were set according to aggregate empirical statistics. We also smoothed and subsampled the data so as to make each discrete state correspond to 100ms of real time, resulting in observation sequences of length approximately 8000–10000. For duration distributions, we chose to again employ the delayed geometric distribution with the prior on each state's wait parameter as uniform over $\{40, 41, \dots, 60\}$. In this way we not only impose a minimum duration to avoid rapid state switching or learning in-speaker dynamics, but also force the state sampler to make minimum "block" moves of nontrivial size so as to speed mixing.

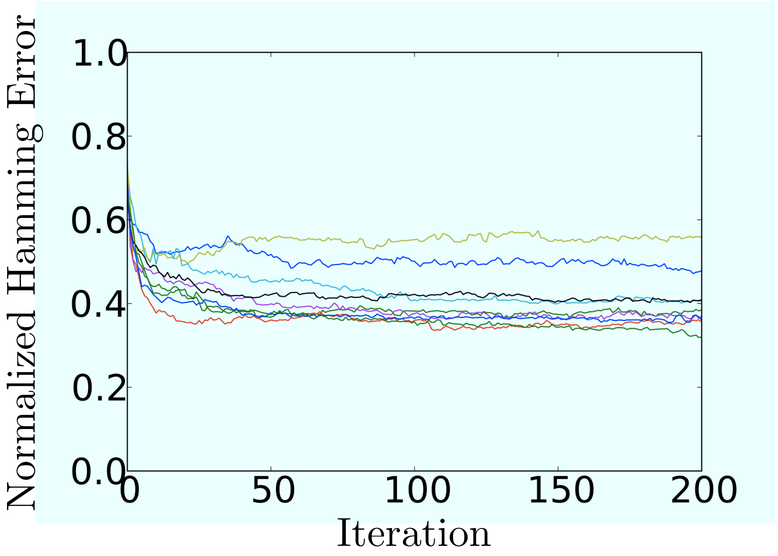


Figure 4-8: Relatively fast mixing of an HDP-HSMM sampler. Compare to Figure 3.19(b) of [3].

Our observation setup closely follows that of [2], but an important distinction is that each discrete state of [2] corresponds to 500ms of real time, while each discrete state in our setup corresponds to 100ms of real time. The 500ms time scale allows durations to better fit a geometric distribution, and hence we chose a finer scaling to emphasize non-geometric behavior. Also, [2] uses the full 19-dimensional features as observations, but in our experiments we found the full dimensionality did not significantly affect performance while it did slightly increase computation time per sampling iteration.

Figure 4-8 shows the progression of nine different HDP-HSMM chains on the NIST_20051102-1323 meeting over a small number of iterations. Within two hundred iterations, most chains have achieved approximately 0.4 normalized Hamming error or less, while it takes between 5000 and 30000 iterations for the Sticky HDP-HMM sampler to mix to the same performance on the same meeting, as shown in Figure 3.19(b) of [3]. This reduction in the number of iterations for the sampler to "burn in" more than makes up for the greater computation time per iteration.

We ran 9 chains on each of the 21 meetings to 750 iterations, and Figure 4-9 summarizes the normalized Hamming distance performance for the final sample of the median chain for each meeting. Note that the normalized Hamming error metric is particularly harsh for this problem, since any speakers that are split or merged incur a high penalty despite the accuracy of segmentation boundaries. The performance is varied; for some meetings an excellent segmentation with normalized Hamming error around 0.2 is very rapidly identified, while for other meetings the chains are slow to mix. The meetings that mixed slowly, such as that shown in Figure 4-8, were generally the same meetings that proved difficult for inference with the HDP-HMM as well [3]. See Figure 4-11 for example sample paths of prototypical low-error and high-error meetings.

Finally, Figure 4-10 summarizes the number of inferred speakers compared to the true number of speakers, where we count speakers whose speech totals at least 5% of the total meeting time. For each number of true speakers on the vertical axis, each cell in the row is drawn with brightness proportional the relative frequency of that number of inferred speakers. The dataset contained meetings with 2, 3, 4, 5, 6, and 7 speakers, and the figure is extended to an 8×8 square to show the frequency of the inferred number of speakers for each true number of speakers. There is a clear concentration along the diagonal of the figure, which shows that the HDP-HSMM is able to effectively infer the number of speakers in the meeting by learning the appropriate number of states to model the statistics of the data.

Overall, this experiment demonstrates that the HDP-HSMM is readily applicable to complex real-world data, and furthermore that the significant mixing speedup in terms of number of iterations can provide a significant computational benefit in some cases.

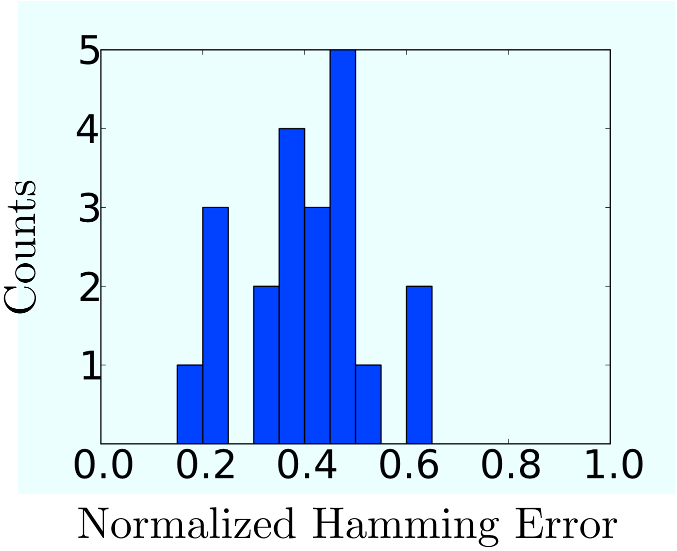


Figure 4-9: Diarization Performance Summary

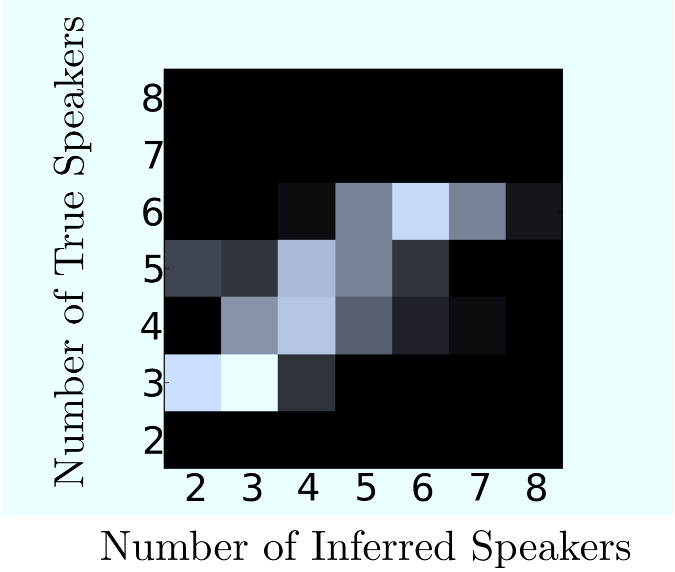
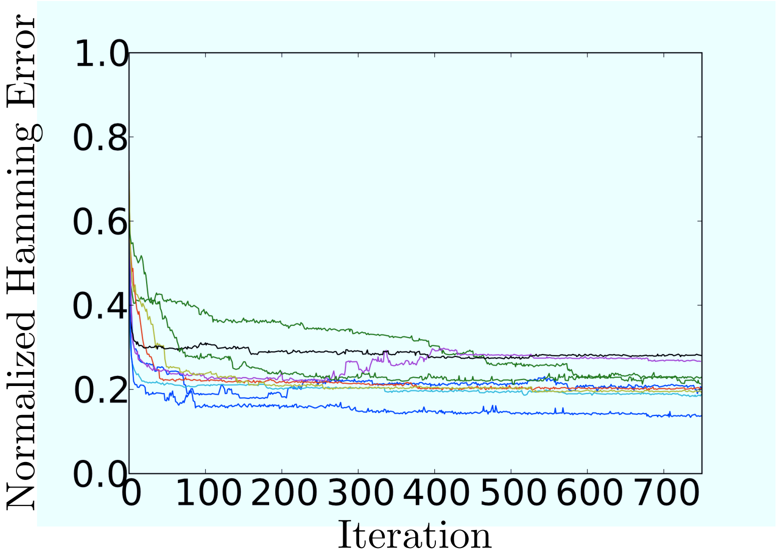
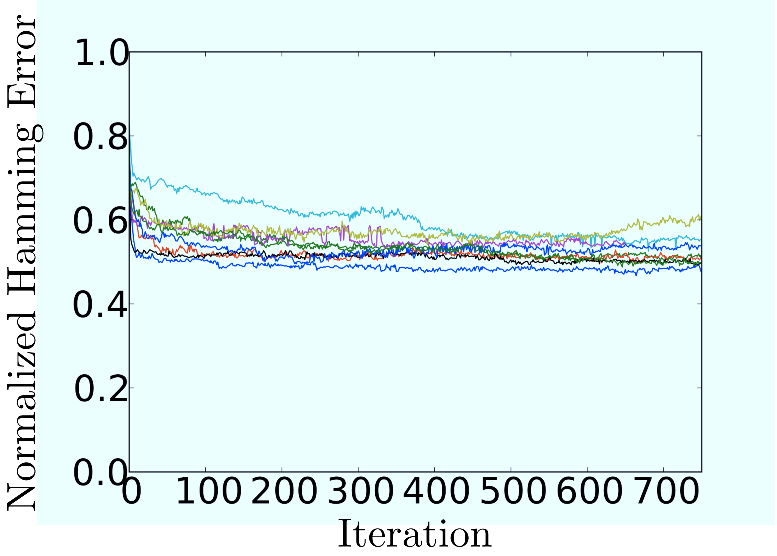


Figure 4-10: Frequency of Inferred Number of Speakers



(a) Good-performance meeting



(b) Poor-performance meeting

Figure 4-11: Prototypical sampler trajectories for good- and poor-performance meetings.

Chapter 5

Contributions and Future Directions

In this thesis we have developed the HDP-HSMM as a flexible model for capturing the statistics of non-Markovian data while providing the same Bayesian nonparametric advantages of the HDP-HMM. We have also developed efficient Bayesian inference algorithms for both the finite HSMM and the HDP-HSMM. Furthermore, the sampling algorithms developed here for the HDP-HSMM not only provide fast-mixing inference for the HDP-HSMM, but also produce new algorithms for the original HDP-HMM that warrant further study. The models and algorithms of this thesis enable more thorough analysis and unsupervised pattern discovery in data with rich sequential or temporal structure.

Studying the HDP-HSMM has also suggested several directions for future research. In particular, the HSMM formalism can allow for more expressive observation distributions for each state; within one state segment, data need not be generated by independent draws at each step, but rather the model can provide for in-state dynamics structure. This hierarchical structure is very natural in many settings, and can allow, for example, learning a speaker segmentation in which each speaker's dynamics are modeled with an HMM while speaker-switching structure follows a semi-Markov model. Efficient sampling algorithms can be made possible by employing a combination of HMM and HSMM message-passing inference. This richer class of models can provide further flexibility and expressiveness.

In summary, the HDP-HSMM provides a powerful Bayesian nonparametric modeling framework as well as an extensible platform for future hierarchical models.

Bibliography

[1] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[2] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "An HDP-HMM for systems with state persistence," in *Proc. International Conference on Machine Learning*, July 2008.

[3] E. Fox, "Bayesian nonparametric learning of complex dynamical phenomena," Ph.D. Thesis, MIT, Cambridge, MA, 2009.

[4] Y. Guédon, "Exploring the state sequence space for hidden markov and semi-markov chains," *Comput. Stat. Data Anal.*, vol. 51, no. 5, pp. 2379–2409, 2007.

[5] K. A. Heller, Y. W. Teh, and D. Görür, "Infinite hierarchical hidden Markov models," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, vol. 12, 2009.

[6] M. I. Jordan, *An Introduction to Probabilistic Graphical Models*. Unpublished Manuscript, 2008.

[7] D. Kulp, D. Haussler, M. Reese, and F. Eeckman, "A generalized hidden Markov model for the recognition of human genes," in *in DNA,??? in Proc. Int. Conf*, 1996.

[8] K. Murphy, "Hidden semi-markov models (segment models)," *Technical Report*, November 2002. [Online]. Available: http://www.cs.ubc.ca/~murphyk/Papers/segment.pdf

[9] I. Murray, *Advances in Markov chain Monte Carlo methods*. Citeseer, 2007.

[10] L. Rabiner, "A tutorial on hmm and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

[11] J. Sethuraman., "A constructive definition of dirichlet priors." in *Statistica Sinica*, vol. 4, 1994, pp. 639–650.

[12] W. Sun, W. Xie, F. Xu, M. Grunstein, and K. Li, "Dissecting Nucleosome Free Regions by a Segmental Semi-Markov Model," *PLoS ONE*, vol. 4, no. 3, 2009.

[13] Y. W. Teh, "Dirichlet processes," 2007, submitted to Encyclopedia of Machine Learning.

[14] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical Dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006.

[15] C. Wooters and M. Huijbregts, "The ICSI RT07s speaker diarization system," *Multimodal Technologies for Perception of Humans*, pp. 509–519, 2008.

[16] S. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, 2009.